

iSPY

Image Surveillance and Protection Yielder



University of Central Florida
Senior Design II – Fall 2024, Spring 2025

Group 5

Alejandro Lopez Zelaya
Armani Garcia
Gabryella Baldaci
Luke Reyes
Robert Josephson

Photonic Science and Engineering
Electrical Engineering
Photonic Science and Engineering
Computer Engineering
Computer Engineering

Instructors

Dr. Aravinda Kar
Dr. Lei Wei

Reviewer Committee

Dr. Guifang Li - CREOL
Dr. Suboh Suboh - ECE
Dr. Nazanin Rahnavard - ECE
Dr. Stephen Eikenberry – CREOL

Table of Contents

Chapter 1 – Executive Summary	11
Chapter 2 – Project Description.....	12
2.1 Project Background and Motivation	12
2.1.1 Background.....	12
2.1.2 Motivation.....	12
2.2 Commercial Technologies and Existing Projects	13
2.2.1 Koruza.....	13
2.2.2 MOSTCOM	13
2.2.3 Drone Mounted Object Detection & Recognition using ML and IR Sensors .	14
2.2.4 S.T.E.A.L.T.H. Secure Transmission via Electronic and Laser Technology Hardware.....	14
2.3 Goals and Objectives	15
2.3.1 Goals	15
2.3.2 Objectives	16
2.4 Required Specifications	18
2.5 System Diagrams	19
2.5.1 Hardware.....	19
2.6 House of Quality	29
Chapter 3 – Research	31
3.1 Imaging Sub-system.....	31
3.1.1 Sensor Parameters.....	31
3.1.2 Comparison of Camera Modules	35
3.1.3 The Optical Parameters.....	36
3.1.4 Image Acquisition System: The Optical Design.....	38
3.1.5 Lens Background	39
3.1.6 Lens Material	41
3.1.7 Lens Coating	41
3.1.8 Optical Aberrations.....	43

3.1.9 Lens Selection.....	44
3.1.10 System Performance	44
3.2 Computer Vision.....	45
3.2.1 C++ Computer Vision Library.....	46
3.2.2 Java Computer Vision Library.....	46
3.2.3 Python Computer Vision Libraries	47
3.2.3.1 TensorFlow	47
3.2.3.2 SimpleCV.....	47
3.2.3.3 OpenCV	47
3.3 Haar Cascade vs. Histogram of Oriented Gradients (HOG).....	48
3.3.1 Haar Cascade	48
3.3.2 Histogram of Oriented Gradients (HOG)	49
3.4 Newer OpenCV Algorithm's	50
3.4.1 YOLO (You Only Look Once)	50
3.5 Comparison of Computer Vision Algorithm Models	58
3.6 Classifiers and Object Detectors	59
3.7 Body Recognition vs Body Detection.....	60
3.7.1 Body Recognition	60
3.7.2 Body Detection	61
3.8 Live Video Compression and Decompression.....	61
3.8.1 Codec	61
3.8.2 Lossy vs Lossless.....	62
3.8.2.1 Lossy Compression.....	62
3.8.3 Video Compression Techniques	62
3.8.4 Live Transmission Limitations	64
3.8.5 Codec Comparison.....	65
3.8.6 Transmission of Data	66
3.8.7 Loss of Data	67
3.8.8 Video Compression Libraries	67
3.8.9 MCU Selection.....	70
3.9 Communication Sub-System	77

3.9.1 Communication.....	78
3.9.2 Electrical Communications	80
3.9.3 Optical Communications	81
3.10 Different Types of Power Supplies.....	101
3.10.1 Mean Well.....	102
3.10.2 RS Pro	103
3.10.3 CUI Inc.....	104
3.10.4 Product Comparison.....	105
3.11 Voltage Regulation	105
3.12 Differential Impedance	106
Chapter 4 – Related Standards and Design Constraints.....	108
4.1 Industrial Standards	108
4.1.1 Related Electrical Standards	108
4.1.2 Related Optics Standards	111
4.1.3 AI Standards.....	113
4.1.4 Surveillance Systems Standards	114
4.1.5 Camera Module Standards.....	114
4.1.6 Privacy Regarding Surveillance Systems	114
4.2 Practical Constraints	115
4.2.1 Time	115
4.2.2 Economic	116
4.2.3 Environmental.....	117
4.2.4 Remaining Constraints.....	118
Chapter 5 – Large-Language Model Engine.....	120
5.1 Case Study 1: Asking ChatGPT and Copilot for Camera Module Specifications	120
5.2 Case Study 2: ChatGPT vs. Zemax.....	120
5.3 Case Study 3: Asking Multiple AI Models for Power Supply Recommendations	121
5.4 Case Study 4: Camera Module Options Using ChatGPT	121
5.5 Takeaways.....	122
Chapter 6 – System Hardware Design	124
6.1 Image Acquisition System Hardware	124

6.2 Optical Communications Hardware.....	131
6.3 Power Delivery/Electrical Power System.....	133
6.4 MCU/Peripheral System	134
6.4.1 Ethernet Connection from Transceiver	135
6.4.2 HDMI Connection to Display.....	136
6.4.3 USB Connection for Programming.....	137
Chapter 7 – System Software Design	139
7.1 Software Case Design.....	139
7.2 Software Flow Design.....	140
7.3 Computer Vision Setup.....	141
7.3.1 Importing Library and Model	142
7.3.2 Setting the Channel	142
7.3.3 Scaling Image or Frame	142
7.3.4 Function for Recognition	143
7.3.5 Updating Counter.....	143
7.4 Computer Vision Program	144
Chapter 8 – Prototype Construction.....	147
8.1 Image Acquisition System Prototype.....	147
8.2 Optical Communication Prototype	154
8.3 Computer Vision Program Prototype.....	160
8.4 PCBs	162
8.5 Raspberry Pi Video Compression Prototype	165
8.6 Final Setup	168
Chapter 9 – System Testing and Evaluation	170
9.1 Image Acquisition System Simulation.....	170
9.2 Optical Communication Testing.....	175
9.3 Computer Vision Program Simulations	179
9.3.1 Haar Cascade Simulation.....	180
9.3.2 HOG Simulation	181
9.3.3 YOLO Simulation.....	182
9.4 Computer Vision 1 st Iteration.....	186

9.5 Computer Vision 2 nd Iteration.....	186
9.6 Computer Vision Raspberry Pi 5 Code.....	186
9.7 Computer Vision AI Kit Code	187
9.8 Video Compression and Transmission	187
Chapter 10 – Administrative Content	191
10.1 Budget and Bill of Materials.....	191
10.2 Distribution Worktable	192
10.3 Milestones	193
10.4 Acknowledgements and Declaration	194
Chapter 11 – Conclusion.....	195
Appendix A – References	197
Appendix B – Copyright Permission	212
Appendix C – Data Sheet.....	213
Appendix D – ISO 10110 Drawing	215
Appendix E – Software Code	216
Appendix F – ChatGPT prompts and outcomes	228
Appendix G – Zemax Images	267
Appendix H – Computer Vision	269
Appendix I – Video Compression Images	272
Appendix J – Schematics/PCB Images.....	274

List of Tables

Table 2.1 - Engineering requirements specifications for the performance and physical features of the system.....	8
Table 2.2 - Engineering requirements specifications for the components.....	9
Table 2.3 - Optical components used in the system.....	9
Table 3.1 - Comparison table of different camera modules.....	25
Table 3.2 - Comparison table of computer vision algorithms.....	48
Table 3.3 - Comparison table of different codecs.....	56
Table 3.4 - Comparison table of different libraries for C++.....	58
Table 3.5 - Comparison table of different libraries for java.....	59
Table 3.6 - Comparison table of different libraries for python.....	59
Table 3.7 - Comparison table for different brands of MCUs.....	61
Table 3.8 - Comparison table for different models of Raspberry Pi boards.....	62
Table 3.9 - Specs for Raspberry Pi Compute Module 4.....	63
Table 3.10 - Comparison table of different MCUs.....	65
Table 3.11 - Specs for STM32H753VIT6.....	65
Table 3.12 - Storage requirements for different resolutions and frame rate.....	67
Table 3.13 - Differences across different module form factors of the same brand.....	89
Table 3.14 - Differences across SMF and MMF SFP modules of different brands.....	90
Table 3.15 - Power Supply Comparison.....	95
Table 4.1- DoE Level VI Requirements.....	100
Table 10.1 - Itemized bill of materials.....	178
Table 10.2 - Distribution worktable.....	179
Table 10.3 - Senior Design 1 Milestones - Project design phase.....	180
Table 10.4 - Senior Design 2 Milestones - Project creation phase.....	181

List of Figures

Figure 2.1. General project diagram.....	6
Figure 2.2. General program schematic.....	8
Figure 2.3. Hardware diagram.....	10-11
Figure 2.4. Layout of a wide-lens system.....	12
Figure 2.5. Field of view captured.....	13
Figure 2.6. FSO transmission/receiver system diagram.....	14
Figure 2.7. Optical schematic for FSO.....	16
Figure 2.8. Software diagram.....	18
Figure 2.9. House of Quality.....	20
Figure 3.1. Zemax layout of a 100-degree field of view system.....	28
Figure 3.2. A comparison between the standard lenses and the meniscuses.....	31
Figure 3.3 A comparison between uncoated and coated lenses.....	32
Figure 3.4. A demonstration of how contrast is measured.....	35
Figure 3.5. A simple demonstration of the MTF graph.....	35
Figure 3.6. PGI Architecture.....	42
Figure 3.7. GELAN Architecture.....	42
Figure 3.8. Comparison of State-of-the-Art Real-Time Object Detectors.....	43
Figure 3.9. YOLOv10 Model Variations.....	45
Figure 3.10. YOLOv11 Model Variations.....	46
Figure 3.11. YOLO-NAS Model Variations.....	47
Figure 3.12. Image Depicting Spatial Compression (Intraframe Compression).....	53
Figure 3.13. Image Depicting Temporal Compression.....	53
Figure 3.14. Assigned Bit Lengths of Letters in Sequence.....	54
Figure 3.15. Demonstration of PCM Output.....	56
Figure 3.16. Image of Raspberry Pi Compute Module 4.....	63
Figure 3.17. Linearity of one-way communication.....	68

Figure 3.18. Direct and non-direct (feedback) two-way communication.....	69
Figure. 3.19. Simplistic optical fiber schematic showcasing.....	72
Figure. 3.20. Simplistic electromagnetic beam divergence schematic comparison between an RF beam and optical beam.....	76
Figure. 3.21. Simplistic representation of a non-collimated beam diverging past the aperture and collimated beam.....	77
Figure. 3.22. Step-index profile.....	79
Figure. 3.23. Graded index profile.....	80
Figure. 3.24. Dispersion shifted refractive index profile.....	80
Figure. 3.25. Trench assisted step index profile.....	81
Figure 3.26. An optical transceiver and its breakdown.....	82
Figure 3.27. An optical fiber interfaces with an optical transceiver.....	83
Figure 3.28. Examples of different form-factor modules.....	84
Figure 3.29. A) Edge-emitting semiconductor laser. B) Vertical-cavity surface-emitting laser.....	85
Figure 3.30. Logic difference between NRZ and PAM4.....	87
Figure 3.31. Constellation diagram of a 16-QAM modulated signals.....	88
Figure 4.1. Hierarchy of IPC Design Standards.....	99
Figure 6.1. Pinhole camera.....	114
Figure 6.2. Illumination test.....	115
Figure 6.3. iSPY's final lens design.....	117
Figure 6.4. Zoom-into the aperture stop.....	117
Figure 6.5. 3D model of iSPY's lens design.....	117
Figure 6.6. Optical assembly for the image acquisition system.....	118
Figure 6.7. Encircled energy from the lens system.....	119
Figure 6.8. MTF from the lens system.....	119
Figure 6.9. Fiber collimator lens TC25FC-1064 from Thorlabs.....	122
Figure 6.10. Kinematic mount with a collimator lens.....	123

Figure 6.11. Schematics for the power delivery system.....	124
Figure 6.12. Schematics for MCU connections.....	125
Figure 6.13. Schematics for Ethernet connections.....	126
Figure 6.14. Schematics for HDMI port and HDMI transmitter.....	127
Figure 6.15. Schematics for Micro-USB connections.....	128
Figure 7.1. Use Case Diagram for iSPY system.....	129
Figure 7.2. Software State Flowchart.....	130
Figure 8.1. Diagram of the imaging acquisition prototype set-up with a single lens....	138
Figure 8.2. Frame taken by the laptop connected to the CMOS sensor.....	139
Figure 8.3. Diagram of the imaging acquisition prototype set-up with two lenses.....	140
Figure 8.4. Frame taken by the laptop connected to the CMOS sensor.....	140
Figure 8.5. An actual image of how the prototype was put together.....	141
Figure 8.6. Final structure of Image Acquisition System.....	142
Figure 8.7. FOV test step by step.....	143
Figure 8.8. Live video capture by iSPY.....	144
Figure 8.9. Transmitter for prototype #1.....	145
Figure 8.10. Receiver for prototype #1.....	145
Figure 8.11. Completed prototype #1.....	146
Figure 8.12. Transmitter for prototype #2.....	147
Figure 8.13. Receiver for prototype #2.....	147
Figure 8.14. Focusing lens for prototype #2 receiver.....	148
Figure 8.15. Focal spot overfilling prototype #2 receiver to maximize signal.....	148
Figure 8.16. LED-based free-space optical communication prototype #2.....	149
Figure 8.17. Final FSO Prototype – Prototype #3.....	150
Figure 8.18. Zemax Generated Image.....	150
Figure 8.19. PCB #1: Base Carrier.....	152
Figure 8.20. PCB #2: Ethernet Daughter Board.....	152

Figure 8.21 PCB #3: Voltage Regulator.....	153
Figure 8.22 Fully Constructed PCB.....	153
Figure 8.23. Constructed PCB Using Breakout Board.....	154
Figure 8.24. Raspberry Pi Module.....	154
Figure 8.25. Final file sizes of compression and decompression testing.....	154
Figure 8.26. Video feed before transmission on Raspberry Pi.....	156
Figure 8.27. Video feed after transmission on computer.....	156
Figure 8.28. iSPY's final setup.....	157
Figure 9.1. Lens design – Paraxial approximation.....	160
Figure 9.2. Simulations from the third round of trials.....	161
Figure 9.3. Negative meniscus followed by positive lenses.....	161
Figure 9.4. Second attempt on inverted telephoto lens system.....	162
Figure 9.5. Attempt on inverted telephoto lens system with an achromatic lens.....	163
Figure 9.6. Final lens design for iSPY.....	163
Figure 9.7. Larger entrance pupil diameter.....	164
Figure 9.8. LED-based free-space optical communication prototype.....	166
Figure 9.9. First all-in fiber of testing of optical transceivers using SMF-28e fiber	167
Figure 9.10. Test bench for the transmission of video packets via free-space optics	167
Figure 9.11. Resulting Image from Haar Cascade algorithm program.....	168
Figure 9.12. Resulting Image from HOG algorithm program.....	169
Figure 9.13. Resulting Image from YOLOv11 algorithm.....	170
Figure 9.14. Resulting Image from YOLOv10 algorithm.....	171
Figure 9.15. Window before transmission (Example 1).....	176
Figure 9.16. Window after transmission (Example 1).....	176
Figure 9.17. Window before transmission (Example 2).....	177
Figure 9.18. Window after transmission (Example 2).....	177

Chapter 1 – Executive Summary

As time passes and technology develops, there is a rising need to enhance surveillance systems so that a maintaining of high security is achieved. To help propel this innovation further, we have iSPY, which is short for Image Surveillance and Protection Yielder. As an interdisciplinary team of photonics, computer, and electrical engineers, this project is a pioneering effort to develop a secure surveillance system utilizing Free-Space Optical (FSO) communication and Computer Vision algorithms. Aiming to produce an enhanced system capable of maintaining high security and efficiency in various applications.

The key innovation of iSPY is the use of FSO communication to replace conventional transmission methods such as Wi-Fi or a radio-based transmission method. With this approach we can focus on reducing the vulnerabilities that arise when using a conventional method. This approach to the issue makes iSPY an effective solution to explore with the concepts used in its development. By transmitting data video light, iSPY ensures enhanced proximity-based security, which is what we aim to achieve as optical signals are much harder to intercept without physical access to the system. With the integration of a computer-vision based algorithm, we can further enhance the functionality with an object detection system that is capable of tracking individuals in the monitored area to a high degree of accuracy.

The hardware architecture of iSPY is built around three key sub-systems: image acquisition, FSO communication, and electrical hardware. Employing a wide field-of-view and computer vision software, the image acquisition sub-system captures and preprocesses live video. The FSO sub-system transmits that compressed data securely over the air in a free-space optical channel to a receiver unit, which decompresses and displays that data to an external display. The electrical hardware sub-system includes the two PCBs involved in the entirety of the system, the first being a Raspberry Pi 4 Model B and the second being a custom design that can handle efficient data processing, decompression, and signal integrity.

Through designing and testing, demonstrates our team's commitment and intention of innovating and redefining the standards for secure surveillance systems. With collaboration and the application of many engineering principles, the product shall offer a solution with broad societal implications, in a wide range of environments, whether that be in urban settings, military zones, or natural disaster sites, iSPY has the potential to enhance the protection of sensitive data, situational awareness, and could improve the response time indifferent scenarios.

Chapter 2 – Project Description

In this chapter we will go over the project in detail. We highlight our motivations and what we accomplished (goals and objectives), followed by a list of specifications and descriptive figures related to our design to effectively explain to our audience (market) what the final product is. At the same time, we talk about products in the market with similar functionality to the one we are trying to build.

2.1 Project Background and Motivation

This section is dedicated to the work that has been done in the field of surveillance, how our project relates to that, and the motivation behind our idea – our intentions in this project.

2.1.1 Background

The history of espionage has always been intertwined to optics and photonics. From the research on lightweight cameras for the U2 surveillance aircraft in the 1940s to the recent creation of unmanned aerial vehicles (UAV) [1], [2]. The need for such technologies guides the direction of present research.

Joe Houston, a SPIE Fellow and past SPIE president highlighted the importance of the innovation in already known systems, “The most covert systems that I know of are those systems – either image communications or electronic communications – that are hidden in the skirts of strong signals in a location that you know, but no one else knows” [1].

Our system, iSPY is then a new chapter of this history. The idea is not to hide the cameras but make the transmission of their captured video something untraceable and uncrackable.

Just like many other technologies that were once created having the military in mind, we hope that iSPY can make its way to the everyday market and bring safety to civilian users of surveillance systems since security is a priority in our daily life. Whether you are a homeowner, a soldier, or a first responder, having eyes on the place you want to protect is a necessity. To prevent crimes from happening, save lives during emergency situations, or to reduce the reaction time if an incident does take place, surveillance systems are widely used.

These surveillance systems usually consist of several cameras that capture live footage of their surroundings. The footage is then transmitted to a device(s), via cable or radio, to display a real-time video of the observed location or objects.

2.1.2 Motivation

Due to the rise of crime, the need for surveillance systems has increased. And unfortunately, these technologies, when poorly used, can lead to more safety issues. Several news outlets have reported the hacking of camera systems [3], [4]. The issue is not only in home security systems, but it is a subject of international security as reported by Scott Ikeda from CPO Magazine. He wrote on Russian military attaining crucial information related to landscape of cities by hacking into cameras Ukrainian citizens used in their homes and businesses [5].

Most security systems are connected through Wi-Fi, which unfortunately, is the most common path for hackers. One of the shared tips on how to secure these devices is to disconnect them from the internet [6]. However, another problem arises with this measure, data transfer cannot take place without Wi-Fi or other means of radio communication.

In iSPY, the need for Wi-Fi or radio signal is to be replaced by a free-space optical (FSO) communication system, eliminating the security risk associated with using Wi-Fi or other radio connection. Since data is transmitted through light, it has some features that makes it more secure than the alternative version. One of them is proximity. Light diffracts much less than radio signals, making it hard for an outside entity to tap into the signal being transmitted. Without access to the area where the light is shone it is impossible to access the information. Another attribute of the system is that the receiver's detector is designed and tuned for that specific transmitted signal, the detector is unique to that system, decreasing and almost eliminating the chances of someone matching the specifications needed to decode the data.

2.2 Commercial Technologies and Existing Projects

Market research was done to better understand the specifications and parameters we should keep in mind when building our project. We searched for different products and/or projects, we took notice of how they work, and we tried to understand their strengths and flaws. In this section, we will discuss those products and/or projects and relate them to our own iSPY.

2.2.1 Koruza

Koruza is a wireless optical internet-access system that advertises the secure transmission of data and easy installation process. A product on the market that matches some of the goals we want to achieve with iSPY. They are also the only open-source free space optical system available.

The system was designed to facilitate data transfer in environments where other options (wireless and fiber optics) are not feasible solutions. Koruza is also easily upgraded, as they say, it is “future proof” [7], [8].

2.2.2 MOSTCOM

MOSTCOM is a well-established Russian company that specializes in free-space optics research and development on earth and space. They have a product line that includes

stationary optical systems (OGS), ground-to-space terminals, and mobile terminals. Their current research is on underwater communication terminals [9].

2.2.3 Drone Mounted Object Detection & Recognition using ML and IR Sensors

This was a previous senior design project from a group based out of the University of Central Florida in Spring 2023. This effort combined the ideas of artificial intelligence (AI) and unmanned aerial vehicles to create an autonomous object detection drone. The primary goal for this project was to create a drone-mountable object detection system enabling artificial intelligent visual navigation. The secondary goal was to develop and integrate autonomous peripherals for the drone. The need for this sort of drone technology is widespread for the defense, emergency response, and agriculture industries. The idea, combined goals, and industries of interest of this project are very relevant for iSPY and must be considered closely.

Without knowledge of this project, iSPY intends to build on and design around the successes of this previous collaboration. For example, one of iSPY's stretch goals is the integration of an object detection and free-space optical communication systems on to a drone. Since iSPY is a surveillance system, object detection through computer vision increases the amount of information made available for the end user's discretion. In defense situations, drone-mounted, AI-assisted object detection systems can be employed to detect guns, explosives, or enemy targets in a region of interest. In emergency situations like wildfire, a version of the drone-mounted, AI assisted object detection system could be used to detect trapped people through the foliage. In agriculture, another version of the AI-assisted system could be used to detect pests amongst crops like groundhogs or certain weeds. It's important to note that iSPY doesn't plan to use a drone to demonstrate AI-assisted object detection; however, it's a stretch goal the team is considering if time permits.

The main difference between iSPY and this previous collaboration is the means of communication with the end user. One of the main motivations for iSPY is to implement free-space optical communication as a means of limiting outsider interference to the signal. If surveillance is to be achieved in secret, then a wire or radio signal cannot be used. Instead, an optically confined light pulse is employed. To interfere, or tap, the signal, the optical must be physically intercepted, making the intrusion known.

2.2.4 S.T.E.A.L.T.H. Secure Transmission via Electronic and Laser Technology Hardware

The project was a previous senior design project from a group at the University of Central in Spring 2022. The project focused on a bidirectional Optical link communication system which utilizes the transmission of data and receiving of said data. In this project the goal

was to make the optical wireless communication system more modular and mobile as well as focusing on low size, weight plus power budget within reason.

S.T.E.A.L.T.H. used two different PCB's. One of the PCB's was for serializing and the other was for de-serializing. The design of iSPY intends to utilize a similar transmission of data except without serialization and de-serialization PCB's. Instead iSPY uses FSO transmission which sends data via digital signal through free space and not an optical link or through PCB's that encode/decode the data.

2.3 Goals and Objectives

The main idea of this project is to build a secure communication, computer-vision-capable object detection surveillance system. A working iSPY device relays a real-time surveillance video feed to a display by means of FSO. A product of this caliber has the capacity to support defense, emergency response, and agriculture industries, by yielding the end user valuable information for their discretion and mission. The basic, advanced, and stretch goals and objectives for this project are outlined below.

2.3.1 Goals

To get the project running, the team devised three tier-lists of goals – basic, advanced, and stretch goals. The goals are organized in increasing difficulty, respectively. This gives the team a plan on which to make the project a success.

Basic Goals

- Capture faint images of the environment surrounding the sensor.
- Transmit and receive a laser pulse.
- Designed PCB must effectively house a processing unit, electrical components, power systems, and the necessary connection pins.
- Able to detect objects with the use of computer vision algorithm in video feed.
- Achieve stable power distribution across all components.
- Establish a stable method that can efficiently compress data for transmission and decompress data for viewing on display.

Advanced Goals

- Capture a clear image of the surrounding environment with 80 degrees FOV.
- Transmit and receive usable optical signals using integrated circuits at 1Gbps.
- The power systems must power the imaging acquisition system and an external processing board.
- Capture body recognition as well as count the number of individuals recognized from video feed.
- Minimize signal interference and ensure data integrity on high-speed transmission lines.
- Achieve a high-quality video feed of 720p after compression and decompression.

Stretch Goals

- The iSPY system is modular – can be spread across a drone and a tower.
 - Imaging system and transmitter optics found on the drone.
 - Reception optics found on the tower.
- Aberration-corrected, high-resolution, 120° FOV images.
- Ultra-fast transmission speeds using high-speed FSO transceivers (+\$2000 ea.).
- Tracking body movement of individuals in video feed with confidence scoring and enhancing recognition of people to determine if male or female, young or old and adult or child.
- Optimize thermal management for extended operation.
- Implement compression that is capable of dynamically adapting to video complexity which would allow much higher quality of video feed.

Regardless of the tier level achieved, the end-product will carry out the same function. Additionally, the end-product presented at the end of Senior Design 2 looked similar to the figure below.

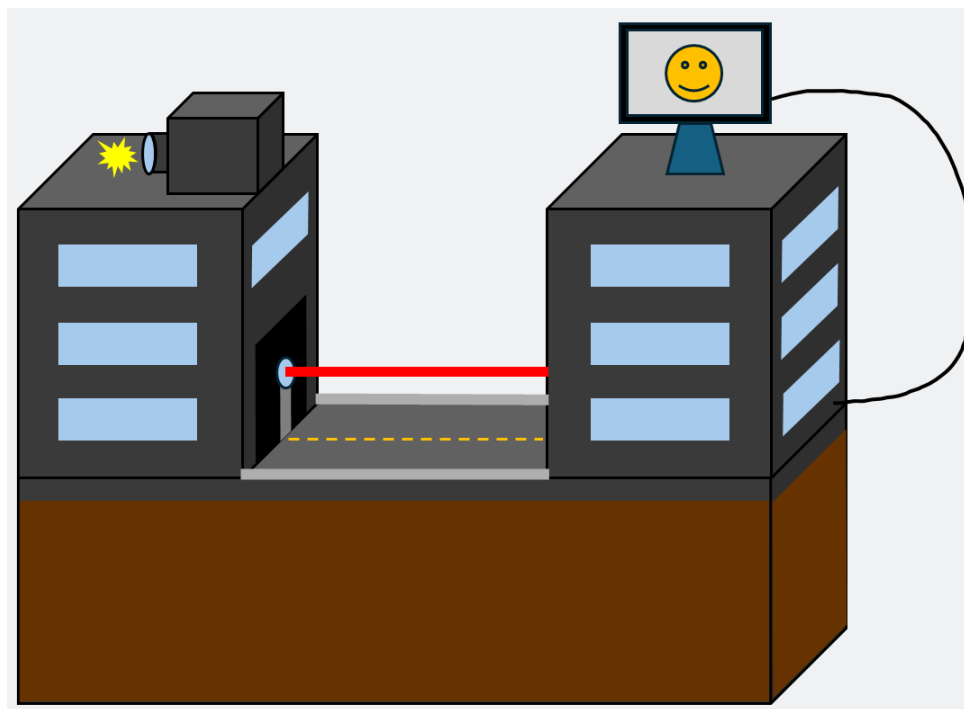


Figure 2.1. General project diagram (not to scale) by Authors.

2.3.2 Objectives

To effectively achieve the listed goals for the project, objectives must be established and understood for each. The objectives, divided according to goal type, are discussed below.

Objectives for Basic Goals

- OOK laser diode and measure electrical signal on photodetector.

- Create a two-lens system to capture an image.
- Design a PCB layout that works for the envisioned system.
- Utilize a computer vision algorithm to detect objects.
- Design a robust power management circuit that can regulate voltage and current for the envisioned system.
- Implement a compression algorithm in our system that can reduce the size of data to reliably transfer data via the FSO system.

Objectives for Advanced Goals

- Integrate optoelectronic transceivers to transmit and receive usable optical signal.
- Create a wide field of view ($\geq 80^\circ$), continuous image feed.
- Test PCB with all the components and confirm that power system is giving desired output ($\leq 5V$).
- Program code for body recognition with an accuracy level of ($\geq 75\%$) and implement a counter for recognized bodies.
- Implement shielding and controlled impedance techniques in the layout of the PCB.
- Implement advanced codecs for real-time compression to achieve low latency and quality display.

Objectives for Stretch Goals

- Design a lens system that captures a high-resolution video and correct aberrations.
- Create a transmission/receiver system with a transceiver, evaluator board, and MCUs.
- Integrate iSPY into an aircraft to survey larger areas and increase applicability.
- Enhance the code to track the bodies, include confidence scoring and break recognition of person into sub-categories as well as increase accuracy of recognition to ($\geq 90\%$).
- Integrate thermal management solutions (heat sinks, thermal vias, etc).
- Develop an AI-driven algorithm to compress and decompress video with higher quality and faster processing.

From a very high level and broad perspective, after achieving the listed objectives, the project has a layout like below.

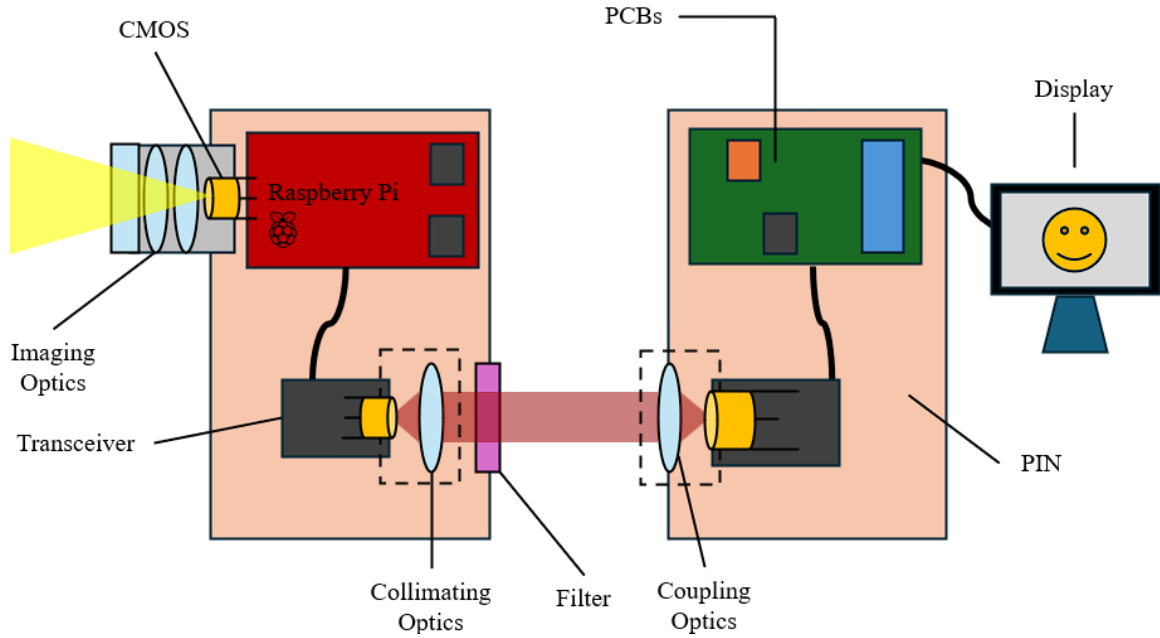


Figure 2.2. General program schematic by Authors.

2.4 Required Specifications

Considering the proposed goals and objectives, the required product specifications are:

Table 2.1. Engineering requirements specifications for the performance and physical features of the envisioned system.

Product requirement	Requirement specification	Unit
The system captures a wide field of view (horizontal)	≥ 80	Degrees
The system captures a wide field of view (vertical)	≥ 50	Degrees
The system produces a high image resolution	720	Progressive scan
The system is designed for low power consumption	≤ 20	Watts
The system has a high data transmission rate	≥ 500 M	Bits per second
The system can simulate a long-haul communications link.	≥ 10	Kilometer
The system transmits an optical signal over some physical distance	0.5	Meter
The system has some latency for processing time	1000	milliseconds
The system is reliable when detecting people (accuracy)	≥ 75	Percent
The system is able to detect multiple individuals	3	People

Table 2.2 Engineering requirements specifications for the components that is be used in this project to build the envisioned system.

Component(s)	Parameter	Specifications
Image acquisition system (Camera)	Field of view	80 degrees (horizontally) 50 degrees (vertically)
Computer vision algorithm (MCU)	Accuracy of detection	75%
Communication system (Optical Link)	Emulated Distance	> 10 km
MCU	Clock Speed	16 MHz
Camera sensor	Framerate	30 frames per second
Power Supply (External)	Voltage Output	5V DC regulated
Display	Framerate	30 frames per second

Tables 2.1 and 2.2 show the engineering requirements specifications for this project. Our goal is to showcase these specifications, especially the ones in the highlight rows.

Table 2.3 – Optical components used in the envisioned system.

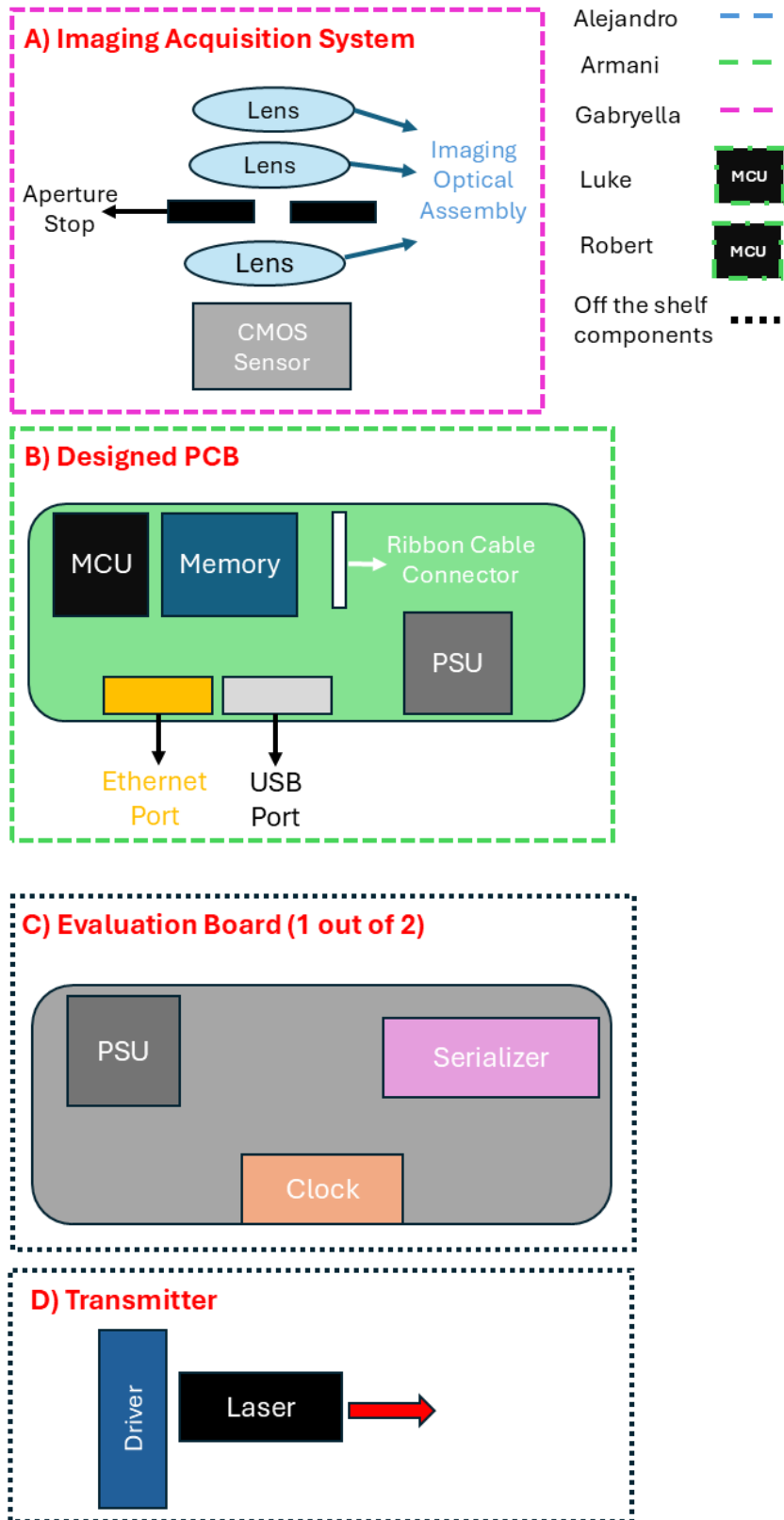
Component	Quantity
Lenses	4
Transceiver	2
Optical Fibers	2
ND Filters	1

2.5 System Diagrams

The engineering scope of this project is divided amongst hardware and software systems, each working in tandem to make iSPY work. This section discusses and presents the hardware and software diagrams necessary to make this project a success.

2.5.1 Hardware

The hardware is divided into three separate sub-systems: imaging acquisition, FSO communications, and electrical hardware. The hardware layout and separation of work is demonstrated below.



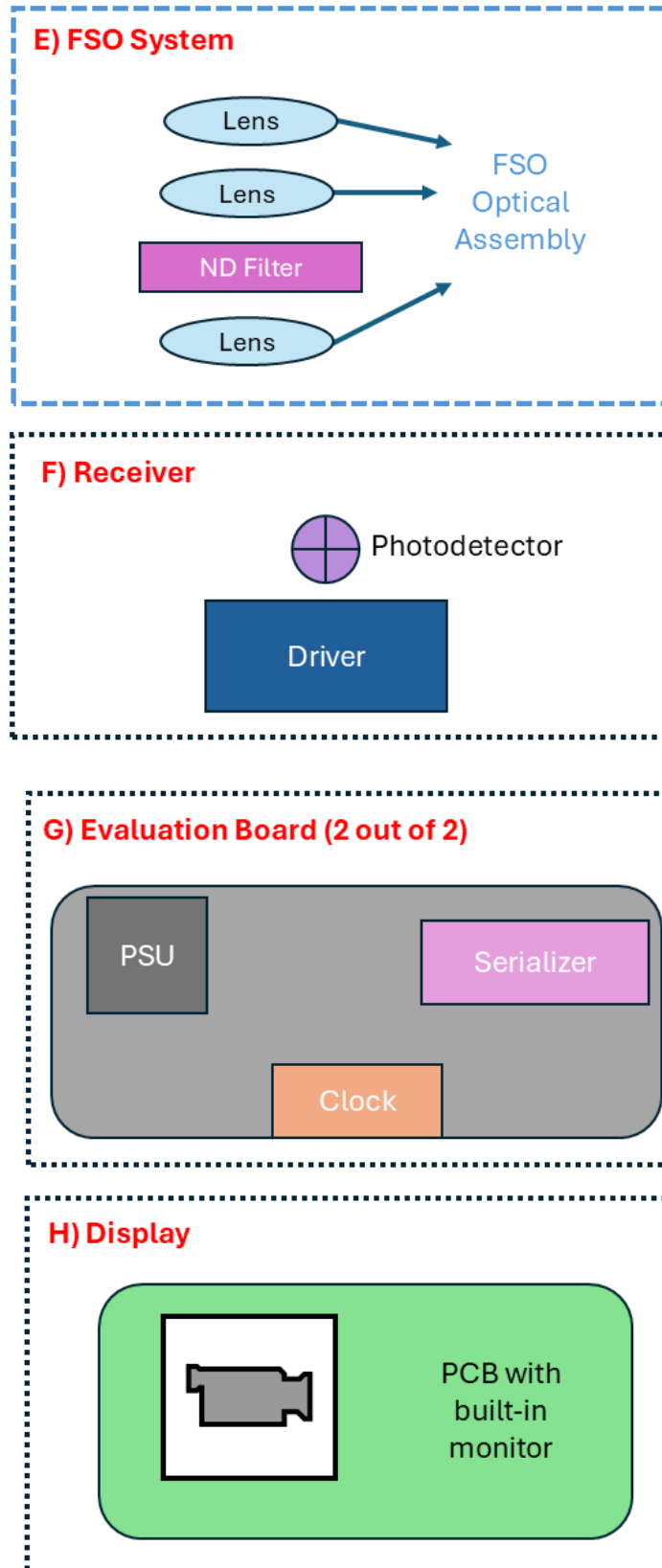


Figure 2.3. Hardware diagram by Authors

2.5.1.1 Image Acquisition Sub-System

One of the most crucial parameters for surveillance systems is the camera's field of view. Field of view "is the open, observable area a person can see through their eyes or via an optical device, such as a camera" [10].

The goal with iSPY is to have a wide field of view without sacrificing too much of the resolution. A wide field of view is needed to capture a large portion of the scene and decrease blind spots. At the same time, high resolution is necessary since the computer vision system requires as much detail as possible to determine what is taking place. Additionally, high resolutions enable the end user to best discern between two or more objects that are close to each other.

The system is designed to be placed in spaces where observing a larger portion of the surroundings is more important than seeing "crisp" details. This system consists of an assembly of lenses designed exclusively for this project. The lenses focus the light, create an image on the sensor, and correct optical aberrations affecting the imaged scenario. These optical components are mounted before a CMOS sensor – forming iSPY's own camera and our image acquisition system.

Choosing the different parts of this system, understanding their interaction, and simulating outcomes on the ray optics simulation software Zemax was the key to obtaining a successful prototype.

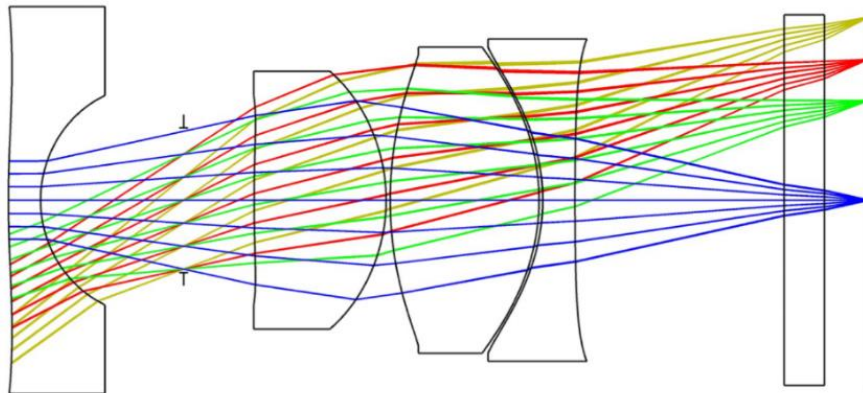


Figure 2.4. Layout of a wide-lens system from Reference 11. [Permission Requested]

Figure 2.4 shows the design of a wide-lens system [11]. Wide-lens and fisheye lens systems usually follow a similar structure, and this is what we had in mind in the beginning. Our system follows industry's frameworks and expands to match the requirements of our design.

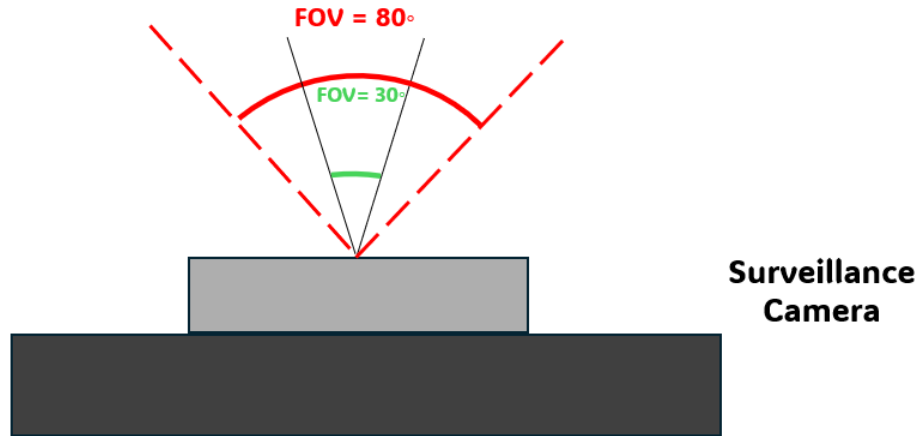


Figure 2.5. Field of view captured by Authors.

Figure 2.5 shows a ray optics model of the field of view captured by the detector. Some cameras can cover the larger region (180 degrees). However, the optics introduce many optical aberrations making high resolution, minimally distorted images difficult to produce for this project. Therefore, we focused on a smaller portion of the observable area and focus on the quality of the image.

Having the scope of this class in mind and facing some uncertainty due to the cost and size of equipment, our visual monitoring system consists of only one image detector and the focuses on capturing live videos of a well-lit environment, such as, inside a building or outdoors during daytime. One of our future goals (stretch goals) is to implement the system outside, mount iSPY onto drones, and have it running regardless of the time of the day.

2.5.1.2 Free-Space Optical Communications Sub-System

For iSPY to achieve secure communications link, FSO communications is employed. An FSO communications system consists of a transceiver (Tx/Rx) for optical signal transmission (Tx) and reception (Rx), optical components, and a signal evaluation board. Though these components are distinct, they must work as a unit to achieve a common goal: to relay the captured live-video feed to the end user. Together, the FSO communications sub-system could ideally look as does below:

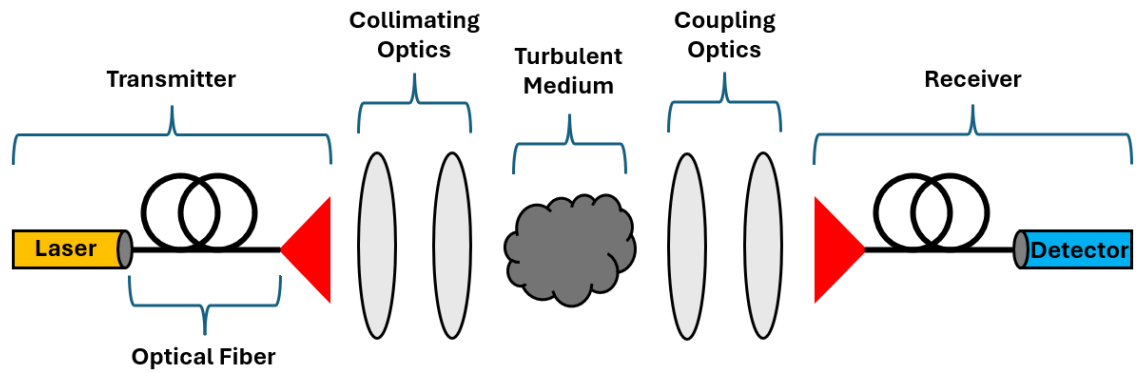


Figure 2.6. FSO transmission/receiver system diagram by Authors.

Optical communications systems, like fiber optic communication or FSO communication systems, typically reach transmission speeds in the order of a couple MHz to tens of GHz. In telecommunications, especially fiber optic communications, bidirectional communication is common and done frequently using an optical transceiver. An optical transceiver is an optoelectronic integrated circuit consisting of a laser diode and a photodetector. In practice, a transceiver takes in two optical fibers: one for Tx and one for Rx. The Tx optical fiber is connected directly to the laser diode output, whereas the Rx optical fiber is connected directly to the photodetector input. Transceiver technology has improved tremendously over the last 40 years, making it an enabling technology for the success of this project.

In iSPY, a transceiver is used as the optoelectronic relay to accomplish Tx and Rx. On the Tx side, the transceiver is to be responsible for generating an intensity modulated laser pulses that correspond to the live video digital signal bit stream. The laser diode is directly responsible for this process. The generated laser pulses are first be transmitted via optical fiber. The optical fiber transports the laser pulses to the free-space channel. To have adequate control of the laser pulses in the free-space channel, the pulses must be Gaussian in shape. This can be achieved using an optical fiber or a spatial filter within the FSO transmission optics. On the Rx side, the transceiver is to be responsible for generating an electrical signal that corresponds to the detected incident laser pulses. The photodetector is directly responsible for this process. The incident free-space laser pulses are first coupled into an optical fiber to guide the pulses to the transceiver photodetector, terminating the Tx/Rx link.

Unlike other optical communications systems, like optical fiber communications where the means of data transmission is confined in a fiber optic cable, FSO communications systems transmit data in free-space, open-air channels. This type of system is not confined by any conventional means like a copper or fiber optic cable. Thus, FSO systems must adequately control the beam size and divergence of its free-space laser beam.

The free-space input and output laser pulses are guided from and to the transceiver by an optical fiber. Optical fibers are low-loss, light waveguides with unique optical properties. Using an optical fiber as an input to the free-space channel comes with some considerations. Single-mode or multi-mode optical fibers may be used for this project; single-mode fibers only guide one spatial light mode whereas multi-mode fibers can guide hundreds. Multi-

mode fibers have high light-capturing capabilities because of their high numerical aperture compared to single-mode fibers. However, the bitrate must be closely considered for multi-mode fibers because of the limited modal bandwidth associated with all multi-mode fibers. Single-mode fibers enable very high bitrates over long distances but are hard to use for free-space applications due to their low light-capturing capabilities. Besides, all fibers suffer from material attenuation and dispersion. Material attenuation is how much light is absorbed by the optical fiber, decreasing the strength of the guided signals. Dispersion is the spreading of guided signals by means of color, mode, or polarization. Managing both attenuation and dispersion must be done under close consideration of each other. Thus, choosing an optical fiber that enables the bitrates of the system, has good light-capturing capabilities, is cheap, and can deliver signals over short distances is critical.

When light exits from an optical fiber, it diverges according to the guided optical mode present at the output – Gaussian for single-mode fibers and some higher-order mode for multi-mode fibers. Controlling the divergence and shape of the free-space pulse is important as it determines how it interacts with the turbulent channel and how well it can launch into the receiver optical fiber. To control the divergence to the desired amount, two optical lenses could be used at the input. Using different focal length lenses changes the shape of the spatial dimensions of the signal pulse and, depending on the alignment of these optical components, the divergence can be increased or mitigated, giving us total control over the beam. Additionally, if the input lenses are aligned accordingly, a 4F lens system can be configured to make use of Fourier optics, enabling spatial filtering of the optical signal pulse input.

When light is to enter the receiver optical fiber, an optical design that mimics the input is used – i.e., two lenses with the same focal lengths. The process of launching free-space light into an optical fiber is known as coupling. When coupling light into an optical fiber, the fiber core size, the fiber numerical aperture, and the lenses' numerical apertures must be considered. When focusing the free-space light signal, the spot size must closely match the fiber core size. The numerical aperture determines how much of the focused light enters the fiber; the higher the numerical aperture, the lighter the fiber accepts. For the fiber to accept most of the focused light, the numerical of the lens system must be less than numerical aperture of the fiber.

An external evaluator board is the signal processing center for the iSPY system. Housed on a PCB, the evaluator board housing a processing unit and transceiver, seamlessly integrating critical signal processing hardware. Per advisor suggestions, the evaluator board is an aftermarket component. Which evaluator board is used depends on the transceiver of choice and cost. The trade-offs associated with this choice are explored in the next chapter.

The evaluator board receives data and converts it to a functional optical signal for the transceiver; this optical signal modulates the laser diode on the transceiver, generating the necessary laser pulses for the FSO component of the project. For incoming laser pulses on the photodetector of the transceiver, the electrical signals generated by the photodetector is interpreted by the evaluator board and transmitted to the next electrical component of

the project – the display. This means that the signal is looped back into the evaluator board using two fiber optic cables acting as a sort of antennae, sending the data over the air between themselves back into the evaluator board to be processed. Essentially, the evaluator board acts as a bridge that connects the display to the Image Acquisition System by facilitating the necessary processing for FSO communications.

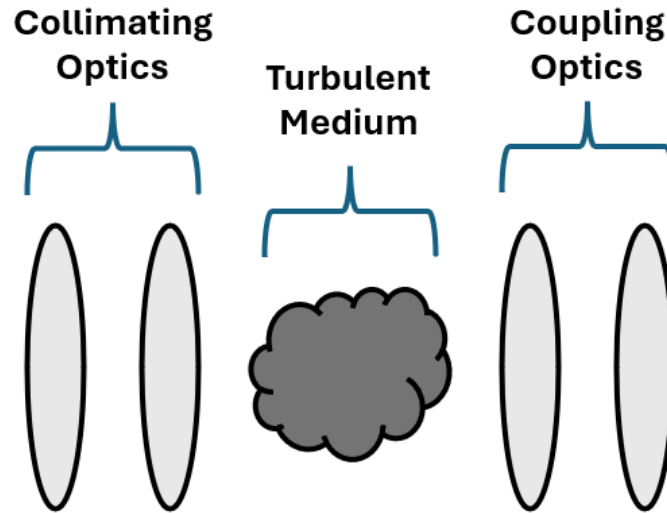


Figure 2.7. Optical schematic for FSO by Authors.

We would have to take extra care in the construction of the circuits connecting the components as we want to make sure that everything can receive enough power to operate but not consume excess power. The distribution of power must be carefully managed across the board to make sure the components are safe, and the important parts of the system won't get burnt out. Signal integrity is a major issue that would make or break the process since we want clear data that we can work with and get an output from.

2.5.1.3 Electronics Housing PCB

The goal of devkit board is to be the main brain for the system. This is where the bulk of our software is as we work with computer vision and video data. The kit is tasked with the intention to take in live video without audio, process that data using computer vision, counting the number of people in the cameras field of view, and prepare the transmission of data to the receiving end, that is the PCB. It should perform these tasks efficiently, balancing processing power, data integrity, and power consumption to ensure real-time operation. We would also like to be able to store some of the video feed into a memory cache to be able to have some playback if necessary.

The MCU on the devkit receives an uninterrupted continuous video feed in a digital format from the Image Acquisition system. With this uninterrupted digital signal, the MCU can break it down into individual frames where it can be analyzed further, including preprocessing if necessary. This could include adjusting the brightness or contrast of the image, if need be, so that the image can be cleaner for better computer vision performance.

Before processing any data and cleaning up the video, we trained a computer vision model to detect and count how many people are in the field of view of the external camera. This program analyzes the video feed and keeps a record of the number of present individuals, ensuring that it is identifying each unique person individually. Then the video is cleaned up more and the resulting data is compressed into a binary bitstream as well as a suitable format that enables efficient transmission. The Transmitter Board processes the video feed using a MCU into a binary sequence that can be broken up into 4 or 8 bits at a time creating this bitstream. A bitstream is a sequence of bits that are transferred from one device to another. Remember that all types of data are made up of bits, the most basic unit of information. As such, a bitstream applies to all data, be it an audio, image, or any other type of file [12].

Once the data has been processed and compressed properly, the devkit sends the data over free-space over to the transceiver module on the other side that is connected to PCB by an ethernet connection. This PCB is only responsible for decompressing the received data and relaying it to a display.

This PCB is connected directly to a power supply as well as power management circuitry that can produce enough power to the system while also ensuring that the computer vision and data transmission can take place while not using up too much power. Also, since the system is used in a way that needs to be operational for long periods of time, we utilized low-power operations with the MCU so that we can reduce the amount of power that is utilized to complete its functionality.

2.5.1.4 Display

An external monitor is used to display the captured and relayed real-time video feed. For the monitor to generate an image, it is connected to an external power source and a video-out port. The acceptable signal format for the monitor of choice is determined from the beginning processing unit discussed in the next section. The signal format varies between monitor models and video resolution.

2.5.2 Software

The software is divided into two separate sub-systems: object detection processing and digital signal processing. The software flowchart and separation of work is demonstrated below.

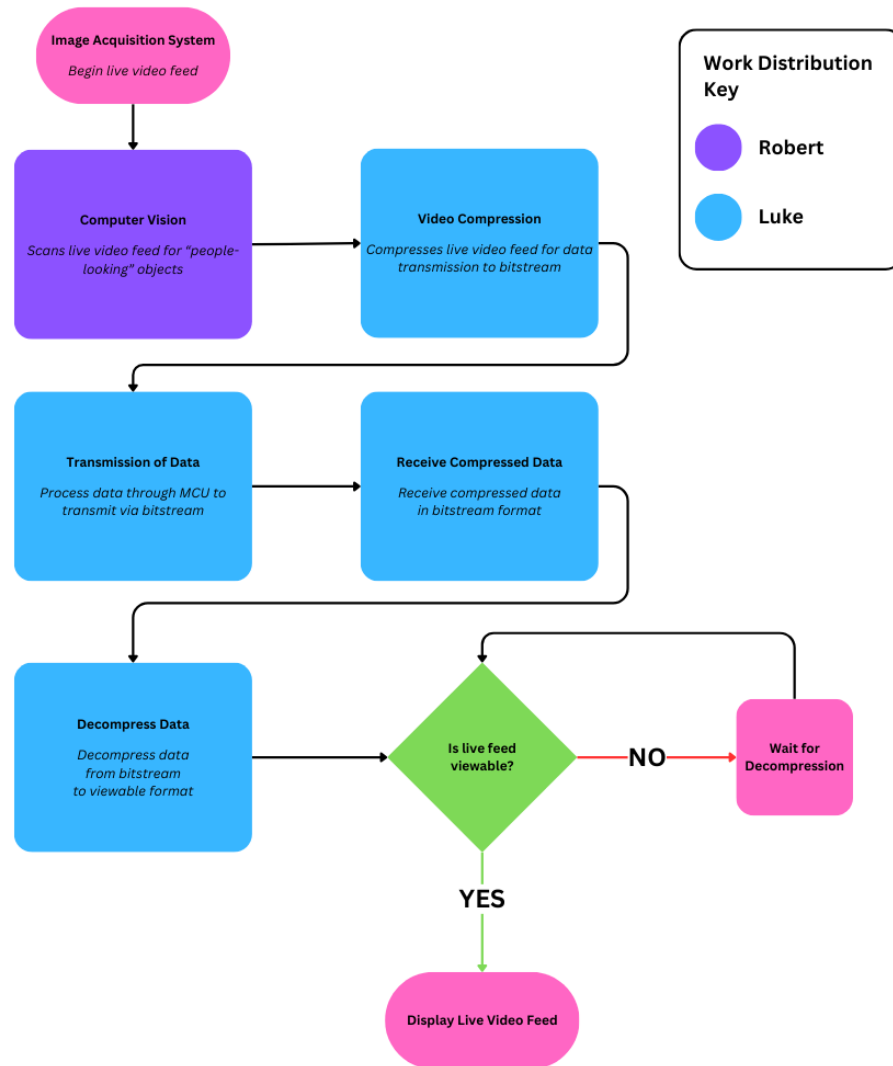


Figure 2.8. Software diagram by Authors

2.5.2.1 Object Detection Processing

The goal of the computer vision side of this project is to detect the number of people on the live video transmission at any time. This can be achieved with real-time processing of the live video feed to acquire an accuracy of 75% or more. A suitable approach to attain these kinds of results could be utilizing deep learning. Neural networking with many layers can give our computer vision the ability to understand more complex detection. Examples of this complex detection could include poor lighting, night vision, different angles of the body, and even clothing covering crucial parts of the body for detection. The program would detect the total number of people in the image and display this number on the output screen so that the viewer of the surveillance system knows exactly how many people are currently on the feed.

Neural networking is a process of machine learning that utilizes a “neural net” of many nodes. These nodes receive data and assign a “weight” to the data received. Each node passes down this data with the assigned weight to the next node and so on. There is a desired weight set for each node. If the node is under the desired weight, it does not transfer the data. If it is above the desired weight, it passes on the data. The weight of the data determines how important the information gathered is. Once the data has gone through the entire neural network, it arrives at the output layer. The output layer is where the result is displayed. [13]

In this case, the program is attempting to recognize human bodies in a live video feed. The program did not produce an accurate result on the first attempt. That is where the process of training comes in. Algorithms in OpenCV using neural networking is used and the best attempts from previous runs were remembered. The program remembers the previous data and use it as a comparison when going through the neural network in future runs. This process repeats until the program can accurately detect a human body 75% of the time.

2.5.2.2 Digital Signal Processing

The image acquisition system relays a live video feed processed by the computer vision algorithm which recognizes an object as a body. Once the process of recognition is done then another algorithm compresses the data into a bitstream which is sent to laser diode as a digital signal. The photodiode receives this digital signal as an optical signal and convert it to a digital signal. The MCU decodes the digital signal into a working bitstream for the display.

2.6 House of Quality

Efforts to carefully plan the project and closely consider customer needs and engineering requirements must be evaluated. A house of quality (HOQ) is a conceptual map that integrates planning, customer needs, and engineering requirements onto one chart. Additionally, it enables customer needs and engineering requirements to be directly related to each other in an organized manner. In this section, the team’s HOQ is shown in the figure below.

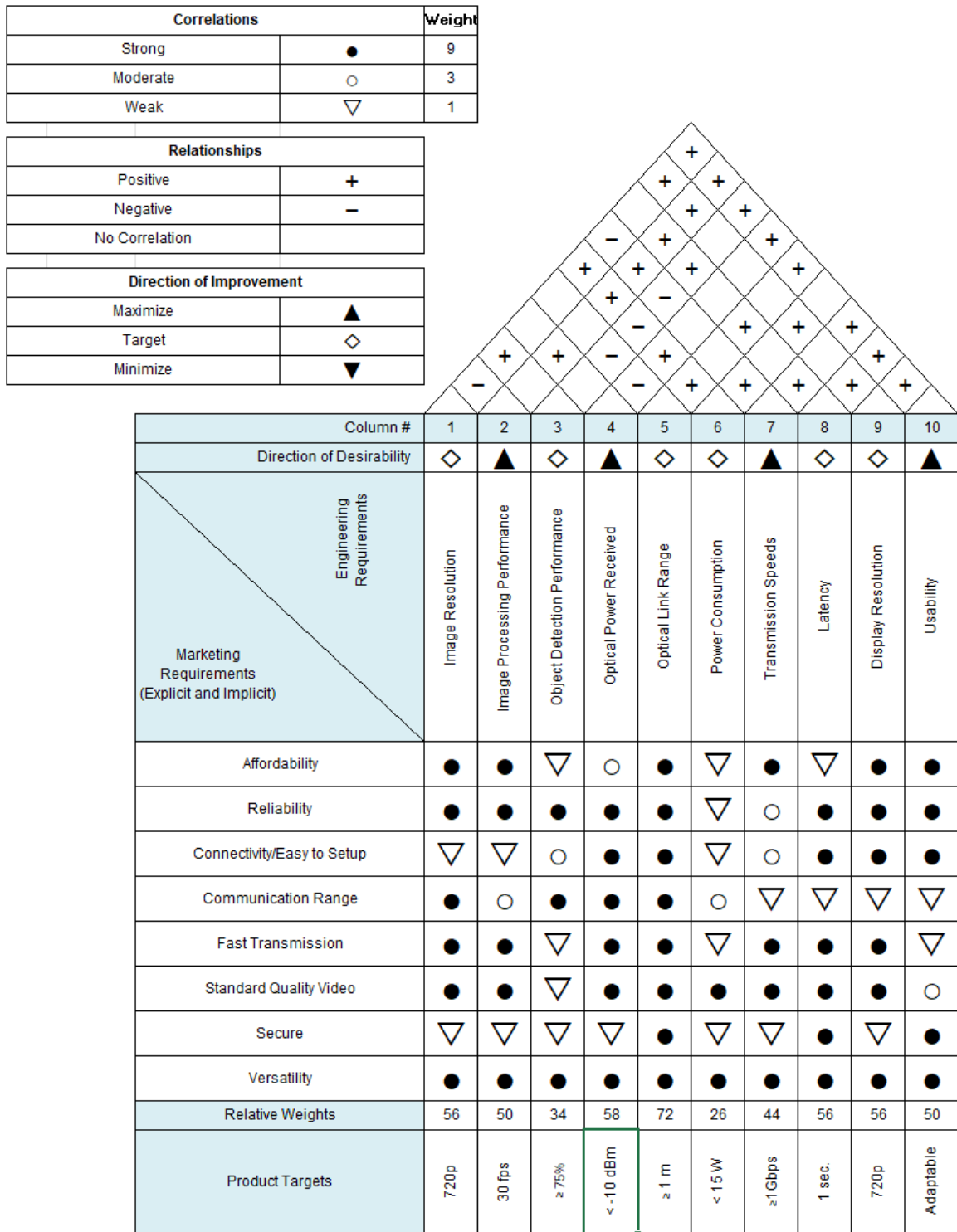


Figure 2.9. House of Quality

Chapter 3 – Research

The success of this project depends on the preparedness of each author and a knowledge-based decision-making process. This section presents the research sought by each author and decision-making process for various components embedded in project.

3.1 Imaging Sub-system

iSPY begins with the image acquisition system and in this section, we will explore the research that took place in order to design it. We separate this section into several sub-sections, starting with the sensor and ending with the lens choice, in later chapters we will talk about the integration of all of these parts.

3.1.1 Sensor Parameters

The first step on building iSPY was choosing the camera sensor. The camera sensor would dictate what kind of signal and video format we would be provided to the rest of the system.

Choosing the sensor is the first step in many optical design projects because there is a standard to be followed. They are usually not custom-made, and especially for a project like ours it would be impossible to design and fabricate one. However, there is freedom for lens choices. Therefore, we followed the steps of other optical engineers and decided to work around a sensor. [14]

When choosing a camera sensor one of the most important parameters is its size. There are so many options, and each choice comes with a tradeoff. Understanding what is important for our system is key. To make sure all parameters were considered ChatGPT was used to point out the main factors to look out for when choosing a sensor. The parameters were: sensor size, resolution, frame rate, dynamic range, shutter type, data interface compatible with the system. I also integrated other parameters into the discussion like depth of field, F-number, and image circle.

Sensor format size is the starting point when choosing a sensor. It dictates nearly all operational metrics of the imaging system [15]. An equation that shows this strong relationship between sensor size and the parameters is:

$$\text{FOV (in degrees)} = 2 \tan^{-1}\left(\frac{w/2}{f}\right) \text{ [Eq. 3.1]}$$

Where f is the focal length and w is the sensor width [16].

For our system (and surveillance systems in general), a wide field of view is crucial. Field of view (FOV) is an angle measurement of how much of a scene is visible to a device or

individual. It is the parameter that lets the viewer know how much they can see. For this system, it is a parameter based solely in one frame from a single device (only one camera).

With equation 3.1 in mind, we can see that a larger sensor increases the field of view. Other benefits of a larger sensor are that it decreases diffraction, gathers more light, in other words, it acquires more information increasing the quality of the electrical signal that is converted by the pixels and having better performance in lowlight environments [17] [18].

The downside of a large sensor is its size, with a larger sensor we also need larger and consequently heavier lenses. The price also increases dramatically [17] [19]. However, an important feature is that for sensors designated for video recording/streaming increasing the size of the sensor does not add too much to the system, it is not a clear linear relationship like it is for photography. A larger sensor should only be considered in specific cases like extremely shallow or deep depth of field and shooting low-light performance [17].

If the aperture (opening) is to remain the same, something we have to take into consideration because in this project that is the case, the depth of field (range of distances where the images are clear) decreases as the size of the sensor increases. The way to counterattack this effect would be to get closer to the subject or to increase the focal length to fill the frame. However, both options are nonviable in our system. The camera does not move closer to a subject and the focal length does not change [19]. So even though larger sensors provides greater detail, depth of field is not something a surveillance system can tradeoff.

A stop is a hard physical aperture that limits the amount of light into an optical system. An example of a stop is an iris; by opening and closing it one can determine how much light passes onto the system. The aperture stop changes light levels and it is also used to help control optical aberrations [20].

Another crucial parameter equation 3.1 displays are the relationship between focal length and field of view. We will go into more detail about focal length and effective focal length of the system in the second part of this second where we explore optical parameters, but even now we see that a short focal length increases our field of view.

Resolution can be defined as “the dimension of the finest spatial detail in the object that can be discerned in the acquired image” [21]. In other words, we decrease the distance between two points until we cannot resolve them separately anymore. Resolution is the ability to tell points/objects apart, to resolve them.

Many elements play a role in the resolution of a system. We separated these differences in two, the resolution of the sensor and the resolution of the lens system will be explored later in this section.

It is common knowledge that the pixel size and quantity define the resolution of a sensor. Megapixels is a measurement of how many millions of pixels a sensor has. The sensor

sizes are standard, but not all sensor sizes have the same number of pixels. A large number of megapixels increases the resolution.

High quality surveillance systems have 4K sensors (four times as many pixels as a 1080p sensor). Cameras on a budget will oscillate between 1080p and 2K. 1080p sensors have a resolution of 1920x1080 pixels. 2K sensors have a resolution of 2560x1440 pixels, and 4K sensors have a resolution of 3840x2160 pixels [22] [23].

But when exploring resolution there is much more to it. The highest frequency a sensor can resolve is called the Nyquist frequency. It can be defined as two pixels or one line pair (2 pixels=1 cycle=1 line pair, or lp). In the industry, a common way to approach the sensor resolution (ξ_{Sensor}), also referred to as the system's image space resolution ($\xi_{\text{Image Space}}$), is “by multiplying the pixel size (s), in microns, by 2 (to create a pair), and dividing that into 1000 to convert to mm.” [24]

$$\xi_{\text{Sensor}} = \xi_{\text{Image Space}} = \left(\frac{1 \text{ lp}}{2 \times s}\right) \times \left(\frac{1000 \mu\text{m}}{1 \text{ mm}}\right) \text{ [Eq. 3.2]}$$

We can also determine the minimum resolvable spot in a scene by taking the ratio between the sensor size (horizontal, vertical, or diagonal) and field of view (FOV). This relationship is also called system magnification [24].

$$m = \frac{H}{FOV} \text{ [Eq.3.3]}$$

By determining the imaging space resolution and the system magnification we can make the lens choice more convenient.

Choosing between the sensor resolution is also a matter of bandwidth. Higher resolution means more data transfer, and it requires faster speed. Another detail to consider is the frame rate the sensor enables. The rate of frames per second (FPS) indicates how many images are taken in one second. More frames per second results in a smoother and higher quality video. Usually, the frame rate found in security cameras varies from 30 to 60 FPS, where the high end of this range is used for traffic monitoring [25].

The tradeoff between a higher number of frames per second is that it also increases the bandwidth and storage requirements. Since the goal with this surveillance system is not to track fast moving objects but people inside a room, a 60 FPS is not necessary. Just as important as the sensor size, the type of imaging sensor dictates the system: charge-coupled device (CCD) or active-pixel sensor (CMOS). When choosing between the two through the lenses of this project, a CMOS sensor aligns more effectively with the goals we plan to achieve. A CMOS sensor offers a higher resolution, low power consumption and the sensors are more cost effective than the CCD ones [26].

The shutter of a sensor controls how long it is exposed to light to record a frame. This period of exposure is called the shutter speed [27]. When this speed is not as fast as we would like blur occurs, therefore, even in well-lit places a shutter speed of at least 1/30s is

recommended. With the basic and advanced goals in mind, we made sure this recommendation is met. However, for the stretch goals this speed would need to increase dramatically. In lowlight environments, shutter speed should be around 1/500s [28].

There are two modes for shutters, global and rolling. In global shutters the whole sensor array is exposed simultaneously, a good way to picture it is to image a flash. Rolling shutters on the other hand, have different array lines being exposed at different times, line-by-line, like a ‘wave sweep’.

Rolling shutters have a disadvantage, they are known for spatial distortion. Because the array is exposed line by line, objects that move extremely fast are not captured accurately if the speed at which the lines (of the sensor) are exposed is lower than the speed of the object moving across the sensor. However, rolling shutters are commonly used in industry and the majority of CMOS cameras have them. Since we are not tracking fast-moving objects, they are enough for our purpose [29] [30].

There were two choices between the sensor for the image acquisition system, the sensor module and the camera module. They have similar functions; however, the sensor module on its own has a limited image processing capacity, making the host side (the PCB with the MCU it is connected to) in charge of both pre and post processing. Also, if the sensor were to change the MCU would have to be reset since the program is unique for each sensor.

On the other hand, the camera module already comes with pre-processing capacity and the host side can take the already clean image and focus on other aspects such as object recognition. It is also more flexible since it does not solely rely on the host side for the processing of images [31].

For this project we used computer vision to detect how many people are at a given place, therefore, our choice is of a camera module instead of a sensor module. The camera module enables us to focus our efforts on the usage of the images since it obtains and processes it for us. We were also able to have higher quality video because we can have two parts of the system (the module and the host) working on polishing the frames.

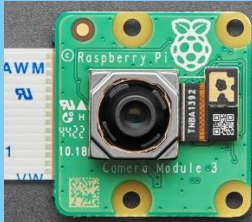
A method by which data is transferred from one device to another is called data interface. Interfaces are standardized to ensure that electronic components can smoothly work in conjunction. When talking about cameras, the data interface refers to how the frames (image or video) are obtained and transferred to a processing unit [32].

Since the interface is the connection between the camera sensor and the host it dictates the overall system performance and speed. The interface supports a two-way communication where the sensor sends images and videos (frames) to the host and the host controls the camera sensor. Choosing a suitable interface is crucial to meet bandwidth and resolution requirements [33]. The most common interfaces in the industry for machine vision are FireWire, Gig-E Vision, USB3 Vision, Camera Link and MIPI.

3.1.2 Comparison of Camera Modules

After analyzing all the parameters from the previous sections, three camera modules were selected for the final cut. They can be seen on the table above. I went ahead and named them camera module “A”, “B” and “C” to simplify the comparison when referring to each one of them.

Table 3.1 Comparison table of different camera modules.

				
Camera Module A				
Model	Frame rate (fps)	Sensor size	Interface	Shutter Type
daA3840-45uc	45	1/1.8”	USB 3.0	Rolling
Brand	Pixel size (H x V)	Resolution	Weight	Price
Basler	2.0 x 2.0 μm	8.3 MP	< 15g	\$199
				
Camera Module B				
Model	Frame rate (fps)	Sensor size	Interface	Shutter Type
Pi Camera Module 3	30	1/2.43”	MIPI CSI-2	Rolling
Brand	Pixel size (H x V)	Resolution	Weight	Price
Raspberry Pi	1.4 x 1.4 μm	11.9 MP	4g	\$25
				
Camera Module C				
Model	Frame rate (fps)	Sensor size	Interface	Shutter Type
IMX519	30	1/2.53”	-	Rolling
Brand	Pixel size (H x V)	Resolution	Weight	Price
ArduCam	1.22 x 1.22 μm	16 MP	8.5g	\$37.99

As noted above, camera module A is the most complete one. It already comes mounted (the plastic surface that surrounds the sensor); it has the highest frame rate and sensor size, and it is equipped with the newest interface. However, it is also the most expensive sensor. One of our goals is to keep the project low cost offering the same, or better quality, therefore, sensor A was not chosen.

The decision then was between camera module B and C. They are similar to each other, so when choosing between the two we went for the one with the largest sensor and of course, the cheapest price. And that is why for iSPY we used camera module B, Raspberry Pi's Camera Module 3.

The brand, Raspberry Pi, offers great documentation for their components which has made understanding the sensor a lot better. The specs table for this camera are attached to the appendix. The camera module comes with a lens; however, it does not fit our requirements so we replaced it with a different lens assembly that improved our field of view.

The camera module chosen uses the interface MIPI CSI 2, one of the most popular interfaces for computer vision. It is a reliable option because of its bandwidth capacities and price [34].

3.1.3 The Optical Parameters

One of the most critical optical parameters is the focal length. The simplest definition of focal length is the distance between a lens and its focal point. This relationship can be described by the equation:

$$\frac{1}{f} = \frac{1}{d_o} + \frac{1}{d_i} \text{ [Eq. 3.4]}$$

Where f is the focal length, d_o is the object distance and d_i is the image distance.

However, for a lens system as complicated as ours this relationship is not so straightforward. Our focal length is actually be called effective focal length, since it is a combination of different focal lengths in our system. Fortunately, the software that is used for the simulation of our system, Zemax, helped us to keep our desired effective focal length in check as we add new surfaces and optimize the design.

Equation 3.1 shows that a short focal length is needed for a wide field of view. It is no surprise then that surveillance cameras have focal lengths of just a couple of millimeters. The optical system for iSPY has a fixed focal length and fixed focus. In other words, the camera does not move around; it is always "observing" that same part of the room/of the scene. Even if objects/people come close to it the focus does not adjust as our main interest is to count how many people we can see and not facial recognition. The use of fixed focal length lenses is common when working with machine vision, another aspect of iSPY [35]. Our surveillance system, at least to meet the basic and advanced goals, it is designed to work on environments where there is not lighting variations (i.e. inside buildings such as

lobbies and hallways), and the fixed focal length design is advantageous because it gives a stable image quality [36].

The three ways that a focal length can be changed are by moving the lens system, replacing the lens system and/or by changing the sensor size. All these options are not feasible for iSPY [10]. We used these constraints in focal length and focus to simplify our system. Our lens design was already complex and adding features like autofocus would increase the cost and time needed to complete the project.

The F-number ($f/\#$) or f-stop indicates how much light passes through an optical system. It is the ratio between the focal length and aperture diameter. Low F-numbers are related to large apertures, more light enters through a lens and the system is great in low light conditions. Lenses with a low F-number are called fast lenses. High F-numbers, on the other hand, describe small apertures, where less light comes into the system and are best used in well-lit places. They also have a greater depth of field than low F-numbers. Lenses with a high F-number are called slow lenses.

Since the security system we want to build is to be used during the daytime and inside well-lit places, lenses with low F-number are not needed. And this also decreases the cost of our system since moderate to high F-number lenses are cheaper.

Moderate F-number lenses were previously mentioned because it is a trade-off that needs to take place to decrease diffraction. A high F-number is not ideal. When talking about F-number, another parameter that comes in question is the numerical aperture, NA. The two are inversely related for infinite conjugate systems by equation 3.2.

$$NA = \frac{1}{2 \times f/\#} \quad [\text{Eq. 3.5}]$$

NA is used when we refer to the cone of acceptance of a system. It is the range of angles that are allowed, “a measure of the collection efficiency of an optical system” defined as:

$$NA = n \sin\theta \quad [\text{Eq. 3.6}]$$

Where θ is the half angle of acceptance and n is the refractive index of the lens [37].

Diffraction is the bend of light (waves) when passing through narrow apertures or around objects. When diffraction results from a finite circular aperture, an airy disc pattern is the outcome [38]. This airy disc is known as the diffraction limit of the system, and it is always present, no matter how perfect the lenses are, “the ultimate limit of resolution of an optical system is set by diffraction” [39] [40].

To create a successful system, we should always take into account the diffraction limited spot. When designing our lens system using the optical design software, great care is taken when comparing the current design to the airy disc. We always want to make sure our resolution is not too different from it.

Lens compatibility is crucial when matching a lens system to a sensor. The size of the lens needs to be greater than or equal to the size of the sensor to totally cover the image circle.

When the sensor is larger than the lens, part of the image is cut-off making dark circles around the edge of the image visible, the optical vignetting [41] [42]. For surveillance systems, smaller sensor format sizes are the most popular. They have a smaller lens image circle making the lens and the system more affordable [42].

Optical vignetting is one of the several aberrations that can take place in an optical system. One of the strategies to reduce its effects is to use aspheric lenses; however, this method requires a larger budget since these lenses are expensive. Another procedure to decrease optical vignetting is to purposely limit the field of view, and this is what we did in our design. The field of view of our camera was deliberately limited due to cost, size and weight of the final product [16].

3.1.4 Image Acquisition System: The Optical Design

The optical design is an extensive one, from choosing a sensor, to finding the lenses with the correct diameter and focal length. It was a process that required a lot of time, trials, and tradeoffs. Wide field of view cameras are complex. When one wants to receive light from larger angles aberrations greatly increase and to minimize their effects several techniques need to be used.

The first step to design a lens system for our camera was to research what is done in the industry, what others with more experience and time have put together. This research showed us that even standard designs are complex and require aspheric lenses. Image 3.1 displays a lens system for a camera of 100 degrees field of view.

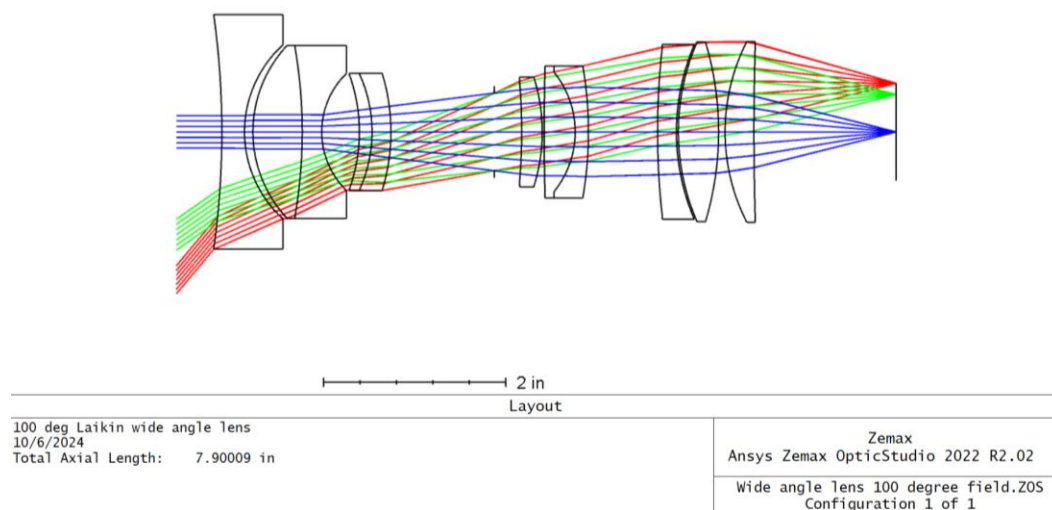


Figure 3.1. Zemax layout of a 100-degree field of view system. This layout can be found under ‘Samples’ inside the Zemax folder when the software is downloaded.

After an extensive investigation on wide-field-of-view cameras used in the medical field, and open patents for phone cameras [11] [43], and seeing the complexity of such systems, I was oriented by supervisors to go for a simpler design, the ones previously mentioned are above the scope of a senior design project, they require years of dedicated work – something like a Master’s thesis. The designs fall under the constraints that will be explored in Chapter 4.

The idea was to find a framework for our design and not start from zero, a common practice in industry. And that is when I stumbled upon the new SPIE’s book, “Designing Optics Using Zemax” by Donald C. O’Shea and Julie L. Bentley. In this book, the writers explore the lens design software, Zemax, with the readers and give them a chance to design a system at the end. This system, that can be found in chapter 12, is the skeleton for the optical design of iSPY.

Following the framework laid by the authors of the book I was able to achieve a similar system with the desired focal length and consequently, the required field of view. However, just like any other simulation some of its approaches involve custom-made lenses, and/or are made for a “perfect scenario” and ignore the feasibility of the design. Therefore, it is up to the optical designer to translate this optimum/perfect design to something tangible that can be recreated.

Both designs, the original one and the one I worked on following the steps from the book [Appendix G] ran into an issue in its early states. The optimal thickness of the lenses was negative, which of course, cannot happen. This issue was fixed by configuring the software and letting it know the limits – setting up boundaries.

Since the book’s intent was to design a lens system for the sole purpose of proving a concept the writers had the freedom to use any type of surface combinations, basically creating their own lens. When following their steps, I also use the surface combinations necessary to make my system work; however, since my plan is to physically build this lens system, to bring it to life, I knew the design would have to be modified.

Having the scope of this project in mind, off-the-shelf lenses are the cheapest and less risky option. Fortunately, the main websites for purchasing lenses from Edmund Optics and Thorlabs have Zemax files for most of their lenses, which facilitates the transition from simulation to a feasible physical system.

3.1.5 Lens Background

The most popular lenses are the standard convex and concave ones. Convex lenses are positive, they can be biconvex (the two surfaces have a positive radius of curvature) or plano-convex (one surface is flat while the other has a positive curvature). Convex lenses are used to focus/converge light onto a specific spot, they also are used to decrease focal lengths. Concave lenses are negative lenses, they can be biconcave (the two surfaces have a negative radius of curvature) or plano-concave (one surface is flat while the other has a

negative curvature). Concave lenses are used to diverge light, they also increase focal lengths.

The physics of lens refraction is the key to understanding any optical system. These standards lenses have been around for centuries, they are the bread and butter of the world of optics. However, the need for more complex systems pushed scientists to come up with ideas such as new surfaces configurations, and the use of different materials to make and coat the lenses.

The standard lenses can also be called singlets. They are the simplest form of a lens. By combining singlets, we can get doublets (cementing two singlets together) or triplets (cementing three singlets together). The reason for such new configurations is to gain more control over the system and refine the imaging process.

An example of a doublet is the achromatic doublet, which is formed by combining a positive and negative lens. The positive lens is called the crown, and for this doublet to work as intended, the crown needs to be made of a material with low refractive index compared to the one material from the flint, the negative lens [44].

The Abbe number is a measure of dispersion when passing through a lens. A high Abbe number is an indicator that different wavelengths have similar index of refraction in a medium, in other words, lower dispersion. In achromatic doublets the crown has a high abbe number while the flint has a low one, and by combining the two lenses chromatic aberration is decreased [45]. We can also use a triplet achromatic or aspherized (more on aspheric lenses in following paragraphs) achromatic for color correction in our system [46].

Convex lenses bend outwardly, concave lenses bend inwardly; however, sometimes a combination of both bends is necessary and that is why meniscus lenses were introduced. Meniscus lenses have curvatures for both of their surfaces, one outwardly and the other inwardly, they can also be called convex-concave lenses. Positive meniscuses are used to reduce the beam's focal spot, while negative meniscuses are used on applications that require beam expansion [47], [48], [49], [50].

A special type of lens is the aspheric lens since it varies its curvature unlike regular lenses. Aspheric lenses are used to reduce aberrations (especially spherical aberrations) and decrease the size of the system since a few aspheric surfaces work just as well if not better than several spheric surfaces when correcting aberrations [51]. Another unique lens is the Fresnel one. Fresnel lenses are formed by etching a series of concentric grooves into plastic. They are used in diverse applications, especially the ones that involve light gathering [52].

The lens design for a wide field of view system is complex, it usually requires several surfaces to decrease aberrations and focal length. While researching the designs used in industry, I found that aspheric lenses are the most popular lens type for such a system, which makes sense since they are extremely efficient in decreasing aberrations. They are custom-made for each application.

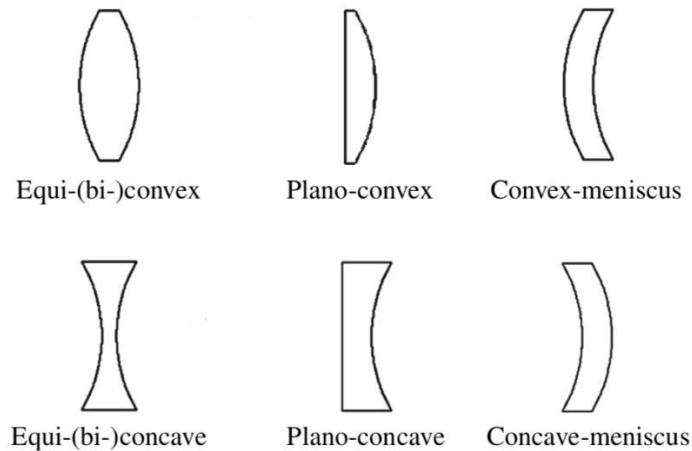


Figure 3.2. A comparison between the standard lenses and the meniscuses found in reference 49.

A widely known method to increase the field of view relies solely on standard lenses, the inverted telephoto lens system. The regular telephoto lens system is made of a “group” of positive lenses followed by another “group” of negative lenses. This configuration enables the focal length to be longer than the physical length of the system. In the inverted version the negative lens or lenses come first than the positive ones, and it is used to widen the field of view [53], [54].

3.1.6 Lens Material

The most common glass types to make lenses are N-BK7, N-SF5 and N-SF11. N-BK7 glass is known for its exceptional optical properties, purity, low-cost, and constant refractive index throughout its surface [55].

N-SF5 is popular due to its high refractive index, and low Abbe number. Due to its high dispersion, it is usually paired with another material with a higher Abbe number to avoid chromatic aberrations, an example of that is the achromatic doublet. N-SF5 is used in applications where high precision is required [56].

N-SF11 is popular due to its high resistance in harsh environments. Lenses with this type of glass are used in systems where the bending of light needs to be significant such as high magnification microscopes [57], [56].

3.1.7 Lens Coating

Another important characteristic of a lens is its coating. Optical coating is the action of depositing a thin layer of materials onto optical elements such as mirrors, lenses, and prisms to change the way light is reflected and/or transmitted. A commonly used optical coating is the antireflection coating in eyeglasses [58], [59].

When choosing a lens, it is important to take into consideration if the lens is coated or uncoated, and what type of coating is being used if any. Depending on the wavelength of the light source different types of coating are used. Another detail is that coated lenses tend to be more expensive than uncoated ones.

For our project, we decided to use lenses with an AR coating. According to a lens manufacturer, uncoated lenses lose 4% of light at each optical surface due to reflections, our system has several optical surfaces and such loss is undesirable [60]. Since we are dealing with the visible part of the spectrum, the wavelength range of our lenses is from 400-700nm. And the material used for the coating is MgF_2 . Magnesium fluoride is the most common compound used for anti-reflection coating. It is known for its durability and cost-effectiveness [61].

In Figure 3.3 we can see the comparison between a coated and uncoated lens. The uncoated lens (Graph A) transmission band is broad, enabling the transmission of a large range of wavelengths, while the coated lens (Graph B) reflects an average of 0.5% along the wavelength range of 400-700nm, in other words, the transmission in this range is higher and wavelengths outside of the visible spectrum are blocked.

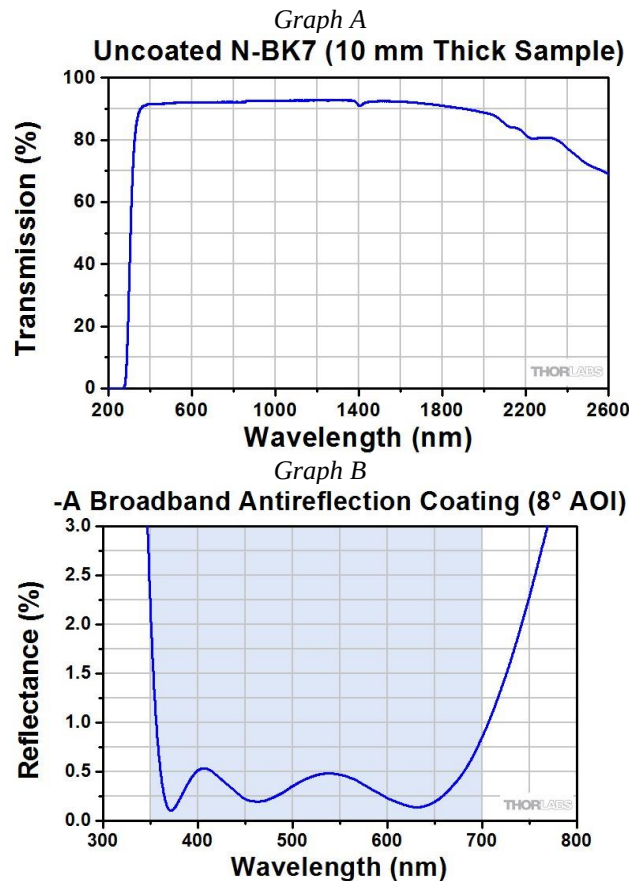


Figure 3.3 A comparison between uncoated [Graph A] and coated [Graph B] lenses from reference 60.

3.1.8 Optical Aberrations

In theory, a lens will always perfectly image an object. However, that is not the case. As the rays move away from the axis, we see a failure to image it, and this distinction between the ideal scenario and the result of actual lenses are a measure of optical errors, also called aberrations. As defined on the Edmund Optics website, “optical aberrations are deviations from a perfect, mathematical model” [62].

When talking about aberrations, we usually discuss first and third order aberrations. Higher order aberrations exist but they are too complex and trying to decrease them improves the image quality only slightly [62].

The first order aberrations are tilt and defocus. Tilt as the name suggests, deals with a shift from the reference wavefront. The actual image is formed at an angle. Defocus happens when the image is formed in the wrong plane [62].

The third order aberrations are spherical aberration, coma, astigmatism, Petzval curvature, and distortion. Spherical aberration happens when light does not focus on the same plane, a mismatch when converging light. This type of aberration blurs the final image, and to decrease we can use aspheric lenses, reduce the entrance pupil of the system so the rays at the edge are cut off, add defocus, use lenses with a higher refractive index or use lens splitting – increase the number of surfaces [62] [14].

Imaging depends on ray heights, and when the rays are coming from far away these heights change – they do not hit the spot for perfect imaging. This change in ray height and magnification with respect to the location of the image causes coma [63], [62]. A way to decrease coma is to move the aperture spot to the center of the system.

Astigmatism refers to the aberration caused when the lens does not perform the same way in the sagittal (horizontal) and tangential (vertical) direction, two focus points are created. This aberration can be decreased by defocusing the system and moving the aperture stop.

Field curvature is an aberration that occurs when imaging onto flat planes (screens, sensors) rather than an ideal curved one, the paraboloidal surface known as the Petzval surface. Its effects can be minimized using spaced doublets [62].

Distortion aberration does not cause blur, but it changes the magnification of the image, it produces images that looked stretched or compressed. Positive distortion is called a barrel distortion, and a negative distortion is called a pincushion distortion. In lens systems designed for wide field of view distortion is the main aberration. The effects of distortion can be decreased by changing the location of the aperture stop [14], [64].

Another aberration that we should take into account especially for this system is chromatic aberration. Chromatic aberration happens when an optical system fails to converge different wavelengths on the same axis, it causes color inconsistency and distortion. To reduce this type of aberration a combination of positive and negative lenses in an

achromatic doublet is used [14]. Achromatic doublets are used to eliminate chromatic aberrations; these types of lenses work by focusing a parallel input beam to a perfect “point” [44].

3.1.9 Lens Selection

The selection of the lenses is one of the most crucial steps in this project, and for this reason it also becomes one of the most time-consuming tasks. Several iterations of this lens system have taken place, and adjustments were continued made until the end of the semester to obtain the highest resolution for our image acquisition system.

On top of considering the details previously mentioned such as lens material, coating, its effect on the overall quality of the image (aberration), we also had to take into account price, and the availability of the lenses. Unfortunately, not all the lenses are in stock and since time is a huge constraint in this project, we had to work around that.

Another limitation was that using off-the-shelf lenses required us to work with already set parameters. We do not have a say on what focal length and diameter we want, we must work with what is available to us.

In the latest iteration of the lens system for iSpy, an aspheric and a convex plano lenses are used. The aspheric is made of D-ZK3 and the convex plano is made of N-SF11. Both lenses are coated so only wavelengths in the visible spectrum can be transmitted in our system. This iteration of the lens design gives us the highest field of view (shortest focal length) and the best image.

3.1.10 System Performance

One of the ways to evaluate the imaging system’s performance is through its Modulation Transfer Function (MTF). Modulation Transfer Function measures the image contrast as a function of spatial frequency (lines per mm) [65].

Contrast has to do with the difference between gray values, where 1 is white and 0 is black. The resolution limit is reached when we cannot tell apart black and white anymore at a given frequency. To measure the contrast of a system a sinusoidal pattern with black and white strips is presented, then the spatial frequency (the distance between the stripes) and the contrast are changed, a visual of this process is displayed in figure 3.4 [66].

MTF is a parameter that lets us know how well our system does in transferring details from the object plane to the image plane. It incorporates contrast and resolution (defined in section 3.1.1) [Ref. Introduction to Modulation Transfer Function]. The optical design software, Zemax, provides users with a graph of the MTF for their lens systems.

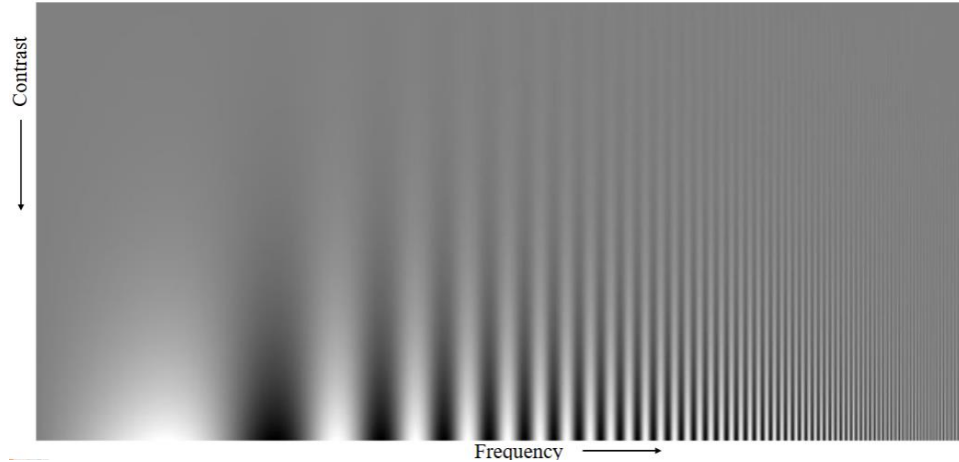


Figure 3.4. A demonstration of how contrast is measured. In the x-axis direction, the spatial frequency increases to the right; in the y-axis direction, the contrast increases downwards. Image from Reference 66.

As figure 3.5 shows, the MTF decreases as the spatial frequency increases, which makes sense, as the separation between lines decreases it is more difficult to resolve the details.

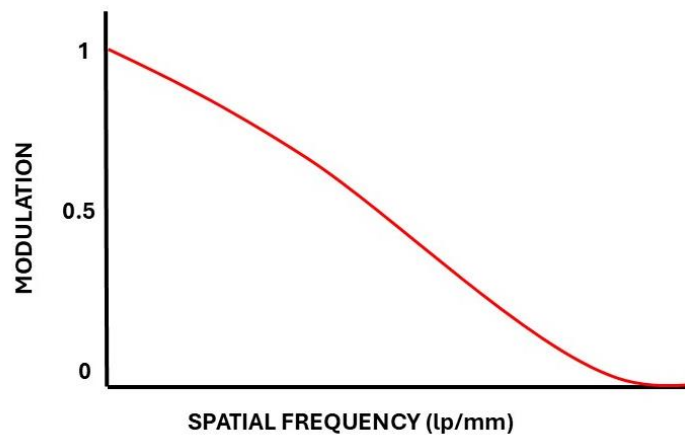


Figure 3.5. A simple demonstration of the MTF graph given by Zemax. Graph made by Authors.

3.2 Computer Vision

The computer vision aspect of iSPY is the most important part of our system. Computer vision uses mainly deep learning algorithms such as convolutional neural networks (CNN), recurrent neural networks (RNN) and natural language processing (NLP) just to mention a few.[67], [68] Convolutional neural network is used for images whereas recurrent neural networks are for video.[67] Both handle their own things to process in a very similar way.

When it comes to CNN, they use a method that breaks down the image into pixels and from the pixelated form of the image then gives a label and tag.[67] CNN also uses a process that utilizes a kernel over the image creating a sort mapping of features or edges.[68] RNN uses this same process though it focuses on frames of a video and the progression of those frames.

The use of both algorithms set up the overall functionality and process of computer vision. There is one other thing which is Deep Neural Network (DNN). If deep learning and computer vision combined into a hybrid process that could send the detected object to a DNN then the number of resources when computing is decreased.[68] The process mentioned is more common today. Even though there are other things out there like 2DCNN and 3D CNN that are out of scope for our project.

The use of neural networks has advanced over time to become more compact into simpler coding commands for computer vision purposes like image classification, object detection, human body detection and even tracking people or objects.[67] The coding libraries that utilize the simple coding commands that can handle computer vision vary from simple to more complex. The coding libraries were created by different programming languages. All these programming languages have their own library to handle computer vision tasks. These programming languages include C++, Java and Python.

3.2.1 C++ Computer Vision Library

The C++ library for computer vision is Convolution Architecture for Fast Feature Embedding (CAFFE) which was created by a research group from University of California, Berkley.[69], [70] The library is an open-source framework that can handle multimedia, vision and speech like tasks. Some features that CAFFE has are object detection, object classification and learning semantic features.[69] CAFFE uses something known as CUDA which was developed by NIVIDA for GPU computations.[69] CAFFE is modular and can be used with other programming languages such as Python, MATLAB and NumPy.[69]

The library is very useful for certain things in computer vision however the library does have its own pros and cons. Somethings that are useful about CAFFE are its high performance for fast feature embedding and it is supports the use of many deep learning architectures.[70] However there are a few downfalls which are more manual coding for certain tasks and limited to only a few features that computer vision can do.[70] CAFFE is only one out of many libraries used for handling computer vision like tasks.

3.2.2 Java Computer Vision Library

The Java based library is known as BoofCV which is an open-source library as well.[70], [71] BoofCV has various features ranging from image processing to geometric vision. It can render features in 2D and 3D spaces.[70], [71] BoofCV was made from scratch for real time applications such as image association, mosaics, visual odometry and 3D structure estimation.[67] The library can also stabilize images. Even though this library has more in the way of computer vision aspects and can handle more tasks it has its flaws. The only flaw that can be determined and seen is slower performance in certain low-level operations.[70] Java based library BoofCV is faster than most other libraries when it comes to certain operation handling.

3.2.3 Python Computer Vision Libraries

Python has various computer vision libraries that are used when it comes to detection, recognition and image segmentation. The libraries utilize pre-built algorithms to achieve different performance factors whether it be accuracy to efficiency or both at once. In this section I examined three different Python computer vision libraries and discussed briefly the choice made for the library I chose to use for iSPY.

3.2.3.1 TensorFlow

TensorFlow is a library that was developed by a research group at Google.[72] It is one of the many libraries in Python that can do computer vision duties. TensorFlow supports many other programming languages and platforms, not just Python or PyTorch. It supports developing models on mobile and embedded devices opening up to more systems.[72] The TensorFlow library for computer vision is simple to use for anyone that needs to use it. It has a few features which are object identification, picture categorization and facial recognition to name a few.[70] TensorFlow has regular updates with new features and enhancements offering more to the user.[70] It has strong features with effective operation though it consumes a lot of resources.[70] There are other components and versions of TensorFlow which to name one is TensorFlow Lite. Although it relies on a bunch of resources it is a very great toolkit for computer vision aspects.

3.2.3.2 SimpleCV

SimpleCV is a collection of libraries and software used for vision applications.[73] It allows you to work with images and video streams from different sources. Although this library can only be used on other platforms like Mac, Windows and Linux, which means it can't be used with other programming languages.[73] SimpleCV has similar features as the other libraries mentioned though can-do object detection, image segmentation and image arithmetic.[72] It can also deal with detecting and recognition in a video. It is more concise and simpler to use based on its functionality. SimpleCV includes libraries like OpenCV without the knowledge of image processing. It is very useful for a beginner in computer vision and understanding how to detect items in an image or video depending on the system. SimpleCV is limited when it comes to features than OpenCV. It also is more used for prototyping.[72]

3.2.3.3 OpenCV

OpenCV is more popular among those that work with computer vision applications. It is very versatile in the number of functionalities it can do and handle. Unlike its predecessor it can deal with more complex computer vision imaging and video aspects. OpenCV has been used by beginners and more experienced individuals for many projects like detection of objects in each image input into the program. One of the only libraries that have over 47 thousand people that use it and exceed 18 million downloads.[70], [74] It was also written

and developed in C++ programming language.[70], [74] Although it is commonly used in Python by the users which is OpenCV-Python.[75] OpenCV-Python uses the Python API and combines Python with OpenCV C++ API.[75]

OpenCV was created by Intel in 1999 and was first released in 2000.[75] The library system has over 2500 optimized algorithms which is more than any other library system when it comes to computer vision.[70], [72], [74] Stands the reason as to why OpenCV is not limited in features as other libraries. OpenCV also has many versions that have added more features and functionalities over time. Some of the features that OpenCV can do are human action classification in videos, object motion tracking, eye movement tracking and 3D model extraction just to name a few of the features.[70], [74] There are many more features as denoted before.

However, this library is not as easy to use and does take some time to learn.[70] Although the benefits out way these issues since this library is so expensive. Earlier I mentioned OpenCV-Python and how it uses a combination of certain things. OpenCV-Python is a library with Python bindings meaning we can utilize Python wrappers with the original OpenCV C++ implementation for computer vision applications.[75] I have decided to use OpenCV-Python for our project. OpenCV uses many algorithms for body recognition which is what we are focusing on. One of them is Viola-Jones Haar Cascade and another is Histogram of Oriented Gradients (HOG).[76]

3.3 Haar Cascade vs. Histogram of Oriented Gradients (HOG)

Algorithms are specifically useful in handling computer vision tasks. The Haar Cascade and Histogram of Oriented Gradients (HOG) are two of the basic and classical algorithms used for handling such tasks like detection and image segmentation to name a few. The two algorithms are the building blocks for more advanced algorithms that are discussed later on in this chapter.

3.3.1 Haar Cascade

The Haar Cascade is an algorithm used to detect objects, bodies and faces in video as well as images. It uses CNN for processing objects, bodies and faces in a video or image. Haar Cascade was introduced by two individuals Viola (Paul Viola) and Jones (Michael Jones) in 2001.[77], [78] They wrote a paper known as “Rapid Detection using a Boosted Cascade of Simple Features” that introduced this technique that has been used for a long time ever since it was introduced.[78] When this technique was developed it utilized the combination of features. These features known as Haar features were able to detect patterns for edges, lines and center-surrounded features.[77] When the features are detected they are recognized and processed so that when scanning an image or real time video those features are not repetitive. The reason for this is all about time and efficiency. This is ideal for

detecting objects in images however there is cascade which breaks down the process of detection[77].

The technique is very popular and is used among the users of OpenCV as well as OpenCV-Python. Haar Cascade is very useful due to its ability to exclude irrelevant regions when scanning an image.[77] Haar Cascade is an algorithm that is used in various ways. Some of these include recognition and detection in real time video or images. The way Haar Cascade works is with pre-built packages known as classifiers and with these pre-trained classifiers we can use them to our advantage in our computer vision program.[77] The Haar Cascade pre-trained classifiers are stored as XML files.[77] The only way for this to work in the computer vision program is to reference to the XML file. Some of these XML files are trained better than others and can be modified to be more accurate.

The detection of objects can have errors due to the use of these classifiers which affect the accuracy of detection or recognition. However, these classifiers can handle almost detecting everything you can think of in an image or real time video such as face, body (lower/upper/full) and even color detection.[77] Haar Cascade is only one option for detecting bodies in a real time video there is another which is HOG.

3.3.2 Histogram of Oriented Gradients (HOG)

Another algorithm used for body detection in real time is HOG. HOG is more of an accurate and fast choice when it comes to body detection. HOG was introduced by Navneet Dalal and Bill Triggs in a paper they wrote which was “Histogram of Oriented Gradients for Human Detection”.[79] The algorithm extracts out features based on the gradients from the pixel intensities which it analyzes to determine the edges of the body.[80] When this is done those edges of the body shall be highlighted to make it clear and known that these are the edges of the person that is being detected. HOG doesn’t allow appearance change due to lighting or position in fact it can handle the different appearances that can be caused by lighting or position.[80]

It recognizes patterns that capture an individual’s body shape. The creation of these patterns helps even if the person is partially shown in an image or video which means it can still identify the visible parts of the body. The process is known as descriptor patterns.[80] HOG features are then input into a classifier which is a Support Vector Machine (SVM) allowing accurate predictions.[80] The HOG feature descriptor can be calculated by certain steps which are preprocessing, calculate the gradient images, calculate histogram of gradient, normalization and calculate the Histogram of Oriented Gradients feature vector.[79], [81], [82], [83]

The algorithm has a process that it utilizes for the best result in detecting objects, people and hand gestures. HOG is a better choice than Haar Cascade due to its efficiency in real-time applications and its ability to handle variations such as different lighting. Behind the HOG algorithm comes an extensive process of how the histogram is created. The first thing that is done is image or frame preprocessing which is adjustment of size on image or frame creating a patch (smaller part from larger image or frame).[82], [83] Next is calculating the

histogram of gradient images or frames by using kernels for filtering the image or frame.[83] If OpenCV is used there is an operator known as Sobel that takes in kernel size which should be set to 1.[83]

Then the magnitude and direction of the gradient is determined by using a function in OpenCV known as cartToPolar.[83] Once every pixel's gradient's magnitude and direction is calculated then it is broken into cells which can vary in size the larger the cell the faster the process runs. The calculation of histogram of gradients happens for every one of the cells.[83] After this the histogram is formed and normalization begins.

The last part of the process brings all the vectors that were generated in the last two steps together to create a giant vector this is known as calculating the Histogram of Oriented Gradients feature vector.[83] All of these steps are used in the HOG algorithm and shows that this algorithm is more structured for real time human detection. Although it still has errors that occur like false positives (non-people) and false negatives (people not detected) when detecting either in an image or video even if it is more accurate than Haar Cascade.[81] Also OpenCV only has detectors for people.[81] There are newer OpenCV algorithm's that have recently come out.

3.4 Newer OpenCV Algorithm's

Recently newer more advanced algorithms have been developed and designed to be used in OpenCV. The newer algorithms use certain aspects of older algorithms previously mentioned. These newer algorithms can do more advanced tasks such as real-time detection, recognition and tracking movement which improves the accuracy as well as the efficiency of those tasks.

3.4.1 YOLO (You Only Look Once)

In OpenCV there is another set of algorithms known for their accuracy, speed and efficiency which is YOLO. YOLO has many versions that all vary in different performance rates and abilities in detection. YOLO was developed by Joseph Redmon and Ali Farhadi in 2015.[84] The real time object detection algorithm both classifies and detects the object at the same time. The process makes this algorithm very versatile in its ability to process the detection of objects faster and more efficiently, which is useful in detection in videos. The algorithm is a single stage object detector which means it can classify and detect in one single pass through a CNN.[84], [85]

YOLO is used in self-driving cars, surveillance systems and robotic systems. Also, it is very useful in real time detection as well as recognition such as real time video analytics and surveillance. Due to the algorithm being useful in various applications it makes other algorithmic models non-existent. It has a pipeline architecture meaning it combines specific regions together.[85] YOLO has its benefits such as the ability to detect objects instantly, resource use is optimized, performance, accuracy, flexibility and efficiency.[85]

Even though YOLO has great benefits making it one of the top algorithms it does have its downfalls. The downfalls for YOLO are that it only focuses on object detection not image segmentation, less accurate than some other methods like two-shot object detection methods, struggles with detecting very small objects and no tracking capability for tracking objects over time. However, there are many versions of YOLO meaning that the limitations might not be the same which is the reason I focused on the most recent three which are YOLOv9, YOLOv10 and YOLOv11.

3.4.1.1 YOLOv9

YOLOv9 is the 9th version of the YOLO algorithm. It combines models together to create a more accurate and efficient algorithm. YOLOv9 was created by Chien-Yao Wang, I-Hau Yeh and Hong-Yuan Mark Liao on February 29th, 2024.[86] Version 9 was mentioned and described in detail in a paper written by Chien-Yao Wang with his team known as “YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information” [86]. The paper went into detail on the design and the structure of YOLOv9.

One of the issues when it comes to deep neural networks (DNN) is that it loses input data information which is known as information bottleneck.[86], [87], [88] There are currently three different methods used to combat this issue which can reduce the loss of information. The three methods used are reversible architecture, masked modeling and deep supervision.[86], [88] Although they reduce information loss they are not well developed in training and inference processes [88]. YOLOv9 introduced two very useful techniques which happen to be Programmable Gradient Information (PGI) and Generalized Efficient Layer Aggregation Network (GELAN) [86], [87], [88]. Both making YOLOv9’s learning capacity increase and able to retain crucial information when detecting objects.[87] PGI and GELAN also help with the performance of version 9. Version 9 tackles the Information Bottleneck Principle and the use of Reversible Functions [87].

The Information Bottleneck Principle is a challenge in DNN which is as data passes through multiple layers of a network there is potentially an increase in loss of information [87]. YOLOv9 implements PGI to handle this challenge by preserving data throughout the network which ensures reliable gradient generation, better model convergence and performance [87]. YOLOv9 can use reversible functions which are reversible if the function can be inverted without any loss of information and with the use of reversible functions it can mitigate risk of information degradation [87].

PGI is a combination of a few things that deal with DNN issues. It has a main branch for inference, an auxiliary reverse branch for reliable gradient calculation and multi-level auxiliary information to deal with deep supervision issues effectively without extra inference costs which is shown in Figure 3.6 [88].

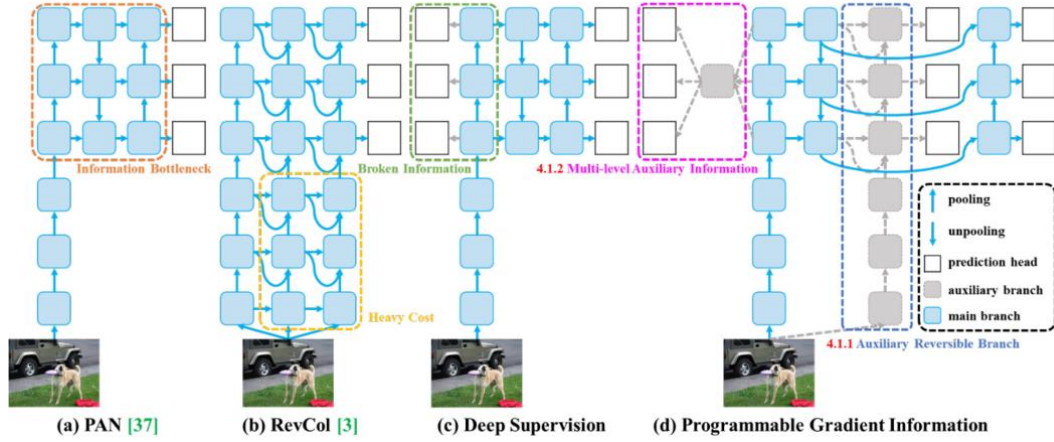


Figure 3.6. PGI Architecture from Reference [88]

PGI also includes an auxiliary supervision node that helps with backpropagation of gradients.[88] The main branch maintains high performance during inference process and can bypass auxiliary components as needed to keep this performance the same.[88] The auxiliary reversible branch utilizes the reversible architecture to maintain information from getting lost in the layers of the DNN and retaining the information for learning.[88] It helps by not allowing anything to affect the inference speed. The last part of PGI is the multi-level auxiliary information which helps with making better predictions of objects no matter the size [88].

GELAN is more of a supporter for the PGI in that it provides a flexible and efficient design that supports various computational blocks shown in Figure 3.7.

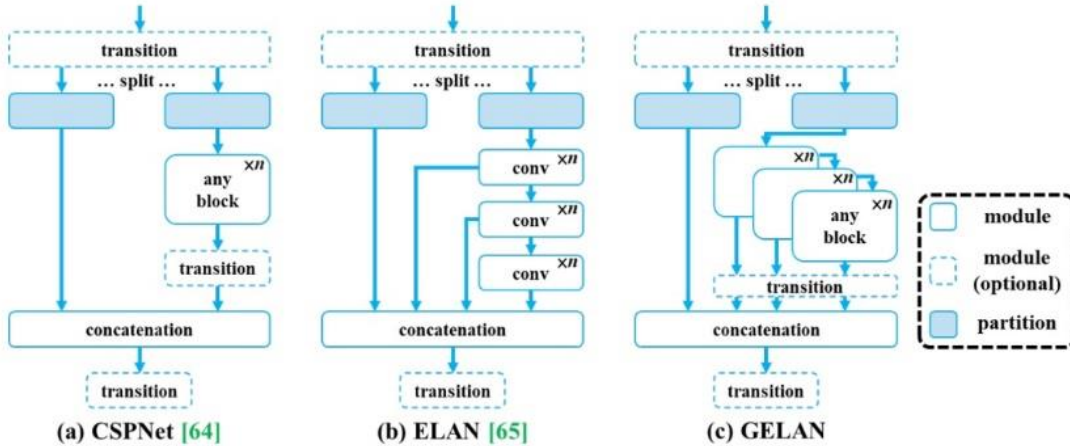


Figure 3.7. GELAN Architecture from Reference [88]

It combines features from CSPNet and ELAN. GELAN takes CSPNet's gradient path plan with ELAN inference speed optimization to make it a versatile architecture.[88] It has quick inference times with retaining accuracy. GELAN has two critical aggregation techniques which happen to be SPPELAN and RepNCSPPELAN4 models.[88] SPPELAN is the implementation of Spatial Pyramid Pooling (SPP) within ELAN architecture. SPPELAN deals with layer aggregation in such a way that it is able to extract from spatial hierarchies [88]. RepNCSPPELAN4 is a component that advances the CSP-ELAN to

improve the feature extraction process [88]. GELAN shows that it can keep and retain information of features after many layers in a DNN.

YOLOv9 has many models which are broken down into lightweight, medium and large models. The models vary in performance ratings for real time object detection which were tested with the MS COCO dataset shown in Figure 3.8.

Detection (COCO)		Segmentation (COCO)			
Model	size (pixels)	mAP ^{val} ₅₀₋₉₅	mAP ^{val} ₅₀	params (M)	FLOPs (B)
YOLOv9t	640	38.3	53.1	2.0	7.7
YOLOv9s	640	46.8	63.4	7.2	26.7
YOLOv9m	640	51.4	68.1	20.1	76.8
YOLOv9c	640	53.0	70.2	25.5	102.8
YOLOv9e	640	55.6	72.8	58.1	192.5

Figure 3.8. Comparison of State-of-the-Art Real-Time Object Detectors from Reference [87]

In the table above there is the number of parameters and computational needs (FLOPs) for each model of YOLOv9 [87].

3.4.1.2 YOLOv10

YOLOv10 was introduced by Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Jungong Han and Guiguang Ding in a paper known as “YOLOv10: Real-Time End-to-End Object Detection” which was released May 23rd, 2024.[89] YOLOv10 is a built up from the previous version of YOLO though includes techniques to handle even more accurate detection in real-time. It has two major components, which are Consistent Dual Assignments for NMS-free Training and Efficiency-Accuracy Driven Model Design.[89], [90] The Consistent Dual Assignments for NMS-free Training consists of two components which are dual label assignments and consistent matching metrics [89], [90].

The process deals with Non-Maximum Suppression (NMS) post-processing with help from Detection Transformer (DETR) Architecture due to redundancies of bounding boxes in detection which is usually caused by NMS post-processing [89], [90]. Dual label assignments assist with duplication and redundancy from NMS post-processing by avoiding the use of NMS post-processing all together. It combines one-to-one and one-to-many matching techniques to make a more sufficient prediction during the inference process [89], [90]. When it goes through the inference process it will disregard the one-to-many head and use the one-to-one head to make the prediction however during training both heads are jointly optimized for the model [89].

The joint optimization allows greater supervision with use of the one-to-many assignment which improves the supervision from previous versions of YOLO.[89] Dual label assignments reduce inference cost and adopts the one selection which performs the same

as the Hungarian matching reducing the training time.[89] Consistent matching metric is used for making sure that one-to-one head aligns with one-to-many head during training.[90] The metric for both heads allow for the model to use the best samples instead of using samples that are not best for both heads.

Next major part to YOLOv10 is Efficiency-Accuracy Driven Model Design. The Efficiency-Accuracy Driven Model Design is for balancing the efficiency and accuracy of the algorithm.[90] Unlike previous YOLO versions that do have improvements in their efficiency and accuracy, they still have redundancies as well as limitations. The Efficiency-Accuracy Driven Model Design is split into two models, one is for efficiency and the other is for accuracy. Efficiency driven model design reduces computational cost without reducing its overall performance.[90] There are three ways that the Efficiency driven model does this which are lightweight classification head, spatial channel downsampling and rank-guided block design.[90]

YOLOv10 adopted a lightweight classification head which uses two depthwise separable convolutions.[90] Spatial channel downsampling is common in typical YOLO versions and models, however, cause computational cost. YOLOv10 decouples the spatial reduction and channel increase operations allowing efficient downsampling.[89] Besides doing both depthwise convolution for reduction and increasing number of channels in a single convolution layer like most algorithms. YOLOv10 can do both by pointwise convolution which can retain more information during downsampling.[90]

The last way Efficiency driven model design reduces the computational cost is by rank-guided block design which assigns ranks to stages in the convolution layers. The rank-guided block design calculates a rank of each stage, identifies redundant stages and replaces them with compact inverted blocks (CIB) structure.[90]

The Accuracy driven model design has two ways to increase the accuracy of the algorithm. One of the ways is large-kernel convolution and the other is partial self-attention (PSA).[89], [90] The large-kernel convolution resizes convolution area to be larger than its original size. When it comes to YOLOv10 the use of large-kernel convolution is used in CIB within deeper stages to increase the size of the second depthwise convolution which is 3x3 to 7x7.[89], [90] Also, adds structural reparameterization techniques which includes the addition of another 3x3 depthwise convolution branch.[90] The addition of this branch helps optimization issues with no additional inference cost.[90]

Version 10 can detect smaller objects by the increase in their performance without any cost to either inference or computation. PSA is introduced in version 10 to use global representation learning with reduction in computational cost.[90] PSA splits features into two parts after a 1x1 convolution which is merged later by 1x1 convolution after being concatenated.[90] The PSA is only applied to low resolution stages to reduce excessive overhead from complexity of self-attention and improves overall capabilities of YOLOv10 algorithm.[90]

Version 10 has many features that are included to improve detection of individuals or objects in real-time. The features are advanced backbone architecture, enhanced feature pyramid network, adaptive anchor-free detection, advanced data augmentation and hybrid precision training.[91] Most of these features are built off from previous versions of YOLO and mainly from version 9. Although they are improved by YOLOv10's structure and design. There are also different variations of version 10.

Model	Test Size	#Params	FLOPs	AP ^{val}	Latency
YOLOv10-N	640	2.3M	6.7G	38.5%	1.84ms
YOLOv10-S	640	7.2M	21.6G	46.3%	2.49ms
YOLOv10-M	640	15.4M	59.1G	51.1%	4.74ms
YOLOv10-B	640	19.1M	92.0G	52.5%	5.74ms
YOLOv10-L	640	24.4M	120.3G	53.2%	7.28ms
YOLOv10-X	640	29.5M	160.4G	54.4%	10.70ms

Figure 3.9. YOLOv10 Model Variations from Reference [92].

YOLOv10 has many different models like YOLOv9. It is separated the same way as well with lightweight, medium and large models. The six different models depend on efficiency and size which are shown in Figure 3.9.

3.4.1.3 YOLOv11

The newest version of YOLO is YOLOv11. It is currently the most recent version of YOLO to date and improves based off version 8 however it is more accurate and efficient over version 10. YOLOv11 was developed by Ultralytics on September 27th, 2024.[93] It was announced at Ultralytics YOLOVision event making its official release to the public.[93], [94], [95] YOLOv11 is an improvement from the team at Ultralytics from version 8 which was the latest version from Ultralytics that was released. YOLOv11 architecture has new integrations and parameter tuning.[94]

YOLOv11 uses a C3k2 block which is a custom implementation of the CSP Bottleneck.[94] The C3k2 block is faster and efficient in computation without loss of performance.[94] It uses two convolutions and not one large convolution.[94] When it comes to Cross Stage Partial (CSP) Bottleneck it is a network that splits the feature map and takes one part through a bottleneck layer then merges it to the other part with the output of the bottleneck.[94] In other words the CSP Bottleneck module divides the feature map into two parts which one part goes through bottleneck layers and the other goes to the output.[94] YOLOv11 also utilizes Cross Stage Partial with Spatial Attention (C2PSA) which improves spatial attention in the feature maps giving the ability to focus on a given region of interest more effectively.[94] If a region is focused on then features can be filtered out spatially.

The algorithm adds C3k2 and C2PSA blocks to the backbone which deals with extraction of features from an input image at multiple scales and resolutions.[94] The additions to the backbone increase the capability of spatial attention and efficiency in computations making it faster and perform a bit better than version 10. The C3k2 block is also in the neck which is the aggregating of features from multiple resolutions and is passed to the head for prediction.[94] If the C3k2 block is included in the neck it once again improves speed and performance of feature aggregation process.[94] The way that YOLOv11 uses C2PSA to focus on key regions for better detection makes it able to detect obscured and small objects.[94]

The head part of the architecture deals with making the final predictions meaning bounding boxes and classifying objects.[94] In the head part of the architecture for YOLOv11 inference is faster and parameters are more efficiently defined by the addition of C3k2 block making this possible.[94] Very similar to previous versions of YOLO YOLOv11 has five different variations.

Model	size (pixels)	mAP ^{val} 50-95	Speed CPU ONNX (ms)	Speed T4 TensorRT10 (ms)	params (M)	FLOPs (B)
YOLO11n	640	39.5	56.1 ± 0.8	1.5 ± 0.0	2.6	6.5
YOLO11s	640	47.0	90.0 ± 1.2	2.5 ± 0.0	9.4	21.5
YOLO11m	640	51.5	183.2 ± 2.0	4.7 ± 0.1	20.1	68.0
YOLO11l	640	53.4	238.6 ± 1.4	6.2 ± 0.1	25.3	86.9
YOLO11x	640	54.7	462.8 ± 6.7	11.3 ± 0.2	56.9	194.9

Figure 3.10. YOLOv11 Model Variations by Reference [90].

YOLOv11 has lightweight, medium and large models just like the other prior versions. Not only does it have the models, but it also has features like enhanced feature extraction, optimized for efficiency and speed, greater accuracy with less parameters, adaptable to be accessed from other environments and a range of tasks supported by version 11.[93] The tasks that YOLOv11 can handle are object detection, image segmentation, image classification, pose estimation and oriented bounding boxes (OBB).[94], [95]

3.4.1.4 YOLO-NAS

Another version of YOLO is YOLO-NAS which is different than the others by not only structure but its functionality. YOLO-NAS was developed by Deci AI and was designed using Deci's AutoNAC NAS technology.[96], [97], [98] The way the algorithm model works is by Neural Architecture Search (NAS) technique meaning it can search for the best network architecture for a certain task without human intervention.[96], [98], [99] NAS searches for an architecture that has the best trade-off between multiple limitations of prior YOLO versions like accuracy, complexity and model size.[97], [98] Although NAS is applied by AutoNAC (Automated Neural Architecture Construction) which is an engine

that allows task input, data characteristics, inference setup and precise performance tasks.[96], [99]

After AutoNAC takes all the required information it then finds and identifies the most optimal architecture. The architecture that is found has a balance between precision as well as inference speed for the certain task provided by the user to AutoNAC. In inference there is a stack known as inference stack that has quantization and compilers. AutoNAC accounts for the quantization and compilers that are included in the inference stack.[99]

YOLO-NAS introduces a brand-new block which is a quantization aware block and selectivity of quantization.[96], [99] The selectivity of quantization is mainly for performance optimization making YOLO-NAS perform at its best. Another newly introduced feature is these quantized versions of YOLO-NAS which are INT8 versions. The INT8 versions use quantization techniques to convert from floating-point models to integer models, allowing more efficient inference.[96], [99] YOLO-NAS still uses inference reparameterization and spatial attention like prior YOLO versions. When the quantized versions are used there is a slight precision drop which is a bit different than other versions of YOLO.

Just like the other versions of YOLO that have features so do YOLO-NAS. The main features of YOLO-NAS are quantization-friendly basic block, sophisticated training and quantization as well as AutoNAC optimization which includes pre-training.[97] Also, YOLO-NAS has its own models like those that have been mentioned before in other YOLO versions. It has lightweight, medium and large models. However, it only has three models YOLO-NAS S for small, YOLO-NAS M for medium and YOLO-NAS L for large unlike other versions that have more variations. Although it does have the quantized variation making the total number of models six which is shown in Figure 3.11.

Model	mAP	Latency (ms)
YOLO-NAS S	47.5	3.21
YOLO-NAS M	51.55	5.85
YOLO-NAS L	52.22	7.87
YOLO-NAS S INT-8	47.03	2.36
YOLO-NAS M INT-8	51.0	3.78
YOLO-NAS L INT-8	52.1	4.78

Figure 3.11. YOLO-NAS Model Variations by Reference [99].

The algorithm model is very flexible, scalable and useful for a variety of things including security systems. It has more performability for speed, accuracy and efficiency for many areas of limitations in other versions.

3.5 Comparison of Computer Vision Algorithm Models

After collecting and gathering the data from the other sections, three of the top computer vision models were chosen to compare. The models can be seen in the above tables with their own variations included. I decided to label them “Model 1”, “Model 2” and “Model 3” for simplicity purposes when referring to them.

Table 3.2 Comparison table of computer vision algorithms.

Model 1		
Model	mAP	Latency (ms)
YOLOv10 N	38.5	1.84
YOLOv10 S	46.3	2.49
YOLOv10 M	51.1	4.74
YOLOv10 L	53.2	7.28
YOLOv10 X	54.4	10.70

Model 2		
Model	mAP	Latency (ms)
YOLOv11 N	39.5	1.55
YOLOv11 S	47.0	2.46
YOLOv11 M	51.5	4.70
YOLOv11 L	53.4	6.16
YOLOv11 X	54.7	11.31

Model 3		
Model	mAP	Latency (ms)
YOLO-NAS S	47.5	3.21
YOLO-NAS S INT8	47.03	2.36
YOLO-NAS M	51.55	5.85
YOLO-NAS M INT8	51.0	3.78
YOLO-NAS L	52.22	7.87
YOLO-NAS L INT8	52.1	4.78

In the tables above Model 1 seems to be everything we look for in a computer vision algorithm model with high precision and low latency levels. Model 1 is referencing YOLOv10 and is broken into comparable variations with Model 2. It has a high mAP rate which is the mean average precision rate. The mean average precision is the precision of detection for objects in either image or video and the accuracy of bounding boxes. Model 1 has low latency levels meaning that the speed of detection is at a faster rate. YOLOv10 N has a 38.5 mAP and latency is at 1.84 milliseconds as shown above. Although this seems

like the route to go, one of our goals is to have high precision as well as accuracy without the cost of speed. This meant that Model 1 was not chosen to be the algorithm model for iSPY.

The decision then was focused on Model 2 and 3. Both models had their own pros and cons, especially when it came to the large model variations. Model 2, which references YOLOv11 has varying rates dependent on model. YOLOv11 N has a 39.5 mAP and latency is at 1.55 milliseconds which is better than YOLOv10 N. It has higher precision with lower latency which is needed for iSPY. Although Model 3, which refers to YOLO-NAS, also has varied rates. YOLO-NAS S is at 47.5 mAP with a 3.21 millisecond latency were as YOLOv11 S has 47.0 mAP with 2.46 millisecond latency. If we move to YOLO-NAS S INT8 then the mAP is 47.03 with 2.36 millisecond latency.

However, if we use YOLO-NAS M then mAP is 51.55 with 5.85 millisecond latency and YOLOv11 M has 51.5 mAP with 4.70 millisecond latency. If YOLO-NAS M INT8 is considered, then mAP is 51.0 with 3.78 millisecond latency. So, Model 3 which is YOLO-NAS will have lower latency at the cost of lower precision on higher models specifically INT8 models for YOLO-NAS than Model 2 which is YOLOv11. Even though YOLOv11 higher end models have higher precision with higher latencies that are not affecting the precision meaning that precision is not at the cost of lower latency though latency is at the cost of higher precision which is why for iSPY we used Model 2 YOLOv11 as our algorithm model for computer vision.

If we are going for high precision and accuracy with lower latency, then either YOLOv11 N or YOLOv11 S are the models to utilize. Even though YOLOv11 M is in the middle of the range it has a much higher latency than the lower two models which makes it a possible choice if YOLOv11 N or S is not selected. Overall YOLOv11 models outperform in speed and accuracy among the other versions.

3.6 Classifiers and Object Detectors

A classifier is used in most computer vision algorithms to classify input data such as input image and video. The classifier can be a pre-built package which is already pre-trained. If the classifier is already pre-trained, then the only thing needed is to reference the file that contains this data. Some of the classifiers used in computer vision algorithms have been mentioned like cascade and SVM classifiers. Many classifiers have switched to object detectors to handle more rigorous tasks such as tracking, segmenting and pose detection. The most recently developed algorithm models for computer vision can do all or at least most of these tasks meaning that the object detector is more advanced than the older classifiers like the ones Haar Cascade and HOG utilize.

Newer algorithm models like YOLO use an object detector. YOLOv9, YOLOv10, YOLOv11 and YOLO-NAS have an object detector which is not a classifier like cascade or SVM, instead it is a single-stage object detector.[84] Also, just like classifiers object detectors can be pre-built packages and pre-trained meaning they can be used by referencing to their file in the computer vision program. The newer models rely on these

object detectors to do the detection and classification together for input image and video data. However, the object detectors still end up utilizing the classifiers to do the classification part as the detector mainly handles the detection and not the classification of the object.

Object detectors and classifiers have their own important way of handling tasks. Object detectors can handle two things at once which helps the algorithms precision, accuracy and efficiency without the cost of loss data such as double bound boxes or bound boxing an item that is mislabeled. The detector can determine objects in real-time which is the reason that improved YOLO versions can focus on pose detection and object tracking. Classifiers can only handle one thing which is classify objects in an image or video usually not used in real-time applications although HOG can handle real-time applications with the SVM classifier.

Even though object detectors and classifiers differ slightly they have commonalities. Both can handle segmentation, captioning and recognition. The only difference when it comes to the commonalities is the way classifiers and detectors handle each one of them. When it comes to segmentation classifiers can give a class label to each pixel in the image whereas detectors can identify then segment the object out of the image. The use of either classifier or detector is dependent on some factors. These factors are computer vision algorithm model used, computer vision task and performance for given task. All these factors determine which one is utilized in the computer vision program.

3.7 Body Recognition vs Body Detection

There is a fine line between recognition and detection which is usually why people mix the two of them up. Security systems can either do one or the other and most current security systems can do both. Body detection differs from body recognition by a couple factors such as labeling, feature extraction and confidence leveling. All these factors have a role in the way detection and recognition work. Although there is a difference it doesn't mean that they can't co-exist together in one system like mentioned earlier in this section.

3.7.1 Body Recognition

Body recognition is like facial recognition and the process is the same with minor additions. Body recognition or facial recognition uses an input image or video frame and scans it using convolution. Breaking the image or video frame up into regions also known as cells which can be 3x3 or larger. After the image is broken up, features are extracted although these features are more detailed. The features that are extracted for facial recognition can be eyebrows, lips and facial hair if applicable. If the facial recognition program is trained with images of different eyes, eyebrows and lips with varying colors then it can determine what eye, eyebrow or lip color the person has by recognition. Also, a facial mapping can be generated using facial recognition.

Body recognition has the same training setup except for size of the individual, movement and pose being added. It differentiates between object classes such as humans, animals and any other object classes that can be created from the image or frame. Body recognition can break down a real-time video and determine the people in each frame by labeling. Labels are used to identify individuals in the frame or image rather than other objects. Mainly body recognition is used for classification purposes to classify an individual based off its features and movement.

3.7.2 Body Detection

Body detection is simply distinguishing an individual in each frame or image from the background. It is trained with features though not to the same extent as body recognition. The features are more rough and rigid than those from body recognition. Body detection boxes out an individual from the image although it won't label them as a person like body recognition would.

When it comes to real-time detection the use of body detection can be used to only box out the person or people in the real-time video frame or image. It utilizes bounding boxes to show the detection of the object whether it is a person, animal or any other object. However, it does not label the object it detects, which means that it is not detailed enough to do the classification. This is why for iSPY we decided to use body recognition instead of body detection for our system.

3.8 Live Video Compression and Decompression

The first thing to discuss is the means of compressing the video to bitstream so that we can properly transfer the data over via the laser. We need to make sure that the live video feed is transmitted efficiently and retains a level of quality that is acceptable to our standards. There are a number of ways of doing this. The most promising option is using a codec to compress the video.

3.8.1 Codec

A codec stands for both encoder and decoder. A codec can be used to compress a large amount of data. They are usually used to compress media to a smaller format for either storage or more efficient transfer of the file. A codec enabled us to compress our data so that we are not wasting a large amount of time and resources transferring the original data. Even further, we could use a predictive codec which uses an algorithm to turn data into bytes. This helped give us a much more efficient way of transmitting the data via bit [100].

3.8.2 Lossy vs Lossless

There are two kinds of compression: lossy and lossless. Lossy compression is able to compress the data by removing some of the original data. Lossless compression compresses data by “removing unnecessary metadata” [101]. Lossy compression is used when a file doesn’t need all the data or if there is a need to have storage freed up. It would basically get rid of things that are not important to the average user viewing the feed. Lossless compression keeps the original quality of the image but doesn’t reduce the size as much.

3.8.2.1 Lossy Compression

Let’s compare the two types of compression. Lossy compression reduces the size of the files greatly at the cost of reducing quality. The bandwidth of data transfer would be lower due to the reduced size of the file. However, because it reduces the size of the files by getting rid of unimportant looking areas, the data lost from the compression is permanently lost and cannot be salvaged. Lossy compression is typically used when it comes to web-based compression but that won’t be an issue for our project. Lossy compression also might cause blurring or pixelation on our feed [102].

3.8.2.2 Lossless Compression

Lossless compression might seem like the best way to move forward but there are a few aspects to consider. Lossless compression will maintain the original quality of the image. This results in no data loss. We can compress the data and return it to its original format without any data loss. Since we are building a surveillance system, quality is a big aspect to consider when choosing our codec. We wouldn’t want to not be able to detect a person in the live feed due to quality loss from compression [102].

3.8.3 Video Compression Techniques

There are many different ways of compressing video feed. It’s important to consider all of them when planning out a project. We’ll be looking at a few methods on how to compress a video and what kind of process it takes. We’ll be looking at spatial compression, temporal compression, and entropy coding.

3.8.3.1 Spatial Compression

Spatial compression (intraframe compression) is a method of compressing data by reducing “...the size of each individual frame independently by reducing redundant information within the frame...” [103]. This allows us to reduce the size of the data while maintaining a level of quality to our image. Let’s say there is an image of a field of flowers of all different kinds of colors and a clear blue sky. This technique looked at the pixels of the blue sky and determine if they can merge some of the pixels into a larger area. This required less information to store as it’s just a large blue area. However, a field of flowers of all

different colors will have a lot of variation. In this scenario, compression would hurt the quality of the image by merging colors together. An example of how spatial compression works can be seen below in Figure 3.12.

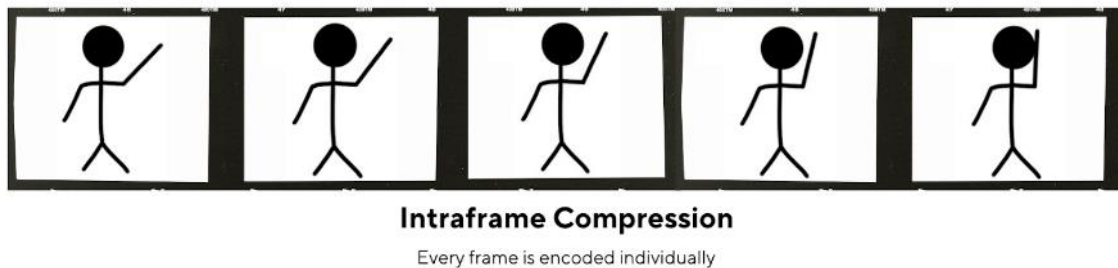


Figure 3.12. – Image Depicting Spatial Compression (Intraframe Compression) [103]

3.8.3.2 Temporal Compression

Temporal compression (interframe compression) is a method of compression that starts with a single image. This image acts as a starting point of reference for the coming frames. It saves the image and only saves changes that occur after the starting point. This starting image is called the “I-frame”. The next frames are classified as P-frames (Predictive frames) and B-frames (Bi-directional frames). P-frames store data based on changes in previous frames. B-frames store data based on changes in both previous and coming frames. This helps in determining what information is important or not to make the least number of changes possible [103]. This method compresses data into a much smaller size than spatial compression but it has potential to lose the quality of the original video. An example of how temporal compression works can be seen below in Figure 3.13.

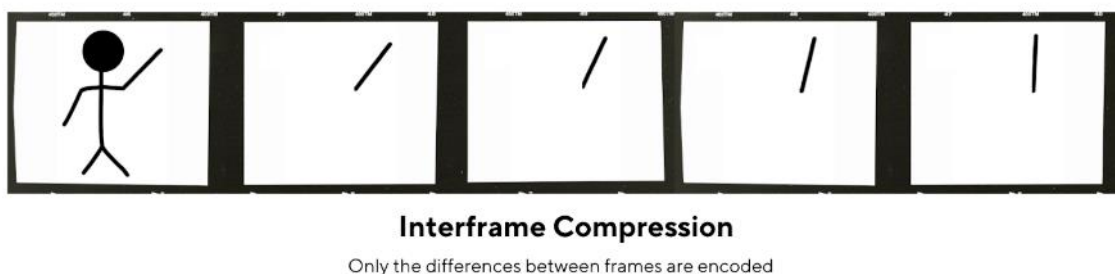


Figure 3.13. Image Depicting Temporal Compression (Interframe Compression) [103]

3.8.3.3 Entropy Coding

Entropy coding is the process of data compression based on statistics. One example of entropy coding is Huffman coding. Huffman coding basically takes frequently occurring symbols (symbols could letters, numbers, words, or in our case pixels) and assigns a small bit value to the symbols. In contrast, the symbols that occur less are assigned longer bit lengths [GPTA]. A good example to explain Huffman coding is an example from Prateek Joshi who wrote an article about entropy coding [104]. Let’s say we have a sequence of “SSCAAACCKKKKAKKKKKKKCCCAKKKK”. It takes 5 bits to represent every

letter of the alphabet and the length of the sequence is 30. It would take $30 \times 5 = 150$ bits to display the sequence given. Huffman coding allows us to shorten the sequence by quite a margin. S only occurs 2 times which makes it the least occurring letter in the sequence [104]. We can do this for every letter in the sequence and determine its frequency. The assigned bits are shown in the table below (Figure 3.14).

S = 000 (3 bits)
A = 001 (3 bits)
C = 01 (2 bits)
K = 1 (1 bit)

Figure 3.14. Assigned Bit Lengths of Letters in Sequence [104]

We assign the longest bit length to the symbol that occurs least, and we assign the shortest bit length to the symbol that occurs most. Now we can show the sequence using only 51 bits ($3 \times 2 + 3 \times 5 + 2 \times 7 + 1 \times 16 = 51$) [104]. That is nearly a third of the size of what we were initially going to transmit. We also kept the original data intact by doing this method. Huffman coding is also compressed into the form of bitstream which allows us to transmit the data via bitstream with our FSO system.

Arithmetic coding is another example of entropy coding. Arithmetic coding can be more efficient than Huffman coding because it encodes “entire sequences of symbols as a single number (usually a fraction)” [GPTA]. Arithmetic coding looks at the individual frames of the video and predicts the probability of different symbols. Huffman coding assigns bit lengths to each symbol based on how frequently it shows up. However, “...arithmetic coding represents the entire sequence of symbols as a fractional number within a range based on their probabilities” [GPTA]. This resulted in a better compression to bitstream. Arithmetic coding requires more computational power which could result in issues later on.

3.8.4 Live Transmission Limitations

There are two main factors to consider when talking about the live transmission of data. The first factor is bandwidth. In our case, bandwidth is the maximum amount of data we can transfer over a set amount of time [105]. Lower bandwidth gave us less room to work and may require us to compress the video feed further, which would also reduce the quality of the original feed. Higher bandwidth gives us the freedom to transmit higher resolution imaging and avoiding delays. We had to choose between methods of compression to best suit our needs according to the bandwidth.

The second factor is latency. The entire process of compression and decompression took time and is something that should be considered when dealing with a live transmission. Compression of more data will decrease the bandwidth required but will increase latency due to the amount of time it will take to compress more data. Inversely, less compression

will result in a higher bandwidth requirement but lower the latency [GPTB]. Propagation delay is also a factor in latency. The laser will take a certain amount of time to reach the receiver.

3.8.5 Codec Comparison

We've gone over what a codec is and the different kinds of methods when it comes to video compression in codecs. Now we will discuss the different kinds of codecs developed by different companies. There are a wide range of codecs to choose from when it comes to video compression. We'll be going over the most popular and practical ones that we can use in our system.

3.8.5.1 H.264

One of the most popular codecs is H.264. It allows for streaming at higher qualities while maintaining a lower bitrate. It is a lossy codec which could result in permanent data loss of the original image. While the data loss could be mostly unnoticeable, it is something to consider when comparing it to other codecs. However, there is a lossless version of H.264. Due to H.264 and its popularity, it is widely available and compatible with many devices [106].

3.8.5.2 H.265

H.265 is the successor to H.264 (there is also a lossless version of H.265). It compresses data at about half the bitrate of H.264. The stream should support much higher resolutions due to the decreased bandwidth and has better quality over H.264. However, it is a newer model and is not as widely used as H.264. This could cause run ins with compatibility issues. H.265 also requires much more computational power because of the higher complexity of the process [106].

3.8.5.3 VP8/VP9

VP8 is a lossy codec designed by On2 Technologies and now Google owns it. Google later improved upon it and developed VP9 (also a lossy codec). VP9 is good for web streaming and is fairly good at compressing and transmitting data with lower bitrates. The codec is also open source which allows for easy access [106].

3.8.5.4 AV1

AV1 is a lossy codec developed by Alliance of Open Media (AOMedia). It's main purpose is to stream quality video over an internet connection. This codec is open source and supports developments in technology like higher resolution or even virtual reality. However, AV1 is not as widely used as other codecs. It also requires more computational power compared to other codecs [106].

3.8.5.5 FFV1

FFV1 is a lossless codec developed by Michael Niedermayer. It is popular for both its speed and size. The codec is mainly used for archival or editing purposes and was acknowledged to be a “viable archival encoding” by the U.S. Library of Congress. The codec also utilizes entropy coding and is compatible with a wide variety of devices and software [107].

Table 3.3 Comparison table of different codecs.

Parameter	Codecs				
Brand	H.264	H.265	VP8/9	AV1	FFV1
Type	Lossy/Lossless	Lossy/Lossless	Lossy	Lossy	Lossless
Efficiency	High	Very High	High	Very High	High
Pros	Widely used and compatible with most devices	Better version of H.264	Open source and efficient on web	Open source and very efficient	No data loss during compression, open source, efficient
Cons	Not the best for higher resolutions	More computational power and not as widely used	Not as widely used	Requires more computational power	Not typically used for live transmission

3.8.6 Transmission of Data

The codec transforms the live video feed into a compact bitstream format. This bitstream format is exactly what we need to transmit the data via the FSO system. There are a few ways we could approach the issue of transmission. Pulse code modulation (PCM) is when an analog signal is converted into a binary sequence. In the process of demodulation, we get the original analog signal [108]. However, since we have already converted the original data into bitstream via a codec, we can transmit the data directly using PCM.



Figure 3.15. Demonstration of PCM Output [108]

The laser can pulse in intervals to indicate different binary digits to properly decompress the transmission. As seen in Figure 3.15 above, there are varying lengths of pulses of which

could be different values being sent based on the length of the pulse [109]. When the laser is off, the binary value was set to 0. When the laser is on, the binary value was set to 1.

Amplitude modulation (AM) can act in a similar way to PCM. We can amplify the laser to different levels of intensity to indicate a change in data. We could also accomplish this with frequency modulation (FM). While not necessarily the best option, we could use phase modulation (PM) as well [110] [GPTE].

3.8.7 Loss of Data

What do we do in the scenario that some data is lost during transmission? There are a few ways we can solve that. Lost data is bound to happen from time to time due to interference with the laser. This could be from a bird flying through the laser or dust blocking it for a moment. The first way to solve this problem is to simply store records of the recorded footage so that, if needed, the footage can be viewed at a later time if something was missed.

Another way to solve this problem is with error detection. Reed-Solomon codes are one of the ways to solve this problem. Reed-Solomon codes were developed by Irving S. Reed and Gustave Solomon. How it works is that it takes a data block and adds extra parity bits (redundant bits). When the transmission has been received, the decoder checks to see if any bits have been corrupted and replaces them with the parity bits [111].

Hamming code is another code that is used for bit errors and correcting these errors. It was developed by R.W. Hamming. It is like Reed-Solomon code in that it inserts parity bits (redundant bits) before transmission. It then performs recalculations to detect errors and find out the position of said errors [112].

3.8.8 Video Compression Libraries

Libraries are very helpful when it comes to any kind of code. There are some things that are extremely complex when it comes to individual coding. For example, writing a codec on your own is very complex and time consuming. This is why we'll be looking at the different libraries when it comes to video compression in three potential languages.

3.8.8.1 C++

C++ has a few different libraries that allow us to utilize codecs. The first library is FFmpeg (libavcodec). This library supports all kinds of codecs. It includes H.264, VP8, VP9, AAC, MP3, and a lot more. It is widely used and documented well which enabled for easy understanding. It is also very optimized to include all of these different codecs [113] [GPTC].

The second library is GStreamer. GStreamer is much more complex and allows different kinds of codecs to act as plugins. While it is complex, it allows for customization that other libraries do not offer [114] [GPTC].

The third library is OpenH264. This codec only focuses on H.264 but since it only focuses on H.264, it is quite efficient. This library is useful for real-time video compression. The library is also open source which allows anyone to use H.264 for free without any licensing costs. The only drawback to this library is that it only uses H.264 and does not support other formats [115] [GPTC].

Table 3.4 Comparison table of different libraries for C++

Parameter	Libraries		
Library	FFmpeg (libavcodec).	GStreamer	OpenH264
Performance	High	High	High
Supports Multiple Codecs?	Yes	Yes	No
Well documented?	Yes	Yes	Yes
Optimized For Live Feed?	Yes	Yes	Yes

3.8.8.2 Java

Java has its own libraries as well. The first library is called JCodec. It is purely written in Java and supports codecs like H.264 and VP8. Unfortunately, it has limited codec support and lack optimization for real-time applications [GPTC]. The developers of the program have acknowledged that since the library is all in java, it will be “an order of magnitude slower than native implementations (such as FFMpeg)” [116].

Xuggle (Xuggler) is a library in java that allows FFmpeg to be implemented into java. Because it is based on FFmpeg, it supports most codecs and is able to handle encoding and decoding well [GPTC]. Xuggler is open source and has cross platform support. It allows for flexibility and community support due to the project being open source [117].

The third library for video compression in java is JMF (Java Media Framework) [118]. It is a much older library that has support for a few codecs but is no longer being worked on [GPTC]. Due to the outdated nature of the library, I think it’s safe to say there are better options.

Table 3.5 Comparison table of different libraries for java

Parameter	Libraries		
Library	JCodec	Xuggle (Xuggler)	JMF (Java Media Framework)
HighPerformance	Medium	High	Low
Supports Multiple Codecs?	Yes (but limited)	Yes	No
Well documented?	Yes	Yes	No
Optimized For Live Feed?	No	Yes	No

3.8.8.3 Python

PyAV is the first library we’re going to discuss that is used for video compression [GPTC]. According to the official website of PyAV, “PyAV is a Pythonic binding for FFmpeg. We aim to provide all the power and control of the underlying library but manage the gritty details as much as possible” [119]. PyAV can be used to manipulate data via specific processes like codecs, containers, or streams. The developers do warn that PyAV can be a hindrance at times, they also claim that it is a “critical tool” when it must be used [119].

The second library that we’re going to compare is MoviePy. MoviePy is different in that it allows many different functions when it comes to video editing. Some of these features include “...cutting, concatenations, title insertions, video compositing (a.k.a. non-linear editing), video processing, and creation of custom effects” [120]. While it can handle codec related tasks, the primary function of this library is to edit videos [GPTC].

The third library used in Python for video compression is FFmpeg-Python. This is essentially a “Python wrapper for FFmpeg” [GPTC]. This essentially allows the user to fully use FFmpeg commands and associated codecs while using python. It is well documented as the library comes from the same FFmpeg website as the C++ FFmpeg library [113].

Table 3.6 Comparison table of different libraries for python.

Parameter	Libraries		
Library	PyAV	MoviePy	FFmpeg-Python
Performance	High	High	High
Supports Multiple Codecs?	Yes	No (mainly a video editing tool)	Yes
Well documented?	Yes	Yes	Yes
Optimized For Live Feed?	Yes	No (mainly a video editing tool)	Yes

The common denominator between all three of these coding languages and the libraries they use seems to be that they use FFmpeg. After researching several different libraries across several languages, FFmpeg keeps popping up as a reliable and efficient library to use for our video compression needs.

3.8.9 MCU Selection

We now know that certain codecs require more computing power than others. We also know that computer vision requires processing power as well. One of the most important decisions for this project is the MCU selection. It is what allows us to run our code efficiently and quickly.

3.8.9.1 First PCB Selection

ChatGPT was asked about three potential MCUs that could work well with our needs [GPTD]. The first MCU provided was an ESP32-S3. The second MCU provided was an STM32H7 Series. The final MCU listed was a Raspberry Pi Pico W (RP2040 with Wireless). I did have the initial idea to use the MSP boards from previous classes at UCF but after research, it was found that they are not powerful enough for our needs.

The first MCU to consider is the ESP32-S3 from Espressif. It is an “Xtensa 32-bit LX7 dual-core processor that operates at up to 240 MHz” [121]. It contains 512 KB of SRAM. It does well with neural network computing which is helpful with the computer vision aspect of the project. The chip also handles signal processing well. The storage temperature of this chip is between -40 °C and 150 °C [122]. It costs around \$1.85 at the current time of writing this [123].

The second MCU to consider is the STM32H7 Series. It is a “...32-bit Arm Cortex®-M7 core, running at up to 600 MHz” [124]. It can come in both single and dual core (Cortex®-M7 + Cortex®-M4). It offers flash memory between 64 KB and 2 MB. The chip comes with between 564 KB to 1.4 MB of SRAM. The storage temperature is between -40 °C and 125 °C [124]. I found chips from this series to be ranging from 7 to 10 dollars [125].

The third MCU is a Raspberry Pi Pico W (RP2040 with Wireless). It is a “Dual-core Arm Cortex-M0+ processor, flexible clock running up to 133 MHz” [126]. There are 264 KB of SRAM on board. It’s storage temperature is between -40 °C and 85 °C. On CanaKit, this cost around \$1.00 [127]. One thing to note about raspberry pi is that it known for its modularity when it comes to programming. We were also told by our advisors that raspberry pi boards are powerful.

After careful consideration, we decided to go with the brand raspberry pi. Raspberry pi was recommended by our advisors. It is known for its flexibility with programming. In addition, it’s very easy to find components that fit well with a Raspberry Pi. This includes things like the sensor for our project.

Table 3.7 Comparison table for different brands of MCUs

Parameter	MCUs		
Brand	ESP32-S3	STM32H7 Series	Raspberry Pi Pico W
Clock	240 MHz	600 MHz	133 MHz
SRAM	512 KB	564 KB to 1.4 MB	264 KB
Flash Memory	N/A	64 KB to 2 MB	N/A
Storage Temp	-40 °C to 150 °C	-40 °C to 125 °C	-40 °C to 85 °C
Cost	\$1.85	\$7 to \$10	\$1

We know that we want to use Raspberry Pi as the MCU for our project. Now we need to decide which raspberry pi to use that best suit our needs. Once again, ChatGPT was asked to provide potential raspberry pi boards that could fit our needs [GPTD]. The first raspberry pi board was a Raspberry Pi 4 Model B. The second raspberry pi board was a Raspberry Pi Zero 2 W. The third board suggested was a Raspberry Pi 3 Model B+.

The Raspberry Pi 4 Model B is a “Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8GHz” [128]. It has 1 GB, 2 GB, 4 GB, or 8 GB SDRAM depending on which model we choose. This board comes with H.265 decoding (at 4k 60fps) and H.264 encoding (1080p at 60fps) and decoding (1080p at 30fps) which is perfect for our needs. It comes with a variety of ports like USB ports, a USB-C connector, micro-HDMI ports and more. It has gigabit ethernet. The operating temperature is 0 °C to 50 °C [129]. The cost varies depending on the amount of on-board RAM you decide to get. It ranges from \$35 to \$75 on CanaKit [130].

The Raspberry Pi Zero 2 W is a “1GHz quad-core 64-bit Arm Cortex-A53 CPU” [131]. It comes with 512 MB of SDRAM. It comes with similar ports as the Raspberry Pi 4 Model B but since it is smaller, it has less ports. It comes with both H.624 encoding and decoding (both 1080p 30fps). The operating temperature is -20 °C to 70 °C [132]. The board cost on CanaKit is \$15 [133].

The Raspberry Pi 3 Model B+ is a “Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz” [134]. It comes with 1 GB of LPDDR2 SDRAM. It comes with H.264 decoding and encoding (both 1080p 30fps). It has gigabit ethernet but a maximum throughput of 300 Mbps. The operating temperature is 0 °C to 50 °C [135]. It was unavailable at CanKit but is \$35 dollars on Chicago Electronic Distributors [136].

All of these boards have camera connectors and H.264 encoding and decoding capabilities. However, The Raspberry Pi 4 Model B is a bit above the rest in terms of performance. It has gigabit ethernet and has more RAM options. It is the same price as the Raspberry Pi 3 Model B+ while having higher specs.

Table 3.8 Comparison table for different models of Raspberry Pi boards

Parameter	MCUs		
Model	Raspberry Pi 4 Model B	Raspberry Pi Zero 2 W	Raspberry Pi 3 Model B+
Clock	1.8 GHz	1 GHz	1.4 GHz
SDRAM	1 GB, 2 GB, 4 GB, or 8 GB	512 MB	1 GB of LPDDR2
Codec	H.265 decoding (at 4k 60fps) and H.264 encoding (1080p at 60fps) and decoding (1080p at 30fps)	H.624 encoding and decoding (both 1080p 30fps)	H.264 decoding and encoding (both 1080p 30fps).
Ethernet	Gigabit	N/A	Gigabit (300 Mbps throughput)
Operating Temp	0 °C to 50 °C	-20 °C to 70 °C	0 °C to 50 °C
Cost	\$35 to \$75 (depending on RAM)	\$15	\$35

3.8.9.2 MCU for Second PCB

The Raspberry Pi 4 Model B is a great board that is able to handle running OpenCV and H.264 for video compression. However, we need an MCU that we can use in our custom PCB. A custom PCB is a requirement in our project and requires a main processing unit to run our simple tasks. The Raspberry Pi 4 Model B is already a complete board. It can be used for some of the project, but we still needed an MCU for the custom PCB. Initially, our plan was to use the chip from the Raspberry Pi 4 Model B board. Unfortunately, the chip used in making the Raspberry Pi 4 Model B is not commercially available and is sold in bulk to companies to be used for manufacturing [GPTF]. This idea was no longer going to work because we had no feasible way of acquiring this chip.

We discovered an alternative to the Raspberry Pi 4 Model B. The solution was a Raspberry Pi Compute Module 4 (CM4). This board uses the same chip as the Raspberry Pi 4 Model B but comes without a PCB. This would allow us to utilize the powerful capabilities of the Raspberry Pi 4 Model B while allowing for modularity and customization of our PCB. The CM4 can include up to two camera connections which would allow us to implement the camera module selected for our surveillance system [GPTF]. We can also choose to have a microSD card in case we want to store video [GPTF]. This option would allow us to run python. This includes running OpenCV and FFmpeg efficiently [GPTF]. Most importantly, it comes in a small form factor that allows for custom PCB design (see Figure 3.16 below). There are no connections, and it is a bare PCB with the chip and some other components.



Figure 3.16. Image of Raspberry Pi Compute Module 4 [137]

The CM4 has a “Broadcom BCM2711 quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz” [137]. It has options for 1 GB, 2GB, 4GB, or 8 GB SDRAM. It comes with H.265 encoding at up to 4Kp60 and H.264 decoding at up to 1080p60 and encoding at up to 1080p30. It has options for “0GB (‘Lite’), 8GB, 16GB or 32GB eMMC Flash memory” [137]. It also comes with Bluetooth and wireless capabilities. A brief table can be shown below to show the specs.

Table 3.9 Specs for Raspberry Pi Compute Module 4. [138]

Parameter	Compute Module 4.
Clock	1.5 GHz
SDRAM	1 GB, 2GB, 4GB, or 8 GB
Codec	H.265 decoding (at 4k 60fps) and H.264 encoding (1080p at 60fps) and decoding (1080p at 30fps)
Ethernet	Not on board but capable of Gigabit ethernet when attached.
Operating Temp	-20° C to 85° C
Cost	Depends on how you customize it with flash memory and RAM. Average is around 40 dollars.

We reached out to our advisor, Dr. Lei Wei. We wanted to make sure that our choice for using the CM4 aligned with guidelines and to see whether it was the right choice. Unfortunately, we were told that it would not be wise to use the CM4 as well as the Raspberry Pi 4 Model B. We were told that if we used one type of board for everything in our project that we would run into time scheduling issues and that our project would “stop and go, out of your control”. We took his advice and started to look elsewhere for MCUs that would fit our needs for our custom PCB.

The original plan was to have our custom PCB being the first board to connect to the camera and then have the Raspberry Pi 4 Model B as our second board. However, due to having to change our plans, the first board connecting to the camera is going to be the Raspberry Pi 4 Model B and the second PCB is going to be our custom PCB. This PCB has the job of decompressing the transmitted data and displaying it to a display in real time. This allows us to use a simpler MCU that doesn't need as much computing power as the Raspberry Pi 4 Model B.

ChatGPT was very helpful in searching for MCUs because of how specific you can get with the information you give. It knows all about the scope of this project and the requirements of what we need to do. It gave us suggestions on what MCU to use for H.264 decompression. It suggested four different MCUs for consideration. Those MCUs are the ESP32-S3, the NXP i.MX series, the STM32 MP1 (STM32MP157), and the Raspberry Pi Compute Module 4 (of which we can rule out because of prior research) [GPTG]. The ESP32-S3 offers a fair amount of processing power. However, it's not the greatest option when it comes to real time decompression. Using particular software could make it viable but there are better options to consider. The NXP i.MX is popular in embedded systems and could work well with H.264. However they can be a bit complex in terms of power and pin count. STM32 MP1 utilizes both ARM Cortex-A and Cortex-M. This makes it very viable for H.264 video decompression. It suggested that we should either utilize the NXP i.MX or the STM32 MP1 (STM32MP157).

I did my own research to see if I could find a good STM32 MCU. I found the STM32MP157AAC3 and asked ChatGPT if this was a viable option. The chip is specifically designed for multimedia applications and should be able to handle H.264 decompression for our project. There are arguments to have for NXP i.MX and STM32 series. It's important to properly compare before we make any rash decisions.

The STM32MP157AAC3 includes "... an ARM Cortex-A7 core with multimedia support and a display controller capable of driving an LCD directly" [GPTG]. It also supports external memory which is handy for video data. The i.MX6ULL is part of the NPX series and also is able to handle multimedia applications. The STM32MP157AAC3 has low power consumption and is able to take on higher performance tasks. The i.MX6ULL is also power efficient but not quite as good at performing higher performance tasks. This is in part because the STM32MP157AAC3 is dual-core (Cortex-A7 + Cortex-M4) while the i.MX6ULL only offers a single core (Cortex-A7) [GPTG]. Due to the popularity of the STM series of MCUs, there are a lot of resources available to make working with it a lot easier. This will potentially make working with OpenCV and H.264 relatively painless. While still having resources, the i.MX6ULL will probably have less resources than the STM. The STM32MP157AAC3 is going to be more expensive than the i.MX6ULL but for good reason. It has twice the cores and is more suitable for multimedia applications. The i.MX6ULL is cheaper but at the cost of having slightly less processing power. The specs of the two processors can be seen below.

Table 3.10 Comparison table of different MCUs. [139], [140], [141], [142], [143], [144], [145]

Parameter	MCUS	
Model	STM32MP157AAC3	i.MX6ULL
Clock	800 MHz + 209 MHz (A7 + M4)	900 MHz
RAM	8-Gbit density (1 Gbyte), 16 or 32-bit LPDDR2/LPDDR3 or DDR3/DDR3L up to 533 MHz	16-bit LP-DDR2, DDR3/DDR3L
Ethernet	Supports Gigabit	Supports Gigabit
Codec	Does not come with codec but capable of H.264 decompression	Does not come with codec but capable of H.264 decompression
Cost	Around 15 dollars	Around 10 dollars

Initially, we planned to use the STM32MP157AAC3 chip in our custom PCB. However, the chip has a rather large footprint with hundreds of pins which can make designing the PCB a bit challenging. We went with another chip for the time being so that we could get a proper prototype up and running. We landed on the STM32H753VIT6. This chip has a smaller footprint while still being able to handle the load we're trying to put onto it. The specs for this chip can found below.

Table 3.11 Specs for STM32H753VIT6. [146], [147]

Parameter	Compute Module 4.
Clock	480 MHz
RAM	1 MB
Codec	Capable of H.264
Ethernet	Capable of ethernet
Operating Temp	-40° C to 85° C
Cost	\$16.66

3.8.10 Storage

One of our secondary goals is to implement a storage system for the surveillance system. If anything interferes with the transmission of data during free space, it would be useful to have a way of going back and viewing what was missed during transmission. There are many ways of storing or capturing data. In this section, we're going to do an in-depth analysis of how we can achieve this.

3.8.10.1 Challenges of Free Space Transmission

We very briefly discussed that a loss of data could occur earlier in the research process. Loss of data could occur due to something interrupting the free space transmission. Someone's hand could be waved in front of the transmission. In a realistic scenario, a bird could fly in front of the transmission. The long-term goal of this project is to have our project be applied in realistic and useful scenarios. For example, a secure transmission of data can occur from a drone recording while flying over a scene. Naturally, there are bound to be things that get in the way of the drone's transmission. That's why it's important to have a failsafe that captures the data before it is transmitted so that there can always be a way to view the data if there are problems with transmission.

3.8.10.2 Storage System Requirements

The storage system needs to have a few requirements. Storing video can quickly fill up storage. We need to figure out how much storage video takes up. Firstly, there are different sizes of video depending on different factors. The resolution of a video contributes to the size of a video. Obviously a larger resolution will require more data because there are more pixels in the video itself. Image resolution is typically talked about in terms of PPI (pixels per inch) [148]. Since we want to film in 720p, our pixel length will be around 1280 by 720 (this is standard in 720p resolution). More intense resolutions like 4k have pixel lengths of 3840 by 2160. This is not necessary when it comes to our project but it's important to understand how it works. In short, a higher resolution will take up more size because there are more pixels to be stored.

The frame rate of a video is another factor that is important to take note of. All forms of video utilize frame rate. Video can't actually simulate real motion. Instead, a video plays a certain amount of still frames in a second and our brain is tricked into perceiving motion [149]. Most movies are shot at 24 fps. This is enough that our brain will register "motion" and understand what is happening in a video. Every second, 24 still images are shown. At 60 fps, 60 still images are shown within a second. Higher frame rates provide a smoother video that feels less choppy. However, most video is shown at around 24 to 30 fps. Our project will be using 30 fps.

Why would we use lower frame rates when higher frame rates provide a smoother viewing experience? Higher frame rates require more storage to store and more bandwidth to transfer. 30 fps will give our surveillance system a smooth enough feed while maintaining a lower storage requirement and a lower bandwidth requirement to transfer. In Table 3.12, you can see the different requirements of storage compared to different kinds of resolution and frame rate. The storage requirements shown are using a calculator provided by the company Seagate. This company is known for their storage systems.

Table 3.12 Storage requirements for different resolutions and frame rate. [150]

Resolution and Frame Rate	720p 30 fps	720p 60 fps	1080p 30 fps	1080p 60 fps	4k at 30 fps	4k at 60 fps
Number of Cameras	1	1	1	1	1	1
Time recorded per day	1 hour	1 hour	1 hour	1 hour	1 hour	1 hour
Number of days stored	1	1	1	1	1	1
Video Quality	High	High	High	High	High	High
Codec	H.264	H.264	H.264	H.264	H.264	H.264
Storage used	1.16 GB	2.32 GB	2.78 GB	5.51 GB	10.39 GB	20.78 GB

As you can see from Table 3.12, different kinds of resolution and frame rates severely impact the amount of storage used. In our project, we are aiming to have a resolution of 720p and a frame rate of 30 fps. An hour of footage would take up roughly 1.16 GB of storage. If we want to store footage from the feed, we'll need to have a decent sized storage system. It would be wise to have at least a day's worth of storage. For application in real security systems, it would be wise to have a much larger storage system. This project is a prototype so 32 GB will be a fair amount of storage for us to store video.

3.8.10.3 Recording for Storage

How are we meant to record the video? It's not too difficult but it requires some programming. Let's say we have a webcam. We want to record what the webcam sees and save it as a video so that we can view it later. A video from YouTube shows a man coding using OpenCV to capture a short clip of his webcam [151]. We can use OpenCV (which we are already using) to record the webcam and store it for later. Our project already has the raspberry pi camera module. We can custom set the framerate and custom set the resolution.

3.9 Communication Sub-System

Signal communication is a critical component for the project. Without clear signals communication, the recorded data will not be displayed. Understanding what communication is, the different types of communication (i.e. interpersonal, electrical, and optical), and how to achieve communication is necessary. The following sections will explore these questions and highlight the decision-making for the communication sub-system for the project.

3.9.1 Communication

Communication is the form of relaying messages to people or groups. The relayed messages are the ideas conveyed by a sender, the communicator, to a receiver, the participant. The process for sending messages is known as transmission. This unique exchange of information is transmitted through different channels. A human channel uses verbal and non-verbal communication – verbal involves the exchange of spoken and written information while non-verbal happens without the exchange of spoken or written information. Technological advancements in analog and digital electrical systems have enabled new and distinct channels for communication, like copper cable communications, radio communications, fiber optic communications, and free-space optic communications. These avenues will be discussed in succeeding sections. Regardless of the channel of choice, the agreed roles between the sender and receiver dictates the style of communication to be one-way or two-way.

3.9.1.1 One-way Communications

One-way communication, also known as linear or non-transactional communication, is the form of transmission in which information flows in a single, linear direction from sender to receiver [152]. An example is shown in fig. 3.17.

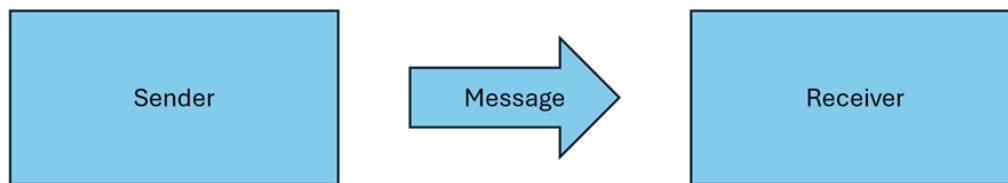


Figure 3.17. Linearity of one-way communication. Fig. created by Authors.

As seen on fig. 3.17, the sender controls the message by determining what information is shared, how it is presented, and when it is shared; the receiver's role is passive and has little influence over the communication process [153]. The sender transmits information without any feedback or response from the receiver, making the event non-transactional [154]. The limited interaction between sender and receiver severely restricts engagements, understanding, and trust; additionally, information is transmitted without consideration of the receivers' feedback [154]. One-way communication is critical for information dissemination, instructional purposes, persuasive influence, and entertainment like radio and motion pictures [152]. One-way communication is widely applicable to technologies like one-way video surveillance systems and notifications. Perfect examples include baby monitors and natural disaster alerts. In the case of a baby monitor, the sender is the baby while the receiver are the parents. The baby's actions dictate what the parents observe or hear giving them the necessary information to intervene on the baby or not. For natural disaster warnings, one-way communication enables the transmission of important information to the public with the goal of rapid dissemination. In this case, the sender is law enforcement, and the receiver is the publics.

3.9.1.2 Two-way Communications

Two-way communication is the form of transmission in which information is exchanged by both sender and receiver directly or indirectly [155]. An example is shown in fig. 3.18.

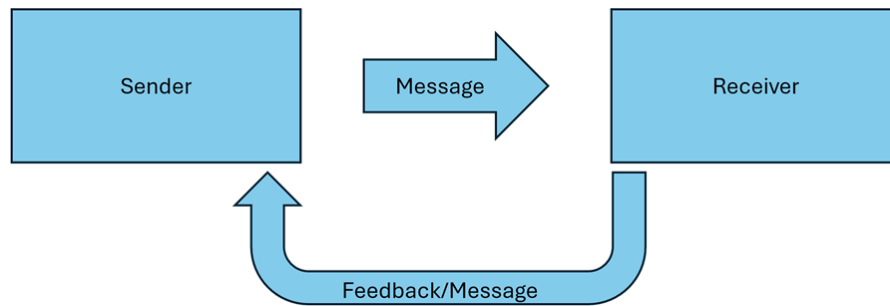


Figure 3.18. Direct (message) and non-direct (feedback) two-way communication. Fig. created by Authors.

As shown in fig. 3.18, in this style of communication, information flows first from sender to receiver then receiver to sender, creating a feedback loop between the sender and receiver; unlike one-way communication, two-way communication encourages interaction and transactions amongst sender and receiver [156]. The highly interactive nature of two-way communication, engagements, understanding, and trust is well established amongst sender and receiver. Two-way communication is the way humans communicate. Naturally, two-way communication is widely applicable to technologies that facilitate interpersonal interactions amongst people like two-way radio systems, telephones, and social media platforms. Two-way radios enable law enforcement officers and security teams to immediate back-and-forth communication; in this case, the communication channel is a low-frequency electromagnetic wave in air. Telephones operate under the same principle as two-way radio. Social media platforms enable users to communicate and interact through posts, comments, and likes; this style of two-way communication takes place different servers (one for the sender and one for the receiver) connected via fiber optic cables transmitting information using light pulses.

3.9.1.3 One-way Communications over Two-way Communications

This project intends to build a one-way communications surveillance system because receiver feedback is not necessary. The intended functionality of a surveillance system is to transmit information from the sender, the immediate environment, to the receiver, the end user, for their discretion. Two-way surveillance systems would complicate the scope of the project because it would involve the receiver providing the sender with some information. When analyzing the stretch goals and applications for the project, two-way communication is not necessary for military, first response, or agricultural surveillance. For example, a two-way channel for military surveillance could compromise the surveillance effort.

3.9.2 Electrical Communications

In communications, a channel is any way information is transmitted. Verbal channels transmit information using voice. Non-verbal channels transmit information using facial expressions, body language, or writing. Technological channels transmit information using analog or digital electrical signals via copper cables. Over the last 150 years, copper has proven to be a reliable conductor for transmitting electrical signals. Copper cables like coaxial cables, twisted pair cables, and ethernet cables have demonstrated copper can be used for short-haul and long-haul electrical communication but with limitations.

Coaxial cables are a type of shielded transmission line designed to conduct high frequency electromagnetic signals [157]. The structure of a coaxial cable consists of a central copper core, a metallic shield, a dielectric insulator, and a plastic jacket [157]. Invented in the 1800s by Oliver Heaviside, coaxial cables offered exceptional low loss and low inter-cable induction interference [157], [158]. The invention of the coaxial cable revolutionized the communications landscape of the past and continues excelling communications capabilities of today. Coaxial cables have been so reliable over the years their design hasn't changed much [159]. Some ways coaxial cables have enabled communication include television broadcasting, long-distance phone service, and internet infrastructure.

Data transmitted through coaxial cables are typically in the form of an electromagnetic radio frequency (RF) wave. RF signals are analog or digital electrical signals that oscillate at frequencies from 300 GHz to as low as 9 kHz, the frequencies associated with radio electromagnetic radiation [160]. Radio signals are typically associated with free-space propagation. However, radio electromagnetic disturbances radiate through shielded copper conductors, like a coaxial cable. The shield coating of coaxial cables is what maintains the RF signals inside the cable [161]. To create a distinction, telecommunication services refer to free-space RF signals as wireless RF and cabled RF signals as cabled RF [161]. It's good to understand a coaxial cable as being a good container for radio signals. RF signals originate at a transmitter and are picked up by a receiver.

Other cables like twisted pair cables are used in electrical communications. Twisted pair cables are widely used today to facilitate computer communication by means of ethernet. As a result, the most common twisted pair cables are more commonly known as ethernet cables. There are different types of ethernet cables in the market and each iteration is known as a category; the most recent advancement in ethernet cables is the Cat8 cable. These types of cables were first for use in local area networks and have quickly moved to be an integral part of day-to-day life. For example, ethernet cables are ubiquitous in homes and workplaces to connect devices and routers to the internet.

Twisted pair ethernet cables are designed to transmit high data rates using a pair of conducting material twisted around each other. This enables the two cables to positively interfere with each other with limited crosstalk between cables. Crosstalk occurs when two conducting materials are in proximity to each other and induce changes on the transmitted signal of each cable, degrading the signal to noise ratio at the receiver. Typically, these

cables have more than two twisted pairs of cables. Ethernet cables can be further improved by shielding each twisted pair that makes up the cable.

To facilitate faster communication speeds with lower latency, local area networks began migration towards optical communication after the dawn of the low-loss optical fiber.

3.9.3 Optical Communications

Electrical communication offered a good solution for many avenues of computer communication. Research and engineering efforts during the mid-1900s demonstrated the limitations of electrical communication and introduced and proliferated optical communication into the telecommunication industry. Optical communication has enabled today's world internet connectivity by means of long-haul optical fiber networks and data center inter-server connectivity. In this section, optical communication methods relevant to the scope of the project are discussed.

3.9.3.1 Fiber vs. Free-Space Communications

Over the last 30 years, the proliferation of internet traffic and the rapid development of the communication industry has optical communication as a cornerstone strategic industry to meet the world's connectivity needs [162]. Advancements in optical fiber technology have enabled high bandwidth long and short haul communication between networks and advancements in free-space optical systems have enabled satellite-to-satellite, air-to-air, and satellite-to-ground communication using lasers. These two methods of communication have positively impacted humanity and the communication industry by providing more effective alternatives for communication at higher bandwidths [163].

3.9.3.2 Fiber Communication and Limitations

Optical fibers are thin strands of glass, or plastic, with a light-guiding capability known as total internal reflection [163]. At a high level, total internal reflection is the complete reflection of light at some interface between two distinct mediums, like air and water; however, this type of reflection can only occur if light is in a denser region [Keiser]. In optical fibers, the two regions are glass of different optical densities, known as refractive index. These regions are known as the core and cladding with the core having a higher refractive index than the cladding [163]. Considering the principle of total internal reflection on an optical fiber, light is confined within the inner core region by the outer cladding region, as shown below.



Figure 3.19. Simplistic optical fiber schematic showcasing the refractive index difference between core and cladding and between core and cladding diameters. Figure by Authors.

The difference in refractive index between the core and cladding is the principal driver of two important optical fiber properties: numerical aperture (NA) and the normalized frequency parameter (V-number) [163]. The NA of an optical fiber describes the light gathering ability of the fiber [163]. The V-number is an optical fiber parameter that correlates to the number of modes guided in a fiber [163].

NA is important because it represents the physical limitation of an optical fiber. Not all light that is incident on an optical fiber is usable – only light that satisfies the NA requirement of a fiber is transmitted [163]. The NA is described by the total internal reflection requirement of the fiber which depends on the light's entrance angle or the refractive index difference between the core and cladding, as shown in the equation below [163].

$$NA = n_{environment} \sin \theta_{o,max} = \sqrt{n_{core}^2 - n_{cladding}^2}$$

When calculating NA using the maximum angle of acceptance for the optical fiber, the refractive index of the environment must be accounted for; this method is derived through Snell's law. Regardless, calculating NA by taking the difference between the core and cladding refractive indices is a more common and easily understandable term; this method is also derived through Snell's law [163].

Considering the dimensions of the fiber core, the light's wavenumber ($\frac{2\pi}{\lambda}$), and the NA, an important parameter connected with the fiber's cutoff condition is calculated: the V-number, as shown below.

$$V = kr_{core}(NA)$$

Since the V-number depends on the NA, it also depends on the difference between the core and cladding refractive indices. The V-number is used to calculate the number of modes that a fiber supports at a specific wavelength, the mode cutoff conditions of the fiber at a

specific wavelength, and the fiber's propagation constants [163]. The light that travels in optical fibers is in distinct optical modes. A mode is a spatial distribution of optical energy in one of more dimensions that remains constant in time [163]. Knowing the number of modes that propagate in a fiber is important because it discerns between single-mode and multi-mode optical fibers.

As stated in the name, single-mode and multi-mode fibers differ in the number of modes light can propagate in - single-mode fibers can only propagate one mode while multi-mode fibers propagate two or more modes of light. The mode content of an optical fiber directly depends on the value of the V-number because the V-number can be used to calculate the number of modes that an optical fiber can support per the following equation [163].

$$N_{MODES} = \frac{V^2}{2}$$

Since the V-number determines whether a fiber is single-mode or multi-mode, then so does the core size, refractive index difference between the core and cladding, and the NA, per the equation expansion below.

$$N_{MODES} = \frac{V^2}{2} = \frac{kr_{core}(NA)^2}{2} = \frac{kr_{core}^2 \sqrt{n_{core}^2 - n_{cladding}^2}}{2}$$

From this difference and current industry trends, single-mode fibers typically have core diameters of 9 μm while multi-mode fibers typically have core diameters of 50 μm with very similar refractive index differences [163]. However, it's typical for multi-mode fibers to have larger NAs than single-mode fibers.

Fiber optical communication has become a popular form of communication since it enables faster data transfer speed with less attenuation, limited inter-fiber interference, small size, and carrying capacity [164]. Regardless, fiber optic communication comes with its fair share of limitations that stem from its advantages, some being more obvious than others. Even though data transfer speeds in optical fibers are done at the speed of light, the signal still degrades over distance; this degradation is known as attenuation and determines the distance over which a signal can be sent without a repeater/amplifier [163]. Signal attenuation causes include loss (attenuation), stretching/broadening of the pulse in time (dispersion), and broadening of the pulse in spectrum (nonlinearity) [163].

As light travels along a fiber, its power decreases exponentially with distance, per the equation below.

$$P_z = P_0 e^{-\alpha z}$$

This makes attenuation a fundamental cost driver for telecommunication systems since an optical fiber signal must be amplified every few kilometers. The three main sources of fiber attenuation are material absorption, Rayleigh scattering, and bending loss [163]. Material absorption consists of material absorption by atomic defects in the glass composition of

the fiber, extrinsic absorption by impurity atoms in the glass, and intrinsic absorption by the basic constituent atoms of the fiber material [163]. Rayleigh scattering is any form of optical light scattering due to microscopic fluctuations in the material density leading to variation in the refractive index of the fiber on a scale smaller than a wavelength of light. Bending loss is any loss of signal associated with large or microscopic bends on the fiber; large bends are known as macrobends while very small bends are known as microbends [163]. Macrobends have bend radii that are large compared with the fiber diameter, such as those that occur when a fiber cable is bend around a corner, while microbends are random microscopic bends on the fiber axis that can arise when the fibers are incorporated into cables [163].

The stretching/broadening of a pulse in time is also known as dispersion. Dispersion will always occur in optical fibers because different “parts” of the light pulse travel at different speeds. The different “parts” of a light pulse that are dispersed are the spatial modes, wavelengths, and polarizations [163]. Spatial mode dispersion, also known as inter-modal dispersion or inter-modal delay, is only observed in multimode fibers and occurs when each mode in a fiber has a different group velocity at a given wavelength [163]. As a function of the data bandwidth and link length, inter-modal dispersion affects communication links by making consecutive signal pulses indistinguishable from the other [163]. The modal bandwidth of an optical fiber highlights the maximum tolerable bit rate for a specified link length where two consecutive pulses will remain distinguishable; this is calculated as shown below.

$$\text{Modal Bandwidth [MHz} \cdot \text{km]} = B \text{ [MHz]} \times L_{\text{Link}} \text{ [km]}$$

The modal bandwidth can be understood as a bandwidth limit on multimode optical fibers [163]. Dispersion by wavelength, also known as intra-modal dispersion or chromatic dispersion, occurs on all materials because the refractive index is a function of wavelength [163]. Light sources used in optical communications have some spectral width, known as linewidth for laser sources. Even if a monochromatic light source were to be used with an optical fiber, the moment the source is modulated to pulses, the pulse will increase in bandwidth because of intra-modal dispersion [163]. Chromatic dispersion is typically measured in $\text{ps} \cdot \text{nm}^{-1} \cdot \text{km}^{-1}$; this unit it known as the dispersion parameter [163]. Dispersion by polarization, known as polarization mode dispersion (PMD), occurs when different polarization of light within the fiber propagates at different speeds. Unlike chromatic dispersion, PMD varies along the fiber and is time-varying due to thermal and stress dynamics occurring along the fiber [163]. PMD is a measured in $\text{ps} \cdot \text{km}^{-1/2}$. Regardless, dispersion can be minimized or eliminated by engineering the refractive index profile between the core and cladding regions of optical fibers.

3.9.3.3 Free-Space Communication and Limitations

Since the proliferation of RF technologies in the mid-1950s, RF-based wireless technologies have enabled humans to communicate and be connected to one another from anywhere at any time without being physically connected to each other [165]. Phone calls once facilitated by coaxial cables now only rely on radio antennas and, sometimes,

satellites, and the term “wireless” has become synonymous to RF communication. The telecommunication RF spectrum window ranges from 30 kHz and 300 GHz of the electromagnetic spectrum; these heavily regulated bands are further divided into sub-bands that are publicly or privately owned for wireless telecommunication [165]. In recent years, the development and implementation of fifth generation (5G) telecommunication networks and its objectives (i.e. 10 Gbps data rates, supporting massive connectivity, reliability, availability) have increased the demand on the RF spectrum, decreasing the available supply for the public, making an RF spectrum bottleneck at some point in the future inevitable [166]. Additionally, the emerging concept of Internet-of-Things (IoT), promising wireless connectivity between all objects, will increase the strain on the RF spectrum within “smart cities” of the future [165], [166], [167]. With IoT-enabling connectivity becoming a lucrative area of research, expanding the telecommunication spectrum towards higher frequencies is under extreme consideration [165], [166], [167].

Expanding the telecommunication spectrum towards higher frequencies pushes the spectrum to include infrared, visible, and ultraviolet light frequency bands (300 GHz to 30 PHz), and leads into the dawn of optical wireless communication technologies [165]. Optical wireless communication (OWC) includes all methods and systems for computer communication that use light to transmit data wirelessly, instead of RF frequencies. Unlike optical fiber communication, light is not confined to a transmissive medium like an optical fiber; instead, light is transmitted in an unguided medium, like air, water, or the vacuum of space, by controlling the divergence and directionality of light within the OWC medium using optics, hence the name FSO communication [165].

OWC operating in the visible band (i.e. uses visible light) is known as visible light communication (VLC). OWC operating in the infrared band (i.e. uses infrared light) is known as infrared communication. OWC operating in the ultraviolet band (i.e. uses ultraviolet light) is known as ultraviolet communication (UVC). Besides OWC increasing the available telecommunication frequencies, OWC also enables ultra-high bandwidth, high directionality, robustness to electromagnetic interference, a high degree of spatial confinement (making each frequency highly reusable in small areas), and inherent physical security (tight confinement means small area for detection) [165]. In the OWC landscape, most investments have been focused on improving data rates and transmission distances, but has recently been used to enable point-to-point communication [166], [167], [168].

Point-to-point OWC systems enabling satellite-to-satellite, air-to-air, satellite-to-ground, or ground-to-ground communication using lasers are known as free-space optical (FSO) systems [165]. FSO systems have picked up traction across the telecommunication industry by offering high data rates per wavelength due to their decreased wavelength; the decreased wavelength associated with higher frequencies prevents electromagnetic waves from diverging too far at a fundamental level, enabling optical detectors to pick up higher incident power signals than the lower incident power RF signals [168]. An example of this is illustrated below.

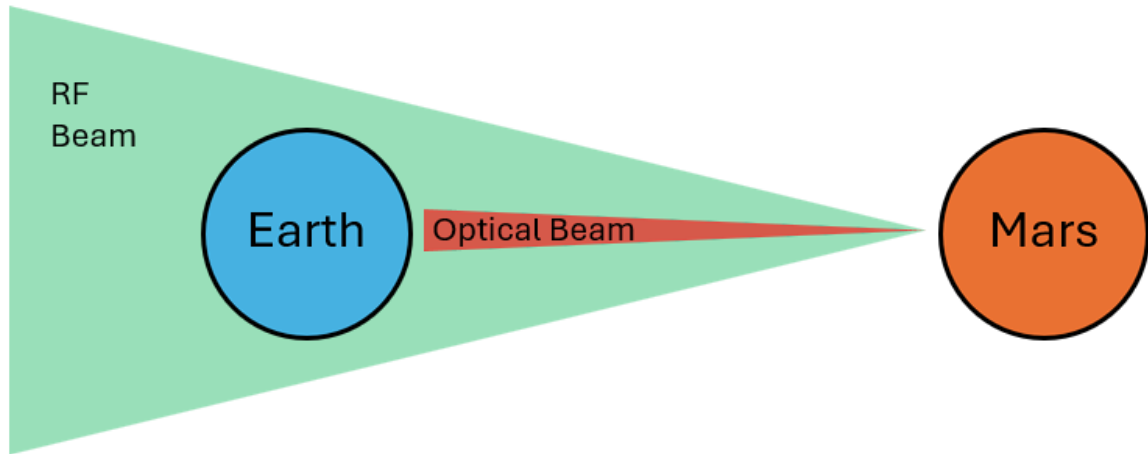


Figure. 3.20. *Simplistic electromagnetic beam divergence schematic comparison between an RF beam and Optical beam. Figure by Authors.*

The above figure illustrates an extreme case of an RF signal spreading out much more than a laser signal (referred to as FSO signal moving forward). Since RF and FSO signals are also used in space for satellite-to-satellite and satellite-to-ground communication, transmissions from Mars to Earth are common across all Mars exploration missions. According to NASA, a complete map of Mars RF signal from Mars can take roughly nine weeks to transmit back to Earth while an FSO signal would take about nine days [169]. This alone illustrates how FSO signals can be used to transmit 10 to 100 more data than RF signals [169]. Besides, FSO systems are estimated to offer around 1,200 times more bandwidth over the 300 GHz RF signal band, enabling further connectivity across all technological communication frontiers [166].

To understand how FSO systems enable ultra-high bandwidth, high directionality, robustness to electromagnetic interference, a high degree of spatial confinement (making each frequency highly reusable in small areas), and inherent physical security (tight confinement means small area for detection), the concepts of beam divergence and radiance must be discussed. Beam divergence, also plainly known as divergence, is the angular spreading of a signal beam with distance from the aperture or antenna from which it is emitted. Radiance is a measure of the incident radiation power unit solid angle and area; in other words, it's the power output radiated from a radiation source into a unit solid angle per unit emitting surface area.

Divergence is a concept that is intrinsic to all forms of electromagnetic radiation regardless of the transmissive medium. Divergence is directly caused by diffraction, a phenomenon classified as the spreading of electromagnetic waves as the waves pass through an aperture or around an object. The causes for diffraction are trivial and not relevant to the discussion of the project. From this definition, it's clear why divergence is important: controlling divergence enables the high directionality, high degree of spatial confinement, and inherent physical security of FSO signals. FSO system beam divergence is governed by the following equation.

$$\Theta = \frac{D}{f}$$

In this equation, theta is the quantified value for divergence, D is the diameter of the aperture, and f is the focal length of the lens used to generate an FSO system beam. An FSO system beam is known as a collimated optical beam. A collimated beam is any form of electromagnetic radiation with an ordered wavefront characterized by parallel rays from a geometrical optics perspective. Collimated optical beams are ideal for FSO systems because the beam has very low beam divergence; in other words, the beam does not spread too much in size over distance. To illustrate this idea, the figure below compares a collimated beam against a non-collimated beam.

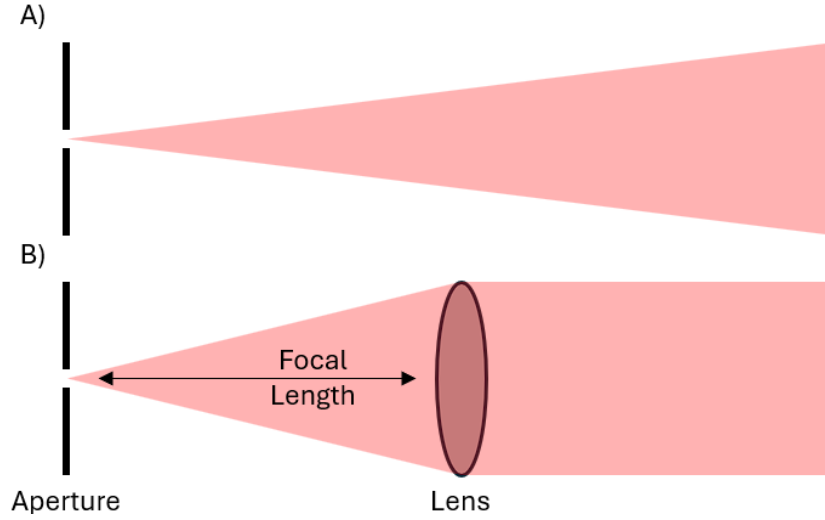


Figure. 3.21. *Simplistic representation of a (A) non-collimated beam diverging past the aperture and (B) collimated beam. Figure by Authors.*

From the illustration, it can be noted that a collimated beam can only be created when the source of the signal, in this case an aperture, is a focal distance away from the focal length of a lens.

RF signals also make use of collimated RF beams. However, diffraction goes beyond minimizing divergence by means of aperture and focal length – the wavelength of electromagnetic radiation is critical. As mentioned earlier in the imaging section, the highest achievable resolution of an imaging system is said to be diffraction limits under certain conditions. This diffraction limit is equally observed in communication systems [170]. The diffraction limit in any system is defined as the smallest realizable spot of a beam for a given wavelength [171]. The same applies in the reverse direction when taking the smallest realizable spot of a beam for a given wavelength and creating a collimated beam. The divergence of any diffraction limited spot is given by the following equation.

$$\theta = \frac{\lambda}{\pi r_{beam}}$$

In this equation, θ represents the radial diffraction-limited divergence, λ is the source wavelength, and r_{beam} is the radius of the smallest realizable spot [170]. From this equation, it becomes clear how a collimated RF signal can spread beyond the diameter of Earth on figure 3.19; a longer wavelength, which is exhibited by RF signals, naturally diverge much more than FSO signals composed of shorter wavelengths. The aspect of divergence also taps into the idea of intensity – a function of power per unit area. For the perspective of this discussion, the further the object is from the FSO source, the lower the intensity of the received signal will be because of the increased signal area. For this reason alone, is why collimated FSO signals persistently transmit faster data rates compared to RF signals, and it all ties in with the frequency because of the diffraction limit.

Radiance is a quantifiable parameter that reflects how much light is received by an optical system. When it comes to RF and FSO communications, radiance describes the amount of signal power received into a unit solid angle per unit of emitting surface area. In other words, radiance is the apparent brightness of a source as viewed from a particular position with a particular aperture. A solid angle is a three-dimensional parameter that describes a visible surface at a certain distance and angle from the center of a sphere. The solid angle can also be understood as a measure of the amount of field of view an object occupies as seen from a particular point by describing the spread of a surface in three-dimensional space. Solids angles are calculated using the formula below.

$$\Omega = \int \frac{1}{r^2} dA$$

This equation makes it clear a solid angle is a function of surface area (A) with respect to distance away from the source (r). Knowing what a solid angle is, radiance can be calculated using the following equation.

$$L = \frac{d\Phi}{d\Omega \cdot d(A_{\text{source}} \cdot \cos\theta)}$$

This equation describes radiance a function of radiant power (Φ), unit solid angle (Ω), and source area and observation angle. Expanding this equation deeper is beyond the scope of the project. However, it's important to note that radiance essentially becomes impartial to distance and, really, only depends on the viewing angle θ , making viewing angles critical for FSO systems.

With an understanding of what FSO communication is, the limitations of such systems must be discussed. Optical beam divergence has already been discussed in detail; however, its importance cannot be stressed enough. Making use of a well-collimated beam will limit the very large power losses associated with overfilling the detector at the receiver [172]. To effectively limit the divergence, a diffraction limited source in tandem with a high quality telescope must be employed [172]. Since FSO systems make use of an open transmissive medium, like air, they're susceptible to sunlight and, thus, background light; this radiation interference can decrease the signal-to-noise ratio at the receiver [172]. By making use of a large telescope at the receiver, as much light from the receiver can be

collected from the transmitter with high sensitivity and directionality [172]. Besides, high precision aiming systems are also necessary to transmit optical signals with high directionality; since transmissive mediums are highly dynamic, high-end FSO systems have built-in adaptive optics on the transmitter enabling communication to compensate for atmospheric turbulence and disturbances [172]. Atmospheric disturbances include clouds, dust, fog, changes in temperature, and changes in pressure. Above all, the system's power budget determines the overall effectiveness at communicating a signal; the power budget of any system describes how much of the signal is considered a tolerable loss to resolve the signal at the receiver [172]. With the power budget in mind, the remaining power at the receiver can be determined, giving insight into the possible data rate, modulation format, acceptable bit error rate, and the various noise sources of the system [172]. Power budget is an integral part of every communication system, so it's critical for it to be considered for this project.

3.9.3.4 Different Types of Fiber

The optical fiber market has many fiber products to choose from. Single-mode, multi-mode, and various specialty fibers are available. Understanding the fiber products available is important for the success of this project.

Single-mode and multi-mode fibers are the most common types of optical fibers. These types of optical fibers are typically step-index optical fibers; this name is derived from the refractive index difference between the core and cladding regions be defined by a step function, as shown in the figure below.

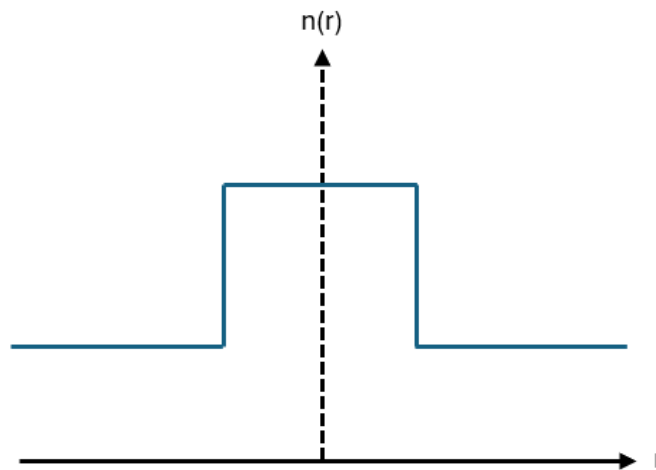


Figure. 3.22. Step-index profile by Authors.

To mitigate the dispersion limitations outlined in section 3.9.3.3 for multi-mode fibers, graded index (GRIN) optical fibers were engineered. GRIN fibers have a core-cladding refractive index profile that resembles a polynomial function; in other words, the core of a GRIN fiber changes according to a polynomial function while the cladding index remains constant. The polynomial equation that governs the change in refractive index and a figure illustrating the change is below.

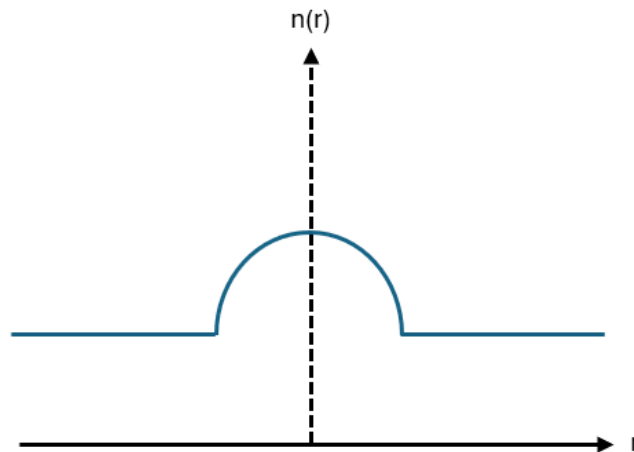


Figure. 3.23. Graded index profile by Authors.

Besides inter-modal dispersion, chromatic dispersion can also be mitigated on an optical fiber. This is accomplished by engineering a core-cladding refractive index profile using a variety of different functions. These refractive index profiles can be quite extreme, like shown below.

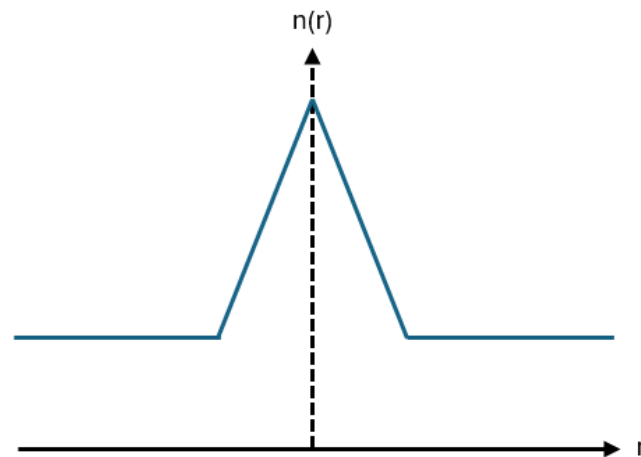


Figure. 3.24. Dispersion shifted refractive index profile by Authors.

In recent years, optical fibers resistant to physical bending have become available. This discovery has enabled the proliferation of optical fiber technology beyond long-haul networks and into the home. These fibers are a type of single-mode fiber with a tailored refractive index profile between the core and cladding, as shown below. This type of refractive index profiles is known as a moat or trench.

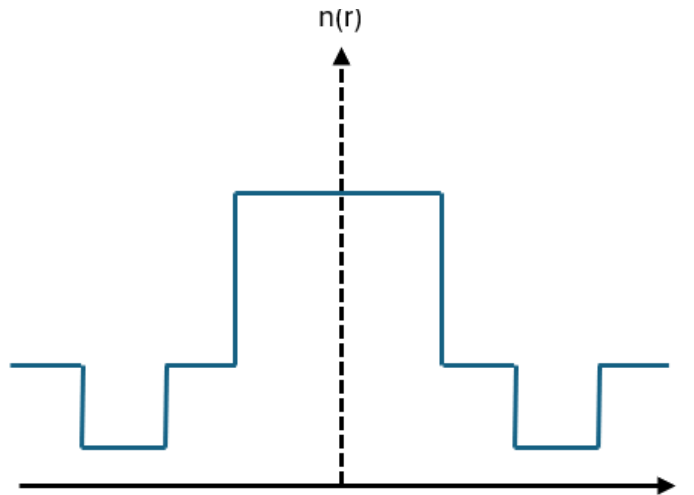


Figure. 3.25. Trench assisted step index profile by Authors.

Which optical fibers will be used for the success of the project depends on fiber NA and attenuation. All single-mode and multi-mode optical fibers have standardized losses, making loss a fixed parameter on our choice. Large NA is typical for multimode mode fibers. A large NA enables light to more easily be gathered by the fiber, making a free space to fiber link easier to accomplish. As a result, a multi-mode fiber will be used to complete the link.

3.9.3.5 Optical Transceivers

Optical modules, also known as optical transceivers or fiber optic transceivers, are the building blocks for all network connectivity inside and outside of data centers [162]. Optical transceivers have revolutionized the communication industry with high bandwidth communication capabilities. Some optical transceivers offer optical connectivity bandwidths, also known as data rates, up to 100 gigabits per second (Gbps) over link distances (transmission distances) of 10 km and beyond [173].

A transceiver is a modular hot-pluggable interconnect device that facilitates bi-directional optical communications using various wavelengths over different types of physical media [162], [174]. In general, optical transceivers are electrical-to-optical and optical-to-electrical converters – they have an electrical interface on the side that connects to the electronic system, like a PCB plug-in connection, and an optical interface that connects to the optical system, like a single mode or multimode optical fiber [162], [173]. Transceivers look like illustrated below in fig. 3.26A.

Optical transceivers have the functionality of both a transmitter and receiver, hence the name transceiver – data is transmitted via optical pulses from a laser diode or light-emitting diode and received using a photodiode [175], [176]. Laser diodes are more commonly used across the telecommunications industry and the reasoning will be explained in a later section. Other components that make up a transceiver include integrated functional circuits,

main control circuit board, housing, and optical and electrical interfaces [175]. A breakdown of all the components together is shown in fig. 3.26B.

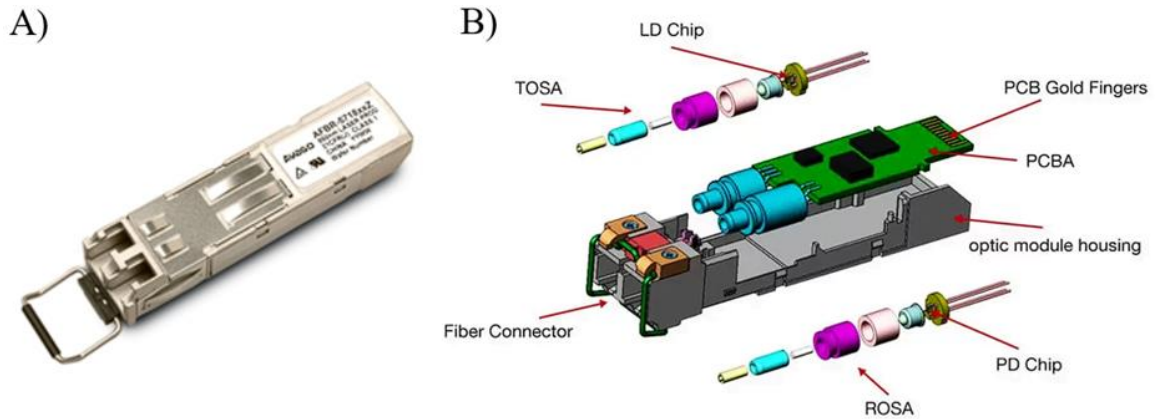


Figure 3.26. An optical transceiver and its breakdown. A) Image of optical module AFBR-5718PZ sold by Broadcom Inc. Fig. taken from Ref. [177]. B) General makeup of an optical transceiver. Fig. taken from Ref. [175].

The transmission side of the transceiver is usually housed in what is called a transmitter optical sub-assembly, or TOSA; the TOSA is responsible for converting electrical signals into optical signals [175]. When the TOSA's internal driver chip receives an input electrical signal, it drives the laser diode to emit an equally modulated optical signal equivalent to the input electrical signal [175]. Optical fiber modulation formats will be discussed in a later section. The receiver side of the transceiver is usually housed in what is a receiving optical sub-assembly, or ROSA; the ROSA is the essential component of any optical communication system because it's responsible for converting optical signals transmitted by the TOSA into usable electrical signals [175]. The work of the ROSA is mostly carried out by a photodiode – an optoelectronic device that is photosensitive and generates a current or voltage signal. When a photodiode is used in tandem with a trans-impedance amplifier and post-amplifier, the incident optical pulses then become a usable digital signal [175]. Both TOSA and ROSA systems integrate optical components, like lenses, necessary to couple light into an optical fiber (TOSA) and focus an optical fiber output to the photodiode (ROSA).

Today, telecommunication connectivity is facilitated by an optical fiber network and a pair of transceivers [178]. Consider a one-way communication computer connection, like shown in fig. 3.24.

To simplify this picture, assume you have a pair of computers – computer-1 is transmitting information via a fiber optic channel to computer-2. Each computer has a transceiver to transmit and receive information but only computer-1 is transmitting and only computer-2 is receiving. When Computer-1 is ready to transmit a signal, it generates an electrical signal on the electrical interface of its transceiver to drive the laser diode and generate optical pulses that represent the electrical signal – this is electro-to-optical conversion [175], [178]. This optical signals travels down the length of the optical fiber onto the transceiver on computer-2. The received optical signal induces an electrical response on the photodiode

generating an electrical signal for computer-2 to interpret – this is opto-to-electrical conversion [175], [178]. This happens millions of times between data center server modules. Data centers are famous for their extreme number of connections and cable links between server modules. Server modules have tens of transceivers working independently of one another and are in bi-directional communication with another server module since transceivers offer that capability.

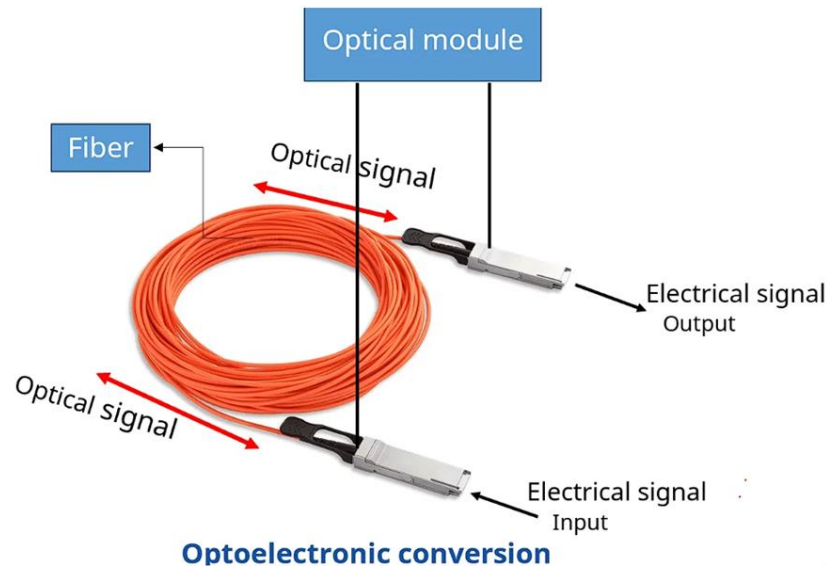


Figure 3.27. An optical fiber interfaces with an optical transceiver. Fig. taken from Ref. [175].

Transceivers are differentiated across a plethora of different classification methods. These classification methods often describe a specific working parameter of interest. The most common classification methods include form factor, data (transmission) rate, wavelength, light source, fiber type, transmission distance, modulation format, communication mode, receiver type, connector type and pluggability [175].

Classification by form factor, also known as package type, is by far the most popular because it determines the additional equipment to be used; for example, Broadcom Inc. lists optical transceiver products by form factor on their website [173]. This method of classification pertains to the size, shape, and overall dimensions of the optical module [175]. As is discussed in a later section, standards and regulations, the dimensions of all optical transceivers are closely monitored by a standards committee and a multisource agreement [175], [179]. Some package type examples include small form-factor pluggables (SFPs), enhanced-SFP (eSFP), SFP-plus (SFP+), SFP28, X form-factor pluggable (XFP), quad small form-factor pluggable (QSFP), C form-factor pluggable (CFP), QSFP28, and QSFP-DualDensity (QSFP-DD) [179]. A few of these are shown below on fig. 3.25.



Figure 3.28. Examples of different form-factor modules. A) SFP+ module. B) QSFP+ module. C) QSFP-DD module. Fig. created by Authors of images collected from Ref. [175].

Classification by data rate is almost equally as popular as form factor. Transceiver transmission rates have only increased since the release of the gigabit interface converter (GBIC), the first small (relatively large compared to SFP) form-factor transceiver capable of reaching speeds of 1 Gbps for ethernet and up to 4 Gbps for an optical fiber channel, in 1995 [175], [179]. The optical modules that flood today's market provide optical connectivity at rates up to 100 Gbps and beyond [173], [179]. The first big jump in transceiver transmission rates occurred in 2006 with the release of SFP+, the still dominant industry format; this form factor enabled minimal rates of 8 Gbps and up to 16 Gbps [179]. In 2014, QSFP28 became the standard interface of choice for high-speed, large-volume data transmission; this form factor had four transmission lanes operating at 25 Gbps with slightly larger physical dimensions than an SFP [179]. The fastest achievable data rates per transceiver now reach 800 Gbps with a maximum of 400 Gbps per transmitter-receiver pair; this became available in 2019 with the release of QSP-DD transceiver [173], [179].

The next two classifications go hand in hand – wavelength and light source. As discussed earlier, the light source housed within an optical transceiver is critical for a transceiver to transmit data and accounts for a large sum of the cost to buy a transceiver [180]. In general, the light sources used for optical fiber communication are either laser diodes or light-emitting diodes (LEDs), both made from semiconductor material. A laser diode is any direct bandgap semiconductor material that amplifies generated or input light by means of stimulated emission of radiation (photons). An LED is any direct bandgap semiconductor material that generates light by means of spontaneous emission of radiation. Both processes generate light from the recombination of electron-hole pairs in the semiconductor material. Regardless, what also differentiates a laser diode from an LED is the concept of optical feedback and optical coherence. Optical feedback is a laser process that utilizes generated light to generate more light. All lasers, whether semiconductor or solid-state, achieve optical feedback by means of a reflective optical cavity; these cavities have a type of reflective material that feeds some optical energy back into the gain material while letting the rest leave the system. Optical coherence refers to the phase difference between the electromagnetic waves that make up the generated light. LEDs generate incoherent light; this type of light spreads away (diverge) from a straight path making LED light non-directional. Lasers generate highly coherent light; this type of light is tightly confined and directional enabling photons to travel together without spread over long distances. Besides, laser diodes and LEDs emit narrow and relatively broad, respectively, spectrums of light centered about a wavelength. The central emission wavelength is determined by the semiconductor material bandgap energy since the emitted photon energy is equivalent to the bandgap energy following an electron-hole recombination event. The light sources that

make up the transmitted of optical modules include Fabry-Perot lasers (FPL), distributed feedback lasers (DFB), vertical-cavity surface-emitting lasers (VCSEL), LEDs, and electro-absorption modulated lasers (EAM) [175]. Which light source is used depends on the link communication link length, transmission rate, and optical fiber being used (single mode vs. multimode optical fibers) [175], [180]. Unlike VCSELs, like shown in fig. 3.26, FPLs and DFBs are edge-emitting semiconductor lasers, also known as in-plane lasers [180].

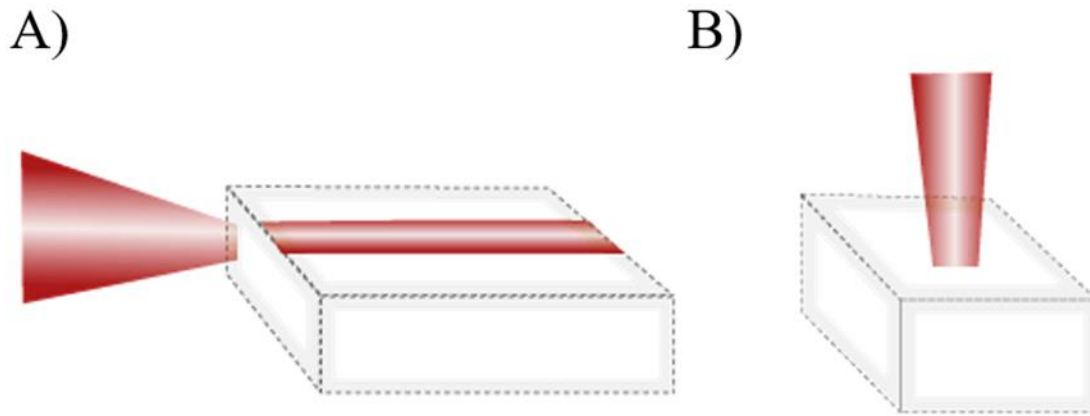


Figure 3.29. A) Edge-emitting semiconductor laser. B) Vertical-cavity surface-emitting laser. Fig. created by Authors of images collected from Ref. [180].

EAMs are a different type of semiconductor laser diode in which it has a built-in modulator [180]. The most common transceiver central wavelengths are 850 nm, 1310 nm, and 1550 nm. These wavelengths are fundamental wavelengths in the optical communications industry because of the low material loss associated with these wavelengths on silica glass, the material that makes up optical fibers. From these three wavelengths, only VCSELs are used to generate 850 nm light [180]. Additionally, LEDs were used in the early years of optical communications; however, the invention and proliferation of VCSELs has taken over the 850 nm industry that LEDs once dominated [181]. Today, FPLs and DFBs are found in most optical transceivers because of their narrow spectrums, low power demands, and highly coherent nature [175], [180]. The coherence of lasers and the central wavelength spread of laser diodes enable transceivers to push transmission capacity and optical fiber limits using modulation methods discussed later.

Transceiver classification by fiber type and transmission distance go hand in hand. Transceivers made to work with SMFs are known as single-mode transceivers and transceivers made to work with MMFs are known as multi-mode transceivers [182]. As discussed earlier, SMFs enable long-haul communication over tens of kilometers whereas MMFs are bandwidth limited for long-haul communications and are, thus, better suited for short-haul communication spanning several meters. The main forces affecting the transmission distance are dispersion and attenuation; dispersion refers to the spread of colors (intra-modal) or the spread between light modes (inter-mode) while attenuation refers to material absorption and scattering [175]. Transceivers are hence optical fiber limited in this classification duality – the transmission distance of a transceiver depends on

the fiber of choice. Additionally, the optical fiber of choice also limits the light source of the transceiver. SMFs typically operate around 1310 nm and 1550 nm central wavelengths, commonly known as the O-band (original band) and C-band (conventional band) in telecommunications, and MMFs operate around the 850 nm central wavelength, called 850 nm band for multimode fibers [175], [183]. Since SMFs are manufactured to operate at 1310 nm and 1550 nm, the light sources of choice are FPLs and DFBs while the light sources of choice for a MMF are LEDs and VCSELs [180]. When choosing between single-mode and multi-mode transceivers, the cost of multi-mode transceivers will always be cheaper [182]. The other classification types, except modulation formats, are not discussed further since they're not as popular and are not relevant to the scope of the project.

For transceivers to reach ultra-high transmission rates, like the 400 Gbps transmission speeds of a QSFP-DD transceiver, the transmitted optical signal must be modulated, also known as encoded, with respect to the input electrical signal. There are two modulation methods: intensity modulation, also known as pulse-amplitude modulation, and coherent modulation, also known as quadrature amplitude modulation [184], [185].

Intensity modulation can be achieved using one of two ways: non-return-to-zero modulation (NRZ), also known as pulse-amplitude modulation 2-level, and pulse-amplitude modulation 4-level, also known as PAM4 [185], [186]. NRZ is a two-level modulation technique that encodes digital signals in binary logic, either a one or a zero, as two different intensity levels; put simply, NRZ establishes logic 0 to be a lower intensity than logic 1 [185]. PAM4 is a four-level modulation technique that encodes digital signals in 2-bit logic; put simply, the intensity waveform has four different levels with each level corresponding to one of either logic 00, 01, 10, or 11 [185]. An example of the intensity difference is shown in figure 3.29.

Figure 3.30 does a great job at highlighting one of the main differences between NRZ and PAM4: bit rate. NRZ only enables 1 bit per clock cycle; for this reason, the bit rate will always be the same as the baud rate [186]. Whereas PAM4 enables 2 bits per clock cycle, doubling the bit rate for a given baud rate [186]. Without having to change many components, PAM4 doubles the amount of data that is transmitted in each channel by using four levels of intensity. Conveniently, signals loss is tremendously reduced in PAM4 because of the doubled bit rate at the same baud rate [186]. However, PAM4 comes with a signal-to-noise ratio and bit error rate cost. By increasing the encoding levels, the intensity difference between levels decreases; this increases the link budget penalty making PAM4 sensitive to noise resulting in higher bit error rates [185], [186]. In general, a transceiver modulating signals with PAM4 will always be more complex and consume higher power than an NRZ transceiver [185].

Coherent modulation can only be achieved through quadrature amplitude modulation (QAM), a technique that modulates the amplitude and phase of light [187]. QAM amplitude modulation is the same as NRZ or PAM4 – the intensity of the optical pulse is varied [187]. The phase modulation aspect of QAMs is more complicated – each optical pulse waveform has a different phase from each other [187]. Like intensity modulation, multi-level phase modulation can be achieved to create modulated signals like PAM4, but that differ in phase

rather than amplitude. An NRZ equivalent modulation in phase is called binary phase shift keying (BPSK) and a PAM4 equivalent modulation in phase is called quadrature phase shift keying (QPSK) – either keying technique falls under the umbrella of phase shift keying digital modulation formats [187]. QAM nomenclature is determined by the evaluation of M , the number of different waveforms possible, where n is the number of bits that make up the transmission symbol (i.e. 01, 0101, 101010, etc.); for example, 16-QAM tells us there are 16 different waveforms for a transmission symbol made up of 4 bits (i.e. 1010) [187]. The modulation levels of QAMs are observed using polar coordinates in a constellation diagram, like shown in figure 3.30 [187].

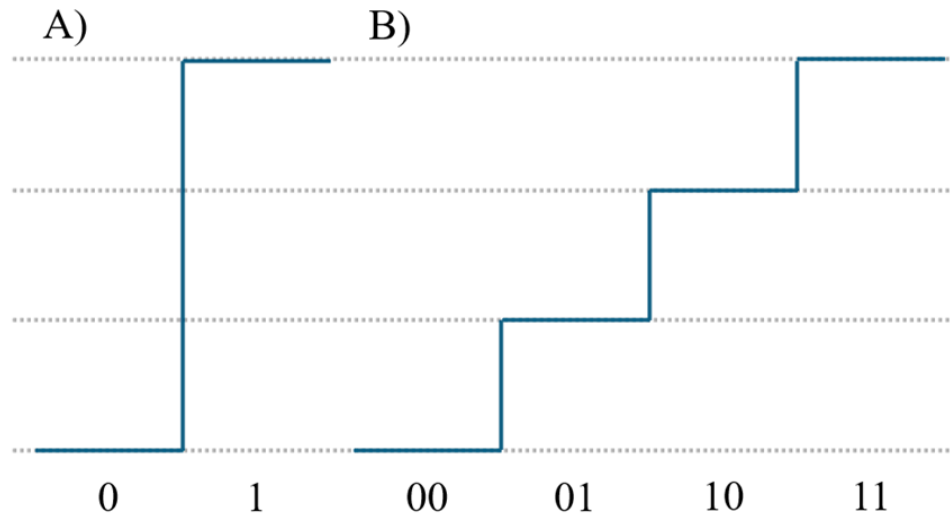


Figure 3.30. Logic difference between NRZ and PAM4. A) Intensity logic of NRZ. B) Intensity logic of PAM4. Fig. created by authors.

QAMs have become ubiquitous in radio communications; for example, WiFi 5 uses 256-QAM whereas WiFi 6 uses 1024-QAM [187]. In optical communications, QAM has become the method of choice for long-haul data links, like trans-ocean optical fiber networks [188].

Other methods of increasing the data capacity per channel exist (i.e. wavelength-division multiplexing techniques and its derivatives) but are not relevant to the scope of the project.

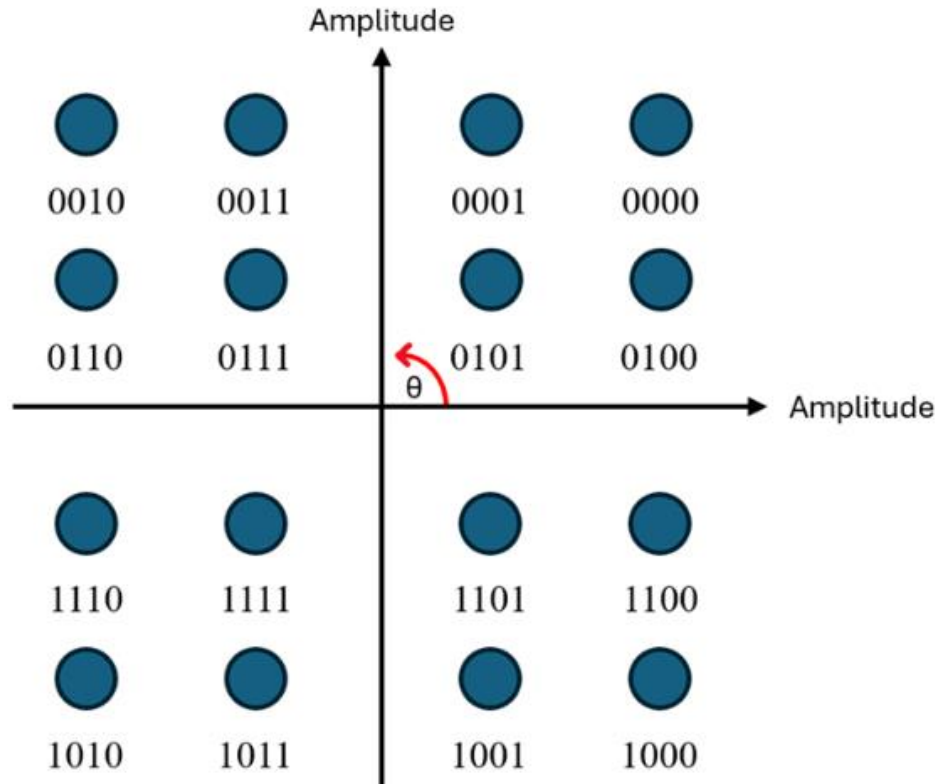


Figure 3.31. Constellation diagram of a 16-QAM modulated signals. Fig. created by authors.

3.9.3.6 Optical Transceiver Parameters of Interest

To effectively transmit a high-resolution video feed, knowing the transmission rate of a transceiver is important and must be near the order of GHz. The maximum transmitted power is necessary to calculate the associated optical losses across all the mediums in the communication channel. Receiver sensitivity establishes the sensitivity of the project's communication channel; the lower the sensitivity, the lesser the constraints are. Wavelength will determine the optical components used throughout the communication channel. Fiber transmission distance is a parameter found in the data sheets of all transceivers; this parameter serves as a reality check for the system since it gives insight into maximum transmitted power, sensitivity, and type of fiber to use. Fiber type is an important parameter: it gives insight into the type of fiber used during the testing process of the devices. Form factor constrains the evaluation board used. Cost is necessary to account for the budget.

For the scope of the project, the transceiver parameters of interest, in order of most to least important, include (1) transmission rate, (2) maximum transmitted power, (3) receiver sensitivity, (4) wavelength, (5) fiber transmission distance, (6) fiber type, (7) form factor, and (8) cost. The trade-offs between these parameters determine the cost. The most valued option will be closely considered, and potentially used, for the success of the project.

3.9.3.7 Optical Transceiver Comparisons

In this section, different transceivers from the same manufacturer, across manufacturers, and across fiber types will be compared. First, the differences between form factors of the same brand, Cisco, are compared below.

From the comparison on table 3.13, optical transceivers across form factors generally have similar values for many of their parameters, except transmission rate and cost. Table 3.9 only serves to demonstrate differences amongst transceivers. As transmission rate increases, the cost increases tremendously. For example, a 10 Gbps increase from 1 Gbps transmission rates doubles the cost; but a 40 Gbps rise in transmission rate increases the cost by a couple hundred dollars. Since the project does not benefit from ultra-fast transmission rates (10 Gbps and above), then SFP form factor transceivers with transmission speeds of 1 Gbps will be looked at more closely in table 3.14.

Table 3.13 Differences across different module form factors of the same brand.

Parameter	Optical Modules			
Mfg. P/N	GLC-LX-SM-RGD	SFP-10G-LR	QSFP-ELR4-40G	QDD-PLR8-800G
Standard	1000Base-LX/LH	10GBase-LR	40GBase-ELR4	800GBase-PLR8
Form Factor	SFP	SFP+	QSFP+	QSFP-DD
Transmission Rate (Gbps)	1.25	10.3125	41.2 (4x 10.3)	850 (8x 106.25)
Tx Power (dBm)	-9.0 - -3.0	-8.2 – 0.5	-4.5 - -2.3	-1.9 – 4.8
Rx Sensitivity (dBm)	< -21	< 14.4	< -12.5	-8.2
Wavelength (nm)	1310	1310	1310	1310
Transmission Distance	10 km	10 km	20	10 km
Fiber Type	SMF	SMF	SMF	SMF
Cost (\$)	10.00	27.00	319	1899.00

Table 3.14 Differences across SMF and MMF SFP modules of different brands.

Parameter	Optical Modules				
Brand	Cisco	Juniper	Arista	Cisco	Juniper
Mfg. P/N	GLC-LX-SM-RGD	QFX-SFP-1GE-LX	SFP-1G-LX	GLC-SX-MM-RGD	QFX-SFP-1GE-SX
Standard	1000Base-LX/LH	1000Base-LX	1000Base-LX	1000BASE-SX	1000BASE-SX
Transmission Rate (Gbps)	1.25	1.25	1.25	1.25	1
Tx Power (dBm)	-9.0 - -3.0	-9.5 - -3.0	-9.5 - -3.0	-9.5 - -3	-9.5 - -3
Rx Sensitivity (dBm)	< -21	< -20	< -20	< -17	< -17
Wavelength (nm)	1310	1310	1310	850	850
Transmission Distance	10 km	10 km	10 km	0.55	0.55
Fiber Type	SMF	SMF	SMF	MMF	MMF
Form Factor	SFP	SFP	SFP	SFP	SFP
Cost (\$)	10.00	8.00	8.00	9.00	9.00

From the comparison on table 3.13, it's very apparent that 10 km, SMF, 1310 nm and 550 m, MMF, 850 nm SFP transceiver modules have relatively standardized parameters. Transmission rates are at least 1 Gbps, transmission optical power ranges between -9.0 to -3.0 dBm, and photodiode optical power sensitivity is below -20 and -17 dBm for SMF and MMF modules, respectively. The cost of an SFP module is very cheap too. Since which SFP module is chosen for the success of the project is, then, not a trivial question.

Between SMF and MMF modules, certain SMF modules can be adapted to fit MMFs [189]. This makes SMF transceivers more versatile than MMF modules for the scope of the project. However, the small core size of SMF fibers adds an extra level of difficulty when coupling free-space light. Regardless of the difficulty, SMFs are a better option than MMFs because of matters discussed earlier and to be discussed later. Due to the limited difference and very low-price difference between Cisco, Juniper, and Arista transceivers, either will suffice the scope of the project. However, the Cisco-branded transceivers are the most available, making it a good choice for the project.

3.9.3.8 Optical Transceiver Evaluation Boards

Optical transceiver evaluation boards are any electronic component that translates input and output electrical signals to and from the optical transceiver. These boards are ubiquitous and can get expensive depending on the transceiver of choice. Optical transceivers operating at 1 Gbps and 10 Gbps are typically operating through NRZ modulation formats making input and output electrical signals easier to work with. Evaluator boards for 1 Gbps and 10 Gbps optical transceivers are relatively inexpensive and require an ethernet connection. These boards feature an SFP or SFP+ edge connector, an SFP or SFP+ cage assembly to connect the optical transceiver, and RJ45 ethernet cable connector.

3.9.3.9 Long-Haul FSO Simulation

Long-haul FSO systems are expensive and very much an area of tremendous research in academia. For example, the National Aeronautics and Space Administration is investing millions into satellite-to-ground, ground-to-satellite, and satellite-to-satellite FSO systems for space exploration [190]. SpaceX is deploying hundreds of internet satellites with satellite-to-satellite FSO capabilities creating a global satellite constellation – each Starlink satellite is equipped with three lasers offering 200 Gbps inter-satellite communication [191]. To make this project a project similarly as interesting, various simulation experiments will be employed to achieve long-haul communication – three will be described in this section.

Over short distances, a well-collimated beam does not significantly change in size. To approximate the extreme divergence of long-haul FSO systems, a negative lens can be placed along the FSO axis to induce beam divergence. Using the divergence equation mentioned earlier, link length with respect to beam size can be calculated; in other words, how much the beam spreads over, say, 10 km can be calculated. With a negative lens, this beam diameter can be approximated, and the system's power budget can be put to the test. Ideal collimated beams cannot theoretical diverge; however, the dynamics of the transmissive medium made jitter the spot on the receiver. A jitter test qualitatively measures the system's susceptibility to jitter at the far-field. This jitter simulation approximates the high precision of high-end FSO systems with respect to atmospheric dynamics, and all it needs is a 4F imaging system. Lastly, the system's power budget with regards to atmospheric attenuation can be simulated using neutral density filters. A neutral density filter absorbs a percentage of light incident on it over its depth; a similar phenomenon occurs in atmospheric mediums but over very long distances.

3.10 Different Types of Power Supplies

This section covers necessary discussions encompassing the power system unit that will power the project. Different power supplies, like Mean Well and RS Pro, are compared in length.

3.10.1 Mean Well

Mean Well is a reputable brand founded in 1982 that can be recognized for its power supplies that are high-quality and reliable. With a strong standing in the industry from a solid reputation, Mean Well's products are used commonly across a range of applications, allowing us to become one of the leading manufacturers of standard power supply products. Industrial automation, LED lighting, and communication systems are just some of those applications. Mean Well is known for producing highly efficient and durable products for electrical needs.

Mean Well has a broad collection of products that are known for high efficiency, boasting a robust design and extensive protection features. The efficiency of some of their models boasts up to 96% while having a wide input voltage range. The power supplies can protect against short circuits, overloads, and over-voltages. Mean Well also includes long warranties for their products, where some can range up to 7 years. On top of this, Mean Well products comply with different international safety and quality standards like the Underwriters Laboratories (UL), Technischer Überwachungsverein (TUV), Conformité Européenne (CE), and more, ensuring that their products are reliable and safe for the multiple uses in the industry.

3.10.1.1 Mean Well LRS-50-5

The LRS series of products under Mean Well's stock offer compact and cost-effective power supplies that boast high efficiency and reliability. As an enclosed style power supply, it holds the AC-DC circuitry with a metal mesh frame, having the connections made with screw-in terminals. Compared to other power sources within Mean Well's repertoire, this is the easiest to work with outside of the GST series. The LRS delivers the best price-to-performance solution on their site while having a full range of protection functions [192].

The Mean Well LRS-50-5 is a 5V power supply unit with a 10A output current that boasts a high reliability and efficiency (up to 90%) while reducing energy consumption and heat generation, which is very helpful for a project like this with a lot of sensitive components that can be damaged easily. In addition to a 90% high efficiency, the metallic mesh case enhances heat dissipation while operating in a -30°C to +70°C temperature range under air convection without a fan [193]. Without the need for a fan, the noise generated by the unit is reduced. Mentioned previously, the LRS-50-5 boasts high reliability, but it has a mean time between failures (MTBF) of 645K hours minimum [194], making it a suitable long-term power source for a security system. With one of the applications listed for this unit as security systems, therefore, it could be a good choice for our product. Also, it is designed to operate in humid conditions, which is important since we are based in a humid area.

3.10.1.2 Mean Well RS-15-5

The Mean Well RS-15-5 comes from the RS-15 series that offers a highly reliable, compact AC/DC power supply. Owning a small footprint with high power density is a suitable choice to combat space constraints depending on the application. Like the LRS, it is an enclosed style power supply, with that similar metal mesh frame. The RS-15-5 can also operate at a range of temperatures from -20°C to +70°C, while also utilizing open air convection for cooling, making it suitable for a variety of environments. Mean Well products boast the inclusion of various protections against electrical complications and this unit is no exception.

The RS-15-5 offers a high efficiency which is around 77%, reducing energy consumption and heat generation. On top of that, with a MTBF of 108.8K hours minimum [195], this unit offers long-term reliability that is essential for a product focusing on security, allowing it to run for long periods at a time while ensuring electrical protection.

3.10.2 RS Pro

RS Pro is a distributor and manufacturer of electronic components founded in 1937 and has grown to be one of the largest distributors in the industry. It is known for its extensive range of products as well as its reliable service, garnishing it as a very reputable brand. These products include electronic, electrical, and industrial components as well as tools and measurement equipment. Having a global reach and appearing as a one-stop shop for an engineer or technician's needs, its online platform allows customers to easily browse and find any products that suit their applications.

In addition to their extensive product range, RS Pro vaunts comprehensive customer service and support. Providing technical support and assistance, customers can find what they need. Based on some reviews, customers have noted that order fulfillment and delivery times had some issues, but with customer support aiding, those issues are resolved in a timely manner.

3.10.2.1 RS Pro 5V DC Power Supply

The RS Pro 5V DC Power Supply or Embedded Switch Mode Power Supply is a 5V output voltage and 7A output current PSU that has an output power of 35W. It is an enclosed configuration that has been designed to offer high efficiency, long life and high reliability for various applications [196]. Although this PSU has a smaller operating temperature range compared to other PSUs, from 0°C to +45°C, it is still a wide enough range where it can operate in humid environments, allowing a variety of environments. There are also built-in protections against complications and burnouts while still meeting various safety standards to ensure it is safe to use in household and industrial applications.

Based on this product's datasheet, this PSU boasts an efficiency of up to 82%, putting it on the lower end of the PSUs being compared but it still allows for a lower energy

consumption and heat generation. Like both Mean Well PSUs, the RS Pro is an enclosed configuration which allows it to use open air convection to cool itself, promoting a long-lasting life.

3.10.3 CUI Inc.

CUI Inc. is a well-known manufacturer of electronic components and power supplies founded in 1989, establishing itself as a leading supplier for the Original Equipment Manufacturer (OEM) market. Focusing on creating ease of access to power supply products and information, they aim to improve the way power products are integrated into designs. Like Mean Well and RS Pro, CUI Inc. possesses a wide variety of power supply products as well as other electrical components. These products are designed to meet a range of applications, whether that be industrial automation, medical devices, or consumer electronics.

CUI Inc. products are designed to meet high efficiency standards as well as comply with international quality standards, including DoE Level VI as well as UL, Puget Sound Energy (PSE), and Federal Communications Commission (FCC) standards. This company is looking to innovate, meaning that they want to offer compact and versatile solutions to meet the needs of their customers. Providing great customer service and technical support, engineers gain a lot of support in their questions regarding integrating their components.

3.10.3.1 CUI Inc. SWI5-5-N-P5

The SWI5-N-P5 is the only power supply that is of a different configuration. The other three are that of an enclosed metal mesh casing but this is an efficient AC-DC wall plug power supply. With an output voltage, current, and power of 5V DC, 1A, and 5W respectively, the SWI5-5-N-P5 is a suitable option for a wide range of applications. These applications could include consumer electronics, industrial equipment, communication devices, as well as security systems, which our product is described as. As a wall plug-in adaptor, we would have to take into mind the distance that we can reach from the point of the outlet to where our system would sit since the cord that comes with the unit is about 1.5 meters in length. The SWI5 has an operating temperature range of 0°C to 40°C, making it the one unit to have the smallest range to be compared. While having a smaller temperature range, this unit also has the smallest MTBF, which is 50K hours as per Telcordia SR-332 [197].

Along with this unit having a range of applications, it makes sense that it would have a range of protections as well as safety approvals. With Class II construction, the SWI5 provides double insulation which enhances safety, while having protections against over-voltage, over-current, and short circuits, further ensuring the longevity of the delicate system. The SWI5 also has UL, PSE, and FCC safety approvals, reinforcing the reliability of the product. On top of that, The SWI5 meets the Department of Energy (DoE) Level VI efficiency standards, meaning that as an AC-DC external power supply with an output between 1W and 49W, we have a minimum average efficiency calculated by:

$$0.071 \times \ln(P_{out}) - 0.0014 \times P_{out} + 0.67$$

While having a maximum power in no-load mode of less than or equal to 0.100, being at less than 0.075W.

3.10.4 Product Comparison

Based on all four of these different PSUs, there are very clear differences while also having a good number of similarities. Looking at the output power of each PSU, we can see that the Mean Well LRS-50-5 gives off the most wattage at 50W, while also has the longest minimum MTBF at 645K hours with an impressive 90% efficiency rate. However, due to these key values, the cost of this unit averages from \$15-20 which makes it the most expensive option. As university students without a sponsor, the total cost of the project must be taken into consideration as we are paying out of pocket. Based on all the comparisons, we chose to use the Mean Well LRS-50-5, which has been highlighted in the table below.

Table 3.15 Power Supply Comparison

Parameters	Power Supply Units			
Brand	Mean Well	RS Pro	CUI Inc.	Mean Well
MFG. P/N	LRS-50-5	777-6171	SWI5-5-N-P5	RS-15-5
Output Voltage	5V DC	5V DC	5V DC	5V DC
Output Current	10A	7A	1A	3A
Output Power	50W	35W	5W	15W
Efficiency	Up to 90%	Up to 82%	Up to 85%	~77%
Operating Temperature	-30°C to +70°C	0°C to +45°C	0°C to 40°C	-20°C to +70°C
Form Factor	Enclosed	Enclosed	Wall Plug Mount Adaptor	Enclosed
Size	99x82x30mm	99x82x30mm	22.6x36x47.7mm 1.5m Cord	62.5x51x28mm
MTBF Min. (Hrs)	645K	N/A	50K	108.8K
Price	~\$15-\$20	~\$10-15	~\$10	~\$10-15

3.11 Voltage Regulation

This section covers discussion regarding the voltage regulators being used in this project as well as the issues that have been encountered with specific products. Based on the advice of our advisors and the issues surrounding a lot of the chosen regulators from other groups

in Senior Design 2, the design and choices made regarding voltage regulators needed to be taken into careful consideration to ensure the safety of the system.

Our advisors suggested that there be multiple guidelines set in place that would help our PCB to function safely. These include considerations being made toward the selection of regulators, the layout of the PCB, the selection of capacitors, the size of the components, the design of the ground plane, and thermal management.

Based on our advisor's recommendation, we can utilize regulators with switching frequencies below 500kHz to 1MHz as a priority since these tend to be older models that are understood better and are less prone to introducing less noise into the system. We can reduce the risk of incompatibility with the other components in the system. Along with the selection of the regulator, we also must make sure that we choose the supporting capacitors as well as feedback resistors carefully to further reduce the noise and ripple of the output. The size of these components also matters, as the bigger 0805 capacitors and resistors are undesirable. The 0805 capacitors introduce too much inductance and create a big restriction in the placement of the components, since we ideally want to place the capacitors as close to the regulator pins as possible. The same can be said for the feedback resistors as the 0805 resistors are larger and the same problem arises. However, in the case of the feedback resistors, we would need to place them as adjacent as possible to the IC.

Thermal management is an aspect of the design that is extremely critical as a lot of other groups had a lot of issues regarded their components burning out due to improper placement of their regulator systems. Voltage regulators in general generate a significant amount of heat, therefore detailed thermal calculations need to be done to ensure that the regulators remain in the operating temperature limits that are detailed within the respective datasheets. We also must include heat sinks and thermal vias in the design of the PCB to manage the temperature of the system.

iSPY relies stable and efficient power regulation for the operation of the FSO communication system so voltage regulators play a crucial role in maintaining a consistent power level for each of the components. Since we have two modules that require power, the Raspberry Pi 4 Model B and the second PCB that is responsible for data decompression. The voltage regulation is mainly implemented in the design of the second PCB since a lot of the chosen components need 3.3V input to operate safely. Since we have a 5V power supply, we would need to step down that input power to properly align to the requirements of the installed components.

3.12 Differential Impedance

Differential impedance is a crucial concept in PCB design, especially in the case of iSPY, a project that relies on high-speed data transmission standards such as ethernet and HDMI. Differential impedance refers to the impedance encountered by a pair of conductors (such as PCB trace) [198]. It is a fundamental property of differential signals which are interpreted by a receiver component. One way to think of a differential signal is this: it is a propagating electromagnetic disturbance that involves two different signals, ideally being

sent together along a pair of traces [199]. These pairs consist of two closely coupled traces that carry the opposite polarity, which help minimize the electromagnetic interference, which is ideal in high-speed digital circuits like the ones in our product. We want to minimize signal degradation within these traces and that is affected by variables like trace width, length, the spacing between the conductors, as well as material properties.

Calculating and controlling the differential impedance within these pairs ensure that the signals can propagate through the circuits within the system without degrading due to interference and noise. Maintaining the correct differential impedance is vital for preserving signal integrity. Mismatched impedance can lead to signal reflections, crosstalk, and electromagnetic interference (EMI), which degrade the quality of the transmitted signal [199]. Signal reflections occur when an impedance mismatch causes signals to bounce back to the original source of that signal, which leads to distortions in the signal and a lowered quality of the overall quality. Electromagnetic interference is when poor impedance matching causes an increase in radiated emissions, resulting in interference in other systems and a failure to meet regulatory standards. Crosstalk occurs when the traces are too close together and it causes noise within the adjacent signals.

If we were to look at what our project would require, since it includes HDMI, Ethernet, and USB connections, we would have to look at what the differential impedance each differential pair would need to operate efficiently. For example, for ethernet and HDMI connections for data transfer, we would need to keep the differential impedance at 100 Ω with a tolerance of $\pm 10\%$ [200], [201]. To calculate the differential impedance, we would also need to calculate another important quantity, that being the odd-mode impedance, which is the impedance of the single trace. The equations that describe the relationship are listed below:

Odd-mode impedance:

$$Z_{odd} = \sqrt{\frac{L - L_M}{C + 2C_M}}$$

Differential Impedance:

$$Z_{diff} = 2Z_{odd}$$

There aren't many analytical models for calculating these two quantities as the process is quite complicated. However, to achieve the desired differential impedance for the specific data, designers can use built-in calculators like the ones integrated into PCB designing software like KiCad, which is what we will be using to design our system. We can utilize the tools and calculators provided to fine-tune parameters that will contain our differential impedance within the recommended amount. With these parameters, we can adjust the trace spacing and width while also accounting for the properties of the materials that are selected in the process of the design. But of course, multiple simulations and iterations must be performed to properly make adjustments that will help our design.

Chapter 4 – Related Standards and Design Constraints

In every design there are safety measures and design specifications that need to be met by standards as well as constraints. In this section we go over the necessary standards that our project would have to comply with in safety, security and industry. We discuss the governmental or national specification standards of electrical components and optical components. The design constraints will be detailed out for our project relative to general constraints such as time, economic influence and other constraints.

4.1 Industrial Standards

Industrial standards are parameters and guidelines specified by governments, companies, or non-profit scientific or engineering organizations to create products and/or systems. To produce a safe, universally understood, and high-quality product, this section discusses the industrial standards relevant for the success of the project.

4.1.1 Related Electrical Standards

Electrical standards include all standards relevant to electrical systems. This section discusses standards with PCB design and electrical power standards.

4.1.1.1 PCB Design Standards

IPC-2221 establishes standards for PCB design such as schematic, material selection, thermal management, DFM, DFA, DFT, and quality assurance [202]. The IPC-2221 standard provides comprehensive guidelines and recommendations for PCB design, which covers numerous aspects that must be adhered to ensure that reliability, manufacturability, and performance are emphasized.

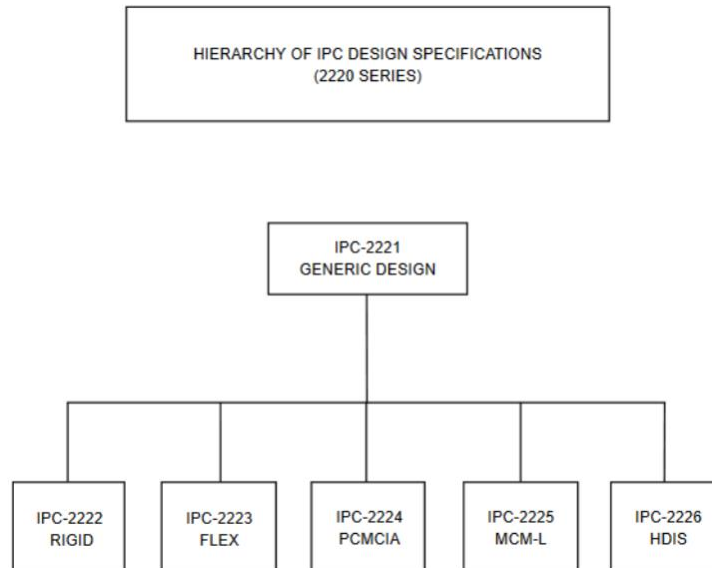


Figure 4.1. *Hierarchy of IPC Design Standards*

The key areas that IPC-2221 addresses are:

Design Guidelines: Guidelines are offered to provide guidance on the parameters that are critical to PCB design. These parameters include trace width, spacing, and layering. They are essential to maintain proper power distribution as well as maintaining the integrity of the signal being sent.

Material Selection: Recommendations are given for materials to be chosen based on specific properties that should be kept in mind when designing PCBs. These properties include those of dielectric strength, mechanical durability, as well as thermal conductivity. To meet performance specifications, it is important to carefully choose the right materials.

Component Placement: The placement of the components chosen on the PCB is incredibly important and IPC-2221 outlines practices that can be followed to enhance the functionality and reliability of the PCB. Placement can minimize signal interference or even improve thermal management.

Trace Routing: The guidelines provided for trace routing emphasize the best practices to minimize signal interference with high-speed signal routing. These practices detail impedance control, signal integrity, and electromagnetic compatibility (EMC). Having efficient routing makes the system more efficient and stable.

Thermal Considerations: Using strategies such as the use of thermal vias and the placement of heat-generating components are addressed by the IPC-2221 to handle thermal management. Keeping effective thermal management in mind will prevent overheating and improve the longevity of the device as well as the components inside.

Quality and Reliability: To ensure that the designed PCBs meet quality standards, criteria are included in IPC-2221. Reliability and compliance with industry regulations are ensured when the criteria are followed.

Design for Manufacturability (DFM): With manufacturability kept in importance, IPC-2221 reduces issues and streamlines production with the DFM guidelines that are included. These guidelines cover aspects like ease of fabrication, assembly, and testing.

Overall, using IPC-2221 as an encompassing guide for designing our PCB, we can create a reliable system that can adhere to industry standards. Utilizing these standards will ensure that our designed PCB will be manufacturable and function properly at a minimum cost.

4.1.1.2 IEC 62368-1 and UL 62368-1

IEC 62368-1 and UL 62368-1 are standards that apply to audio/video, information, and communication equipment that cover safety requirements to protect against electrical, thermal, mechanical, and radiation hazards. UL 62368-1 is the North American equivalent to IEC 62368-1 while ensuring that products meet both international and North American safety requirements. These standards focus on ensuring that products follow the requirements that provide safeguards against likely and consequential faults, foreseeable misuse and external influences such as temperature, altitude, pollution, moisture, and overvoltage [203]. Products and manufacturers, like Mean Well, must undergo testing and certification by UL to ensure compliance while also requiring proper documentation to inform users of safety precautions and compliance.

4.1.1.3 DoE Level VI

The U.S. Department of Energy (DoE) established a set of regulations labeled as the DoE Level VI efficiency standard to improve energy efficiency within external power supplies. Based on the graph provided below, the requirements are listed.

Table 4.1 DoE Level VI Requirements [204]

AC-DC External. Power Supplies (except low voltage and multiple voltage output)		
Nameplate Output Power (P_{out})	Minimum Average Efficiency in Active Mode (expressed as a decimal)	Maximum Power in No-Load Mode (W)
$P_{out} \leq 1W$	$\geq 0.5 \times P_{out} + 0.16$	≤ 0.100
$1W < P_{out} \leq 49W$	$\geq 0.071 \times \ln(P_{out}) - 0.0014 \times P_{out} + 0.67$	≤ 0.100
$49W < P_{out} \leq 250W$	≥ 0.880	≤ 0.210
$P_{out} > 250W$	≥ 0.875	≤ 0.500

The implementation of DoE Level VI is expected to increase energy savings significantly while contributing to lower electricity bills for customers. Although this standard is mainly

for external power supplies, it is still beneficial to explain and understand the benefits and requirements of implementing an efficient power supply to power our system.

4.1.1.4 IEC 61000-4-5 and Other IEC 61000 Standards

For manufacturers to produce safe and reliable electrical and electronic equipment, the products would have to be tested to comply with standards given out like IEC 61000-4-5, which describes the immunity requirements, test methods, and range of recommended test levels for equipment subjected to surges caused by over-voltages. While putting equipment under testing, the standard aims to make sure that this equipment can withstand without suffering from malfunctions. The standard has several performance criteria to pass, with those being operating as intended during and after the test, recovering automatically after temporary loss of function, or requiring manual intervention to restore operation after temporary loss of function.

It is important for companies to comply with the standards presented as they enhance the performance and reliability of the product they are trying to push. Having safe products and those that work as intended gives the company the reputations that they gain. Without compliance, the purchased products are deemed unsafe to sell and would risk the systems that they are integrated into. As mentioned before, one of the companies that our powersupply will come from, Mean Well, follows IEC 61000-4-5 as well as other IEC 61000 standards [205]. What this means is that they comply with the standard as well as test their products so that they manufacture reliable and safe products.

4.1.2 Related Optics Standards

Optical standards are an important part of the project. These standards range from optical design standards to telecommunication standards. The following section only outlines the optical standards deemed relevant to this project.

4.1.2.1 ISO 9001, ISO 10110, and ISO 9345

The lenses used in this project were purchased by Edmund Optics and Thorlabs. These two manufacturers follow strict regulations to ensure that their products are on par with the standards.

Edmund Optics is ISO 9001 certified in several of its facilities, including the company's headquarters. Their products "undergo rigorous and thorough testing as part of the company's quality program and in compliance with Edmund Optics' global quality procedure" [206]. ISO 9001 is the most used standard for quality management globally [207]. [ISO 9001:2015] Edmund Optics has also implemented ISO 14001 standards in their company to improve resource efficiency and reduce waste [206].

Edmund Optics is also ISO 10110 certified. ISO 10110 is the standard for optical drawings. "ISO 10110, Optics and optical Instruments – Preparation of optical drawings for optical

elements and systems, is the primary reference for preparation of drawings for optical elements and systems” [208]. This standard is divided into 13 parts, and they each highlight details that should be taken into consideration when drawing and describing the drawings for optical elements. All of the optical components fabricated by the company have a drawing in which we can see its schematic and description (Appendix D). [208]

Thorlabs is ISO 9001, ISO 10110, and ISO 9345 certified. ISO 9345 is the standard for optics and photonics instruments, it refers to the imaging distances related to the mechanical reference planes [208], [209].

4.1.2.2 IEEE 802.3 Clause 38 Ethernet Standard

This standard governs the physical medium dependent sublayer and baseband medium of type 1000BASE-LX and 1000BASE-SX. In other words, it describes an ethernet standards at the physical network level for computer communication facilitated by fiber optic cables and transceivers [210]. This standard was created for environments that necessitate high-speed data transmission via optical fiber cables by means of transceiver.

The clause stipulates 1000BASE-LX is to refer only to long-wavelength, single-mode fiber for long-haul fiber communication. In this case, long wavelength refers to either 1310 nm or 1550 nm laser operation wavelengths. It also stipulates 1000BASESX is to refer only to short-wavelength, multimode fiber for short-haul networks of a couple to hundreds of meters. The clause describes 1000BASE-LX and 1000BASE-SX as being physical medium dependent which refers to requiring the use optical fiber cables and a transceiver to achieve communication [210].

The clause also specifies how data is encoded for fiber optic communication. The encoding scheme to be used is 8B/10B encoding. Additionally, the encoder and decoder built into each transceiver is to be able to detect errors and maintain a balanced signal [210].

4.1.2.3 Optical Transceiver Multisource Agreement

Optical transceiver modules across companies do not share the same form factor and basic design by mistake – these companies share an MSA. A multi-source agreement, also known as an MSA, is a manufacturing agreement that binds manufacturers to make a product that has the same functionality across different vendors [211]. This agreement was created to address the form, fit, and function, like electrical and optical interfaces, of optical transceiver modules before IEEE created its standards [211].

At a high level, MSA standards make all MSA-compliant transceivers the same while keeping costs low – third-party transceivers, like HiFiber, will offer the same functionality as a name-brand transceiver, like Cisco, for the same price [211]. With an MSA, customers have more options at their disposal without compromising performance or costs. Additionally, this effort enables customers to efficiently exchange transceivers across

switches and routers [211]. All optical transceivers listed in section 3.3.3.3.1 are MSA compliant.

4.1.2.4 IEEE 802.15.7 and 802.15.13

IEEE standards 802.15 are standards related to wireless specialty networks. The specialty networks described by this standards body include wireless personal area networks like Bluetooth, wireless local area networks, body area networks like body sensors connected to an external device (think blood sugar sensors), visible light communication, optical wireless communication (introduced in 2017). The standards associated with visible light communication and optical wireless communication are of importance for the success of this project.

IEEE 802.15.7 are standards governing free space optical communication using visible light called visible light communication. The rise in LiFi (light fidelity) technology ushered in a push in the creation of this standards body. Optical devices like LEDs may be used for LiFi and laser for more long-range scenarios.

IEEE 802.15.13 are standards aimed at supporting stationary and mobile point-to-point optical communication links with an approach like radio communication standards. The standards detail system requirements for short-range optical wireless communication links for industry, home, and office scenarios. These standards highlight the inherent security associated with optical wireless communication systems by detailing communications as being confined by the optical transmission path. However, encryption techniques are described as a method of increasing the security of optical wireless networks. Additionally, the standard tackles the subject of optical modulation methods for industrial purposes. Compared to IEEE 802.15.7, these standards are catered towards optical wireless communication capable of faster data rates compared to those in visible light communication.

4.1.3 AI Standards

AI Standards are very important when it comes to our project. They are focused on the safety and security for individuals. When it comes to the use of AI functionality it is key especially when used globally. Although that is not the only thing the AI Standards are for when it comes to government and business, AI must be relevant to the work or job depending on the circumstances. Threats from the use of AI must be considered and maintenance of systems that require AI. The work that AI does in designing scripts, computer vision techniques and web searching is a huge area of concern. The standard mentions the transparency of data characteristics and the system that is using AI. AI can also cause risks to happen in process or design which there is a focus on risk management for AI as well. Then there are the smaller things about AI use such as terminology, measurements and metrics in development. AI Standards are for prevention as well. Human interaction with AI can cause harm and invasion of privacy. AI Standards cover privacy, security, safety and prevention of harm towards humans.

4.1.4 Surveillance Systems Standards

For surveillance systems the suggested standards are ISO 9001, ISO 27001 and, ISO 22301. The first one was discussed in the previous section, it is a standard about quality management. ISO 27001 refers to information security, and ISO 22301 is about business continuity [212].

ISO 27001 includes requirements for information backup, information deletion, access to source code, and clock synchronization to name a few [213], [214]. According to HighTable, a company that specializes in consulting and implementation of this standard, ISO 27001 was designed to be used “for implementing information security controls based on internationally recognized best practices; for developing organization specific information security management guidelines.”

ISO 22301 provides a “framework for organizations to plan, establish, implement, operate, monitor, review, maintain, and continually improve a documented management system to protect against, reduce the likelihood of, and ensure recovery from disruptive incidents.” [215].

Another standard for surveillance systems is IEC 62676-4:2014. This standard gives recommendations and sets requirements for all the stages, from product selection to installation and testing video surveillance (VSS) [216].

4.1.5 Camera Module Standards

The camera module for iSPY meets the FCC 47 CFR Part 15, Subpart B, Class B Digital Device Electromagnetic Compatibility Directive (EMC) 2014/30/EU, Restriction of Hazardous Substances (RoHS) Directive 2011/65/EU Compliance. Which means, it has been tested and passed the necessary standards and regulations, in this case, the product is not affected and/or does not generate electromagnetic disturbance. And during its fabrication the use of hazardous substances was limited.

4.1.6 Privacy Regarding Surveillance Systems

Surveillance systems that include audio transmission and recording have many benefits as they can alert the user if someone is entering their property or wake up the parents if their baby starts to cry.

However, the recording and storing of audio is a legal issue that deals with the privacy of the individuals being recorded and we had to consider when planning this for project.

The federal Wiretap Act prohibits individuals from secretly recording conversations, and on top of that Florida is one of the states that requires two-party consent for a conversation

to be recorded. In other words, for iSPY to transmit, store and record audio we would need the consent of everyone in the room [217], [218]. Therefore, we decided to not add audio recording to our project. Even if installed in a public place any recording done could not be used in court and/or as part of an investigation. [217], [218].

4.2 Practical Constraints

Constraints can take the shape of any obstacle that a team faces during a project and are an important aspect of the design process that need to be taken into consideration. This is to ensure that the project meets parameters set by various variables that the project is encompassed by. In our case, as student engineers in senior design, we are faced with a multitude of constraints that can set us back if not managed.

4.2.1 Time

For this project, the primary constraint is time. Since this is a senior design project, we have very strict regulations and deadlines that must be met. This only gives our group a limited amount of time to conceptualize, research, design, and fabricate our system and idea. Due to this restraint, we have had to come up with different solutions and follow different paths to focus on meeting our deadlines rather than fully fleshing out a fully optimized final product.

One of the key challenges we faced is that we only had a single academic calendar year to construct and demonstrate a functional system. We are given no extensions and have firm deadlines that must be met to successfully complete the course and pass. Fortunately, we were able to gain a bit of a head start as half of our group was formed in the previous summer semester and started workshopping the idea that we wanted to do. This allowed us to mitigate some pressure as we progressed because we had the general idea, and we can jump straight into research and development immediately.

On top of designing the system, we also must have ample time set aside to conduct extensive testing and debugging, all within the timeframe we were given. In any sort of design project, we assumed that we will run into setbacks and errors, so we allocated a sufficient timeframe that can encapsulate this threat if need be. Unforeseen issues that weren't accounted for are a hindrance to the success of the project plan and the passing of the course.

Another related issue would be the delay that we could see when ordering the parts for the project, as a numerous number of variables can affect whether the components, we ordered arrive in time to complete the project. Projects like this, with such a limited time span, often face delays at this stage because there could be potential supply shortages or issues with shipping that could deter the design's completion. Without the components for the project, we cannot move forward with the critical portion of constructing, testing, and debugging the system, eating up valuable time.

Having such strict time constraints might make us want to take shortcuts to achieve our final goal. However, having that mindset will only lead to a rushed and sloppy project that compromises the quality of the overall function. Adjustments are bound to be made to help solve issues but if those adjustments aren't drastic at such a late time, we can avoid jeopardizing the structure of the project.

One of the ways the deadline has constrained our project is seen in the optical design. When designing a wide-field-of-view system a lot of optical designers can count on parts being custom-made for their prototype to work. However, after talking to some people in the industry who actively need custom-made lenses, we were told that the waiting time is at least three months. Unfortunately, that would not be enough time for us to receive and test our system. Therefore, the system will have to adapt lenses/surfaces that are already available in the market.

Our team has a strong value that we must hold everyone accountable and withhold our responsibilities that we are given or take on. We followed the principle that was explained to us in our class where we "Do What You Say You Will Do" or DWYSYWD. Having a set goal and working towards that goal consistently and efficiently leads to a better product. Communication and proper planning helped mitigate these potential issues that could pop up and allow us to complete our project on time and to a quality standard.

4.2.2 Economic

One of the main constraints that must be taken note of was the economic constraint. Our project has no sponsors, so the whole cost came out of pocket. Due to this, we took extra care in constructing a budget so that we had enough funds to cover the entirety of the project or not spend more than we needed to. We had to carefully investigate different components and comparisons so that we can find the products that have the capability to do what we want to accomplish.

We hope to keep the entire cost below \$700 while still having extra funds if something breaks or burns out during testing. By having a buffer of funds, we can ensure that in an unfortunate event, we allow ourselves to have less worries when working with such delicate components. Although we can replace anything that breaks, we'd like to avoid that possibility since it will only rack up a higher cost in the long run of the project.

Our research in part selection is essential when carrying out the project, as we don't want to spend an absurd amount of money on a component that has no overall functionality for the purpose of the project. Therefore, we must consider the price of components and the functionality in our research, making sure that we can use the component, or it would be a waste of time and money, both of which are very limited as Senior Design students.

The economic constraints that we place upon ourselves can be a detriment to the success of the project because of the need to buy the best products on the market that can do what we would like to achieve while also not being absurdly expensive. This requires us to do exceptional research so that we can compare different products and make sure that the cost

ratio is sufficient. We can also avoid going over the budget if we give ourselves a cost buffer. This cost buffer, as explained previously, will give the team a safety net when we're in the delicate fabrication process. If something were to break, we would already have a time issue since we would need to wait for a component to come in, but instead of being over budget and over time, we can allow some leeway in terms of cost.

As long as we keep researching and making sufficient comparisons, we can choose the best products that can handle the scope of the project while also maintaining our budget and keeping that safety buffer intact.

4.2.3 Environmental

The environmental constraint is one of the biggest design constraints that we had to factor in for this project since we rely heavily on our FSO system. Within FSO, we must work with open air acting as our conductive material for light. This could lead to numerous variables that could potentially affect the functionality of the system since these types of systems are very sensitive to a range of factors.

One of the biggest aspects that can affect our system is light interference. Since a portion of our system is in the open air, sunlight and ambient indoor light can distort and overwhelm the system and the transmitted bitstream, resulting in noisy data and output. Just like light, we can also be wary of electromagnetic interference from any other electronic devices in the vicinity of our system.

A significant challenge that our FSO system can face would be interference from the atmosphere around the system, specifically any air impurities that may be present. For example, in an indoor environment, dust or airborne particulates can be a source of interference as they can diminish the integrity of the signal. Or, in an outdoor environment, foggy and rainy areas can absorb and disrupt the light being transmitted. If our signal gets disrupted or absorbed by other variables then we wouldn't get the right data back, if at all, compromising the range that the signal can be received.

Of course, since we are working with electronics, we would have to think about how temperature and humidity can affect our components. Living in Florida, we get hot temperatures as well as humid or moist weather. Humidity is one of the bigger threats as our components can be short circuited or corroded from the condensation that is gathered. And with temperature, we would have to think about the overheating that can occur from the operation of our system on top of the environment's temperature.

Furthermore, having optical paths makes the system complicated since we must be precise and accurate in our construction. Therefore, any external movement such as vibrations and mechanical shocks can disrupt the system and null out the output we expect to gain. A vibration could knock our pathing out of alignment and disrupt the system so having a robust housing along with secure installation can mitigate the risk that this variable imposes on our system.

For any system relying on sensitive components, it is essential to consider the effects of these constraints so that we can make the proper choices to construct a strong design. We would have to design a robust housing structure for the system that can withstand environmental variabilities just in case any of these constraints prove to be an issue. Maintenance and system redundancy can also improve the longevity of the system.

4.2.4 Remaining Constraints

Constraints are encountered throughout all engineering projects. Some constraints can be accounted for while others cannot. In this section, constraints like sustainability, manufacturability, and ethics are discussed.

4.2.4.1 Sustainability

Since this system has the sole purpose of security, sustainability is a major constraint since we want a system that will last a long time without needing much to maintain it. Having a system that can last a good amount of time energy-wise, while also being able to withstand some weather conditions if used outside, is what a good security system can do.

We would have to make the system with the idea that natural weather or indoor conditions vary, so having a strong casing will have to be a priority in the overall construction of the system. On top of that, having a strong power supply and efficient power management can have a huge benefit to the overall functionality of the project as it will be able to utilize power for a long time without components burning out or not having enough power.

4.2.4.2 Manufacturability

Another constraint that must remain considered for this project is that it needs to be manufacturable, which is important for a group of undergraduate students working on this project. We are limited to resources provided by the university; therefore, we are limited in the scope of the project. This results in our group having to keep the design of our project simple enough so that we can utilize resources provided to us.

However, as we are not required to create things from scratch, we can purchase our components from online resources and modify anything as needed, which reduces the need to go overboard with extensive designing. By allowing us the ability to outsource aftermarket parts, we assembled our system using techniques that we had gathered, thereby minimizing the workload of constructing a functioning system.

Since our project deals with optics, we opted to purchase some pre-made parts that will lessen the burden the fabrication of our system will have on the group. Of course, we designed a lot of how the components come together, but we wouldn't be able to construct the major components from scratch.

4.2.4.3 Ethical

As a group, we have decided that we would like to factor in an ethical constraint into our project, ensuring that we choose companies that are ethical in nature. We would like to avoid purchasing parts and components from companies that outsource from countries that use any type of slavery or labor of an unethical manner. Third-world countries are a prime target for an exploited workforce. Being aware that trade-off is deemed important to our group as we all believe in ethical practices in the engineering industry.

Since our group has a strong belief in the ethical acquirement of our parts, we want to make sure that this constraint is factored properly in our planning. Therefore, we want to only use parts from companies that are deemed trusted based on track records and extra research. To the best of the group's knowledge, we avoided dealing with companies that use unfair labor practices and only source parts from fair businesses that hold a high standard. Limiting unfair business practices will be ensured with the extra research our group will conduct into the choices we make for this product to take shape.

Chapter 5 – Large-Language Model Engine

In the recent past, large-language models have enabled tools for engineers across disciplines. As this tool continues to proliferate throughout the industry, it's important to consider it for the success of the project. In this section, large-language model outputs are discussed in detail with regards to details of the project.

5.1 Case Study 1: Asking ChatGPT and Copilot for Camera Module Specifications

Large Language Models (LLM) are a great tool when used correctly. For this project, it was used as a search engine, like the way we use Google, it was also used to help with brainstorming and reviewing our grammar.

In prompt 1, we asked ChatGPT what specifications/parameters we should consider when searching for a camera module. The outcome was helpful, it gave us a list of details to look for and it explained the role of each parameter and why it mattered. We did not mention that we were on a budget, but the LLM went ahead and told us to watch out for the financial aspect of it.

The same prompt was given to Microsoft's Copilot and the outcome was similar. Copilot went a step further and gave us references (links) for each parameter it mentioned, and it also highlighted the importance of the sensor's availability.

For this case study the results obtained through LLM were helpful, they summarized a big portion of the key aspects we should research before purchasing our camera module, and one of them even gave us websites to check the information.

5.2 Case Study 2: ChatGPT vs. Zemax

The lens design software Zemax is very a unique tool, and it is mostly used by optical engineers. The software is not that popular outside of the field of optics and photonics. Therefore, there is a limited amount of information that can be found making it harder for LLMs like ChatGPT to have accurate data on it. ChatGPT does not know how to design a lens system, the outcome 2 shows that it has ideas on how things should be but when its inputs are brought to Zemax the design does not work.

Also, whenever ChatGPT tells the user to do something in Zemax the instructions are not clear. "Optimize your design iteratively, using the Merit Function Editor in Zemax, focusing on minimizing chromatic aberration, spherical aberration, and distortion." Such direction is not so straightforward as the LLM implies. To use the Merit Function Editor the user needs to set up boundaries, add specifications, check parameters and make sure they fall under the design that is being implemented.

When dealing with peculiar softwares, using LLMs is not beneficial and in fact I believe it only hurts the learning and designing process. Since it does not have accurate data but only a faint plan/idea on how it should be, following their steps leads to issues.

5.3 Case Study 3: Asking Multiple AI Models for Power Supply Recommendations

Using LLM as a secondary search engine is useful for quick research while providing additional resources to the prompts that are inputted. With the help of multiple AI resources, namely ChatGPT, Copilot, and Google AI, I was able to amass a good list of different PSUs that could be integrated into our design to supply power. Some are more beneficial than others as one could provide more information in one pass, whereas another could only provide one general response. As seen in ChatGPT's response, it gave me very broad types of PSUs that could be used rather than actual products currently on the market. It could be asked for more information but that would require a more in-depth inquiry. Copilot and Google AI both provide different products that are currently in the market and available to purchase, with the exception of some products being out of stock. Not only did they provide different products from different brands, but also provided links and references to investigate each product more individually.

One thing that ChatGPT and Copilot have in common, whilst differing from Google AI, is that we can add files and pictures into our prompt to give further information to our question. This allows the model to learn more about the topics and provide a response that is more in line with what we are looking for. Google AI doesn't have that feature since we are searching directly on Google, but it was still able to provide suitable answers.

5.4 Case Study 4: Camera Module Options Using ChatGPT

We also used ChatGPT when looking for camera modules [Appendix F – prompt 4]. Since the LLM can search its database quickly and give us options it was a good way to narrow down our choices, it was a lot easier to ask ChatGPT what the options were then to visit several websites and manually check every single camera module. The LLM also gave us a range of values for each module.

The following question was if it knew what the shutter type of one of the modules was and if there was a way we could replace the lenses from the modules to add the lens design we wish to have. The first question was answered with some uncertainty. It did give the sensor type, however it also explained both types of sensors (rolling and global) as if it was justifying its answer, and then at the end it told us to check the specification tables of the module.

Even though it narrowed our options down ChatGPT could not offer certainty about the specifications, the LLM still has limitations, and it is upon every user to go and check the information.

ChatGPT told us that yes, we can replace the lens for one of the modules and even gave us steps in how to do so. The steps were not so specific, but it came up with it without us even asking for instructions. The LLM also warned us of the issues we might face when removing the original lens and placing something else on its spot.

5.5 Takeaways

When used as a search engine, a tool to brainstorm ideas, a place to obtain an overall picture before getting into the details, LLMs are extremely beneficial. Since websites and applications like ChatGPT and Copilot speed up the proofreading process and answer our questions almost instantly and with great accuracy, they enabled us to direct our energy into understanding how all our areas came together in this project.

We have an idea, each one of us understands their part and how we got there (the mathematics, physics and engineering behind), we know what to look for when purchasing the parts and now is the time to figure out how to place this puzzle together.

I think this is the best part about using LLMs. Of course we must understand what is taking place, but after that they can save us a lot of time and energy. The way LLMs can harm our learning is if we completely replace books, lecturers, and research papers for their output. As good as it is in acquiring and delivering information, it does not obtain the answer for everything, especially if the subject is unique.

It's important to look at ChatGPT as a tool in your arsenal instead of it being your go to for everything related to research. ChatGPT has greatly benefited our learning experience in Senior Design because we used it for guidance instead of using it as our sole means of research. In the beginning of writing sections, it can be difficult to know where to start. Using ChatGPT to give a guideline for what to write can be incredibly helpful to jump start ideas and get rid of brain fog.

For example, the prompt was given to ChatGPT asking what someone should write about when researching codecs for live video compression to transfer data across a laser using bitstream. ChatGPT gave a comprehensive list of what should be covered and what is important to note when researching. ChatGPT's response worked well as a guide rather than a sole source of information. ChatGPT listed the types of codecs that should be covered. Instead of just relying on what ChatGPT said, research was done to get more information that can be used as quality research. Numerous articles were visited trying to research the topic to their fullest so that an accurate decision can be made on what to use.

ChatGPT was always supplemented with proper research done by researching topics from various other sources. ChatGPT can be wrong. As helpful as it can be, it is not perfect and is prone to mistakes. That's why it's important to not rely on it entirely. In that way,

ChatGPT can be harmful. If a person blindly follows what ChatGPT says without verifying it for themselves, then they are doomed to make mistakes.

Chapter 6 – System Hardware Design

In this chapter we will dive into the hardware design of the project. The hardware of iSPY is formed by the image acquisition system, the designed PCB that houses our main components, the FSO system with its optical assembly and the evaluation boards, and the PCB that displays our live video.

This chapter contains detailed information on the specifications of the systems mentioned above, along with schematics to better guide the reader.

6.1 Image Acquisition System Hardware

The image acquisition system is composed of a pinhole, two lenses and a CMOS sensor. An in-depth discussion of the simulation that took place until we reached the final design can be found in Chapter 9, Section 1.

The pinhole in the system works as the aperture stop, it limits the amount of light. And since it is the first element in the assembly, placed in front of the lenses, it also act as my entrance pupil.

The work principle is similar to a pinhole camera, since the pinhole is also the element limiting the light in our system. Figure 6.1 demonstrates how a pinhole camera works.

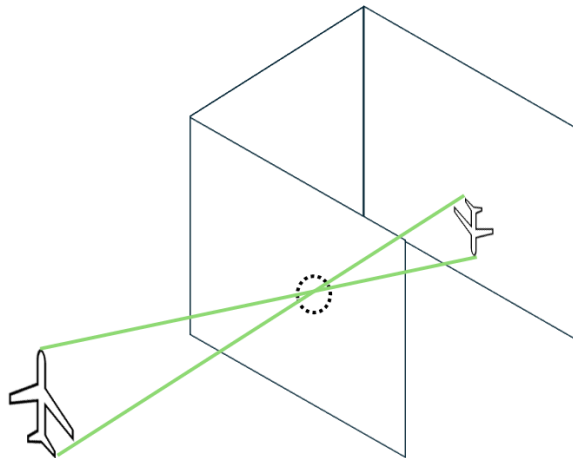


Figure 6.1. Pinhole camera by Authors.

The final entrance pupil's diameter was measured to be 0.46mm (460 μm). This entrance pupil is fairly small, and the concern became if there would be enough light into the system. To test that we used a webcam to simulate our camera module, and we placed a pinhole of 300 μm in front of it. The pinhole is a little smaller than the one we need; however, it was the only available at that moment. The results are in the following image.

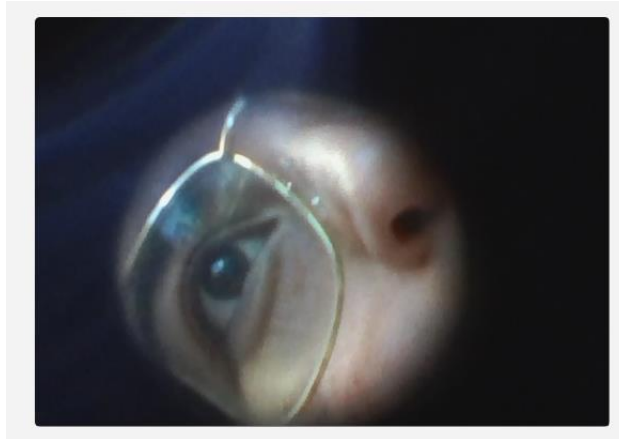


Figure 6.2. *Illumination test by Authors. Image captured using a pinhole in front of a webcam.*

Since the pinhole was placed far away from the lens due to the design of the webcam and because the webcam already has an aperture stop, the edges of the image captured are black – the tested system was imaging the solid surface of the pinhole. This is not something that happens to iSPY since the pinhole is the aperture stop, and it is placed right in front of the lens. Even then, the image captured shows us a circular frame with good lighting, which means our system has enough illumination in well-lit areas, such as an office or a lobby.

The reason why the entrance pupil is so small is due to our F-number (8.00) being large. And the only reason why we get to have such a large F-number (slow lens) is because iSPY is designed to be placed in well-lit locations. One of our stretch goals is to make iSPY work in dim places too but for that to happen we would need an aperture stop that changes size, just like an individual's pupil.

Right after the pinhole we have two lenses. The first lens is an aspheric lens from Thorlabs (354105-A) with a focal length of 5.5mm, made of D-KZ3. The exact type of coating for this lens is not available but it is anti-reflective across the 300-750nm wavelength range. This lens has a diameter of 7.2mm.

The second lens is a plano-convex lens from Edmund Optics (#45-082) with a focal length of 9mm, made of N-SF11. This lens is coated with MgF_2 , which is excellent to maximize the transmission of the wavelengths we want (visible) and block the other ones. Its diameter is of 9mm, which is important because as the lens that is the closest to the sensor it must cover the sensor entirely to prevent vignetting. The sensor size is of 6.45 x 3.63mm (7.4mm diagonal), which means that our lens' diameter is sufficient to cover it.

As you will see later in Chapter 9, the original plan did not include the use of an aspheric lens. Due to its price we tried to stay away from it. However, when trying to accomplish a lens design for a wide field of view aberrations are a huge issue. We got to a point where the quality of our image intake would decrease too much, and that is why we decided to go with an aspheric lens.

Another issue we faced when choosing the lenses was that lenses with a large diameter also have a large focal length. For this design, a short focal length is a must, therefore, we had

to match the lenses well to make sure that the effective focal length of the system would enable us to reach our field of view goal.

Equation 3.1 was used to find our desired focal length. A small change that takes place in the equation is that f (focal length) is now EFL (effective focal length) to accommodate the fact that more than one lens is used. The sensor size is 6.45 x 3.63mm with a diagonal width of 7.4mm. By replacing w with these measurements, we found the desired focal length to reach the field of view we wanted.

$$\text{FOV (in degrees)} = 2 \tan^{-1}\left(\frac{w/2}{EFL}\right)$$

Horizontal FOV

$$EFL = \frac{6.45}{2 \tan\left(\frac{80}{2}\right)} = 3.84\text{mm}$$

Vertical FOV

$$EFL = \frac{3.63}{2 \tan\left(\frac{50}{2}\right)} = 3.89\text{mm}$$

And since we had the diagonal width, we could also set a FOV in this direction and find an effective focal length to achieve it. The FOV diagonally was then set to $< 80^\circ$, just like the horizontal one.

Diagonal FOV

$$EFL = \frac{7.4}{2 \tan\left(\frac{80}{2}\right)} = 4.41\text{mm}$$

With these numbers in mind, I ran several simulations on Zemax to minimize the effective focal length and maximize performance. Thankfully, the main lenses manufacturers have Zemax files for their lenses which can be downloaded and tested in the software before purchasing them.

The best result I could accomplish was a design with a focal length of 3.65mm. Which means that our system has the field of view we proposed.

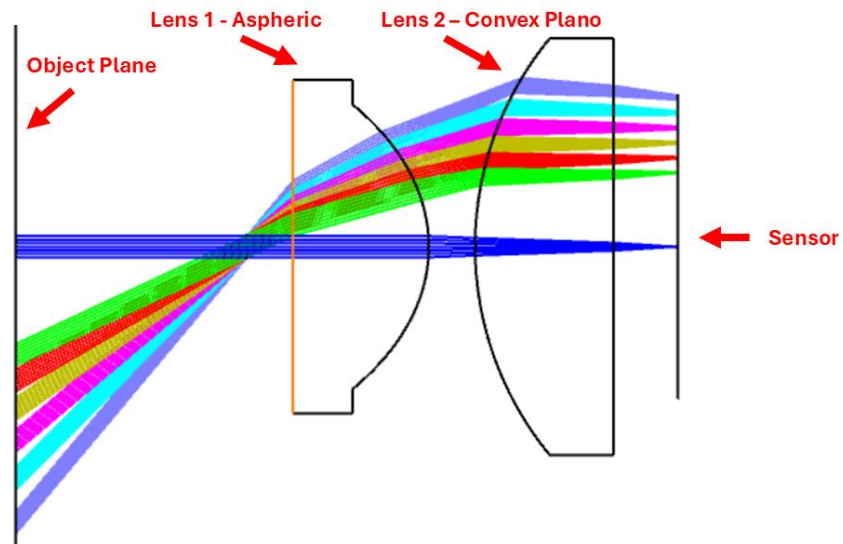


Figure 6.3. *iSPY's final lens design by Authors (Zemax screenshot of the lens system).*

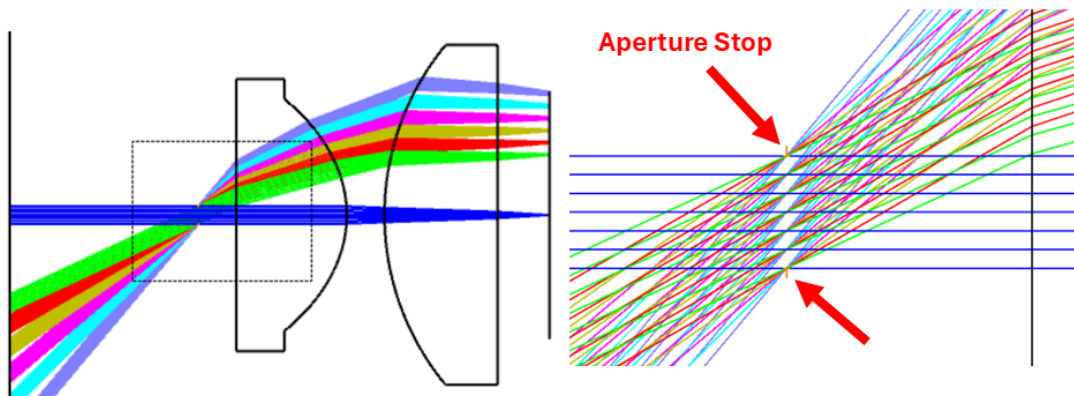


Figure 6.4. *Zoom-into the aperture stop by Authors (Zemax screenshot of the lens system).*

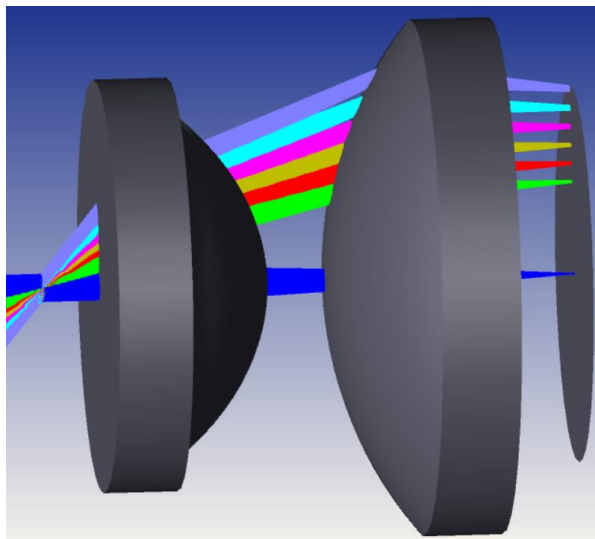


Figure 6.5. *3D model of iSPY's lens design by Authors (Zemax screenshot of the lens system).*

Figures 6.4 – 6.6 display the final lens design. The different colored rays represent fields in our systems and not the colors themselves (wavelengths). Each field is related to an angle, with the lowest being at 0° (dark blue) and the highest at 50° (purple). Zemax also only shows the fields in the positive y-direction; however, these fields in real life are also in the negative y-direction, since light comes from all directions.

In the figures, it is also possible to notice how the aperture stop is limiting the entrance of light. The figures also help with understanding how the aspheric lens differs from the plano-convex one.

The aperture stop is located 1mm in front of the first lens, the lenses are also separated by a distance of 1mm, and the sensor is placed 1.4mm from the second lens. A sketch of how we imagine the optical assembly can be seen in Figure 6.6.

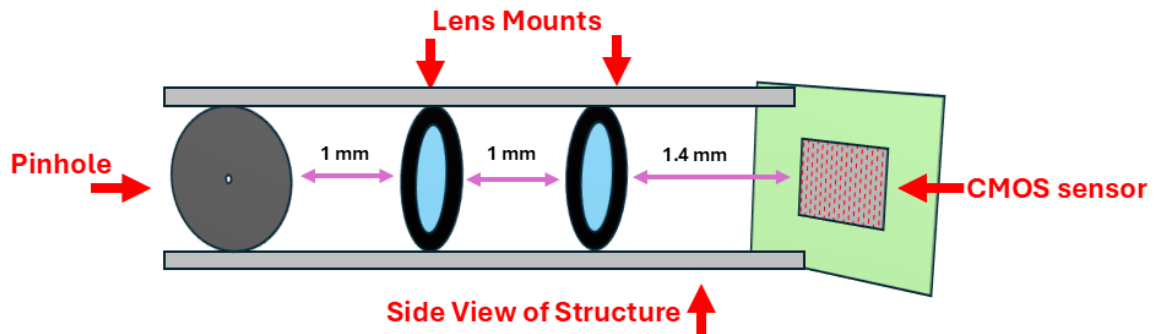


Figure 6.6. Optical assembly for the image acquisition system by Authors.

Since our lenses are small, we have 3D print mounts for it. The structure (light gray) resembles a lens tube.

Image Acquisition System, A Summary

A list from the book “Optical System Design” was taken to summarize all the information from our image acquisition system. This list highlights basic operations and performance specifications and requirements. Since the list was for general use, only the parts related to our system were kept. The completed table can be found on pages 10-11 in the book [219].

Basic system parameters

Object distance: in our system, objects that are even a couple of centimeters are seen as infinitely far away. Therefore, we consider object distance to be infinity.

Image distance: 1.382mm from the last lens, 9.32mm from the pinhole

Object to image total track: since the object is infinitely far away, the total track from the pinhole to the sensor, is approximately 9.32mm.

Focal length: 3.65mm

f /number: 8.0

Entrance pupil diameter: 0.46mm

Wavelength band: F, d, C (visible) – 0.486mm, 0.588mm, 0.656mm
Full field of view: 90° (diagonal), 82° (horizontal), and 52° (vertical)

Optical performance

Encircled energy: The encircled energy can be best explained with a graph. As displayed in the figure below, the best results are seen in the dark blue line – the 0° field, more than 80% of the energy is encircled in a radius of less than 10 μm .

There is a decrease in the percentage of the energy encircled for small radii as the fields increase. Since this is a project with time and economic constraints, such situations are expected.

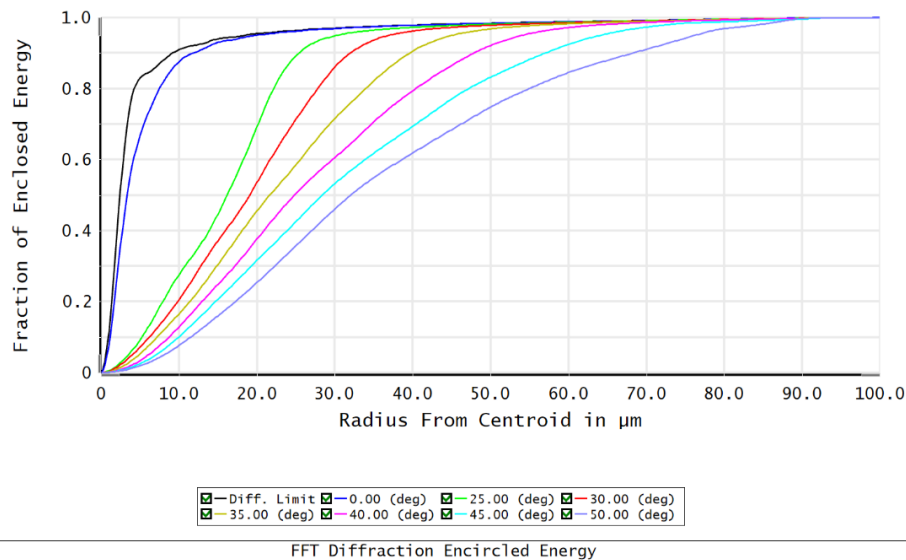


Figure 6.7. Encircled energy from the lens system (Zemax output).

MTF as a function of line pairs/mm:

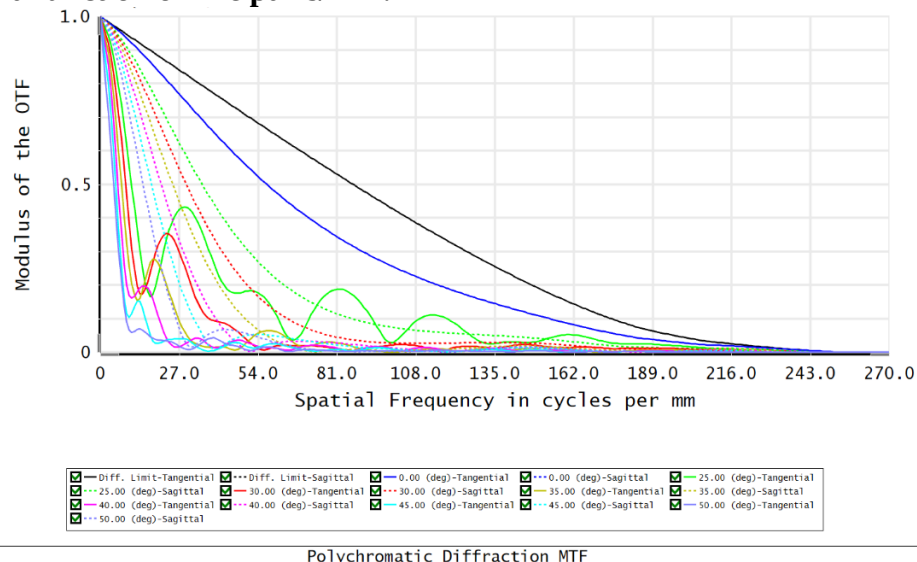


Figure 6.8. MTF from the lens system (Zemax output).

Just like encircled energy, MTF is not a single number but a function and the figure above shows how the MTF for our system looks. The black line is for a diffraction limited system, and in our 0° field we are close to it. However, as the field increases, it is more challenging to keep a good resolution.

This was the best outcome we could have had with our limited time and funds. Chapter 9 will include a section of the trials that took place before choosing this one, and MTF improvement is noticeable.

Distortion: 26%

The percentage might seem high; however, as it will be seen in chapter 9, we were able to cut down distortion by almost half. Wide-angle-lens-systems are known for distortions, so even designs that cost more and take more time to make still run into this issue.

An example of this can be seen in the Zemax software itself. The software comes with a couple of samples/examples made by optical engineers. These designs have been carefully crafted by professionals and the work to create something like that is equivalent to a Master level thesis in optical design. Due to their high quality and reliability, I studied these designs, their parameters and performances, to have a clear image of what I wanted iSPY's design to be like.

Figure 3.1 in Chapter 3 displays a lens system for a camera of 100 degrees field of view. The distortion for that system is 38.8%.

Field curvature: 0.36 mm sagittal, and 1 mm for tangential.

When compared to Zemax's design for a camera with a wide field of view, these values are within the range. The design mentioned above has a field curvature of 406 mm sagittal and 381 mm.

Lens system

Number of elements: two lenses and a pinhole (aperture stop).

Glass versus plastic: glass, D-ZK3M for one lens and N-SF11 for the other.

Aspheric surfaces: one aspheric surface.

Coatings: For the aspheric lens, the coating type information available simply states that it is "A-coated" / BBAR (broadband anti-reflecting coating) for wavelengths in the range of 350 – 700 nm. For the second lens, the type of coating is MgF₂.

Sensor

Sensor type: Color CMOS sensor, rolling shutter (color)

Full diagonal: 7.44mm

Number of pixels (horizontal): 4608 pixels

Number of pixels (vertical): 2592 pixels

Pixel pitch (horizontal): 1.4μm

Pixel pitch (vertical): 1.4μm

Packaging

Entrance and exit pupil location and size: Entrance pupil diameter is 0.46mm, and the exit pupil diameter is 4.18mm. Entrance pupil is located 1mm in front of the first lens, and the exit pupil is located -33.5mm from the object plane.

Back focal distance: 1.38mm

Maximum length: < 15mm

Schedule and Cost

Weight: < 50g

Number of systems required: one

Initial delivery date: March 2025

Target cost goal: < \$200

6.2 Optical Communications Hardware

This section will discuss the optical communications hardware that is used for the project along with the necessary calculations refuting the necessity.

The optical communications hardware for this system is chosen with careful consideration of the parameters and concepts discussed in section 3.9. These considerations include the optical power budget, collimated beam divergence, and external source radiance. The optical power budget maximum and minimum values are determined by the optical transceiver parameters of max transmission power and receiver noise floor, respectively. The collimated beam divergence is determined by the light source and the collimating optics of choice. The external source radiance depends on the free-space launch conditions.

The system's optical communication power budget is explicitly determined by the transceiver. The current transceiver – FS SFP1G-EX-55 – of choice lists the maximum transmission power as -2 dBm and the noise floor as -24 dBm. Considering these parameters, the optical power budget is -22 dBm. When considering the optical communication hardware, the optical losses per component must be considered. Even though an optical transceiver is the true source of light and detector, an optical fiber output delivers light for the FSO system and captures it for detection. For this reason, the power loss associated with each optical fiber must be considered. Equally, the losses of the collimating optics used to create the free-space link must also be considered – these losses can only be measured and are typically listed on the spec sheet.

The project will most likely make use of transceivers emitting 1550 nm light onto an SMF-28e optical fiber. These fibers have a ~9 μm diameter core with a 125 μm diameter cladding. The small core size will make it more difficult to couple light back into the fiber on the receiving end. The attenuation coefficient for these fibers is typically very low: 0.20 dB per kilometer at 1550 nm. Due to size and cost constraints, only 0.5-meter fibers will be used. The loss across each length of fiber is calculated below.

$$Loss = \alpha \cdot L = \frac{3 \text{ dB}}{1 \text{ km}} \cdot \frac{(0.5 \times 10^{-3}) \text{ km}}{1} = 0.0001 \text{ dB}$$

Each 0.5-meter fiber introduces loss equivalent to near-zero. Adding up the two fibers of the FSO link, the total fiber loss is also near zero, making the fibers a great choice for delivering the signal to free space. The atmospheric absorption of the atmosphere must also be considered for the power budget. Studies show a marginal difference in attenuation between 850 nm and 1550 nm signals over short free-space ranges [220]. Over the spec 0.5-meter range, the atmospheric attenuation is nearly equivalent to the loss of 0.5-meter fiber during a clear day [220]. Additionally, MODTRAN atmospheric simulations for 1 km ranges demonstrate atmospheric transmission for 1550 nm light to be 99.99% - almost near zero. The collimators have two points of loss that can only be measured: insertion loss and back-coupling loss. Collimators used for the project will be from the teams' advisor's laboratory and the losses will be measured when ready.

The collimated beam divergence can be calculated using the equation discussed in section 3.9.3.3. But, like the loss associated with each collimator, this cannot be measured until a collimator is handy. Regardless, depending on the collimators borrowed, the divergence is given in the spec sheet. For example, the full-angle divergence for a Thorlabs F280FC-1550 singlet fiber collimator is 0.032 degrees [221]. This tells us the beam is going to spread about 1 mm over a 0.5-meter range.



Figure 6.9. Fiber collimator lens TC25FC-1064 from Thorlabs.

The external source radiance depends on the launch conditions. For example, a satellite-to-ground free-space link is susceptible to radiance from the sun (the external source). Surprisingly, the radiance of these sources can be mitigated by the optics used in the free-space link and by limiting the communication link to only the horizontal direction. The lenses and fibers used in the link can only capture light with the appropriate numerical aperture; as a result, these components serve as a filter towards radiance that could interfere with the FSO signal. A bandpass filter will still be used to spectrally filter down any radiance that might make it to the system. Additionally, the system will only transmit signals at nominally horizontal angles between transmitter and receiver to prevent external light from entering the system.

Overall, the optical communications system will consist of two transceivers, two optical fibers, and two identical fiber collimators. The transceivers will be a third party

1000BASE-EX; each cost \$27.00 or less and can be delivered with little lead time. The optical fibers compatible with 1000BASE-EX transceivers include only SMFs. The fibers must have an LC/PC and FC/PC connectors. This increases lead time and varies the cost. The collimators that will be used will probably be a singlet lens collimator which costs between \$150.00 and \$400.00 each; if borrowed, cost and lead time are negligible. The fiber and collimator can be integrated together, as shown below.



Figure 6.10. Kinematic mount with a collimator lens ready to use in a free-space link.

Using the appropriate kinematic mount like shown in the figure above, the free-space link can be made seamlessly. The kinematic mounts can be borrowed from a laboratory and were tested, but not mentioned, for the prototype discussed in section 8.2.

Ideally, the more of these parts can be borrowed, the more feasible this system will be because of cost constraints. Our budget cannot exceed, and the team must work within its means. Thus, adequate consideration will be given to each component discussed in this section.

6.3 Power Delivery/Electrical Power System

The power delivery system draws power from the chosen power supply unit, the Mean Well LRS-50-5, which is connected through a USB-C port. This PSU supplies 5V to the entire port via this port. However, most of the components in the system require 3.3V to operate so this would require a voltage regulator to stepped down to 3.3V. Since we would have to connect the board to a computer to flash the MCU and program the algorithm, we also have a micro-USB port, which also supplies 5V to the system. These two power sources won't be connected at the same time but either way, both would need to be stepped down to 3.3V to safely operate the system. This part of the design is mostly unchanged in later revisions. To clarify, the only change made was that the power supply port is a USB-A port while we have an external CP2102 USB adapter that is being used to program the board. The actual voltage regulator is unchanged. Figure 6.11 shows this system.

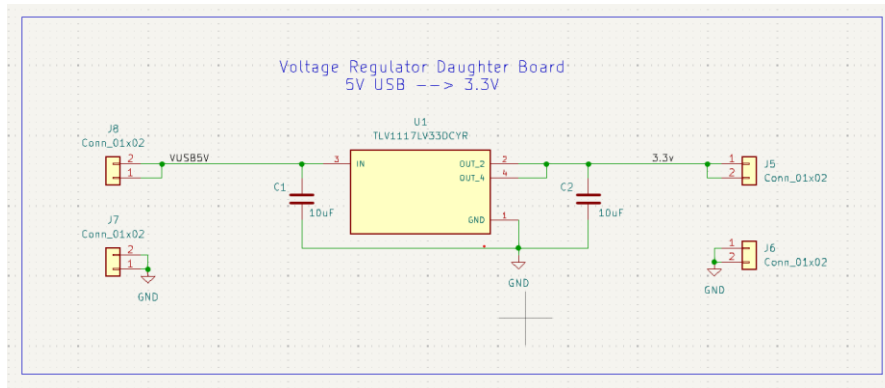


Figure 6.11. Schematics for the power delivery system

6.4 MCU/Peripheral System

For this project, the designed system needs to include three ports to function properly.

Those ports being HDMI, USB-A, and Ethernet. The main processing unit changed throughout the design process but would still need to be responsible for controlling all the peripherals in the system. Since this PCB's main functionality is to receive the compressed video data from the evaluation board through Ethernet to then decompress that video data and then send it to a display via HDMI. It will handle data processing, communication protocols, and computation. This design utilizes a USB interface for communication, as well as connections through its various GPIO ports to connect to the peripherals. We also have 3.3V power coming in from the voltage regulator stepping down from the 5V power supply. As we go through the design process, a lot of the design won't work in practicality, so we went through a handful of changes to the design, components, and functionality to lower flaws.

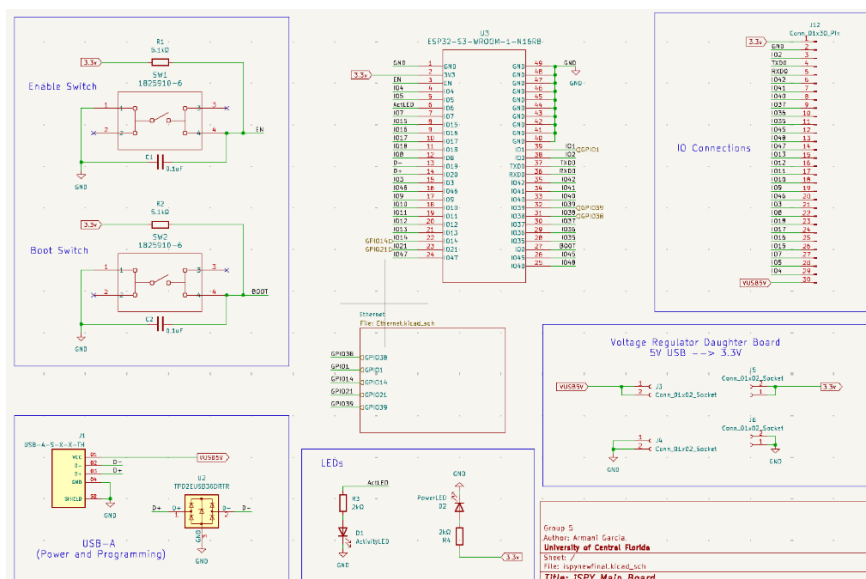


Figure 6.12. Schematics for MCU connections

One of these changes ended up resulting in the omission of the HDMI daughter board, since the chosen ESP32-S3 does not natively generate HDMI signals, which would've set us back further past the deadline. Also, we changed the general functionality of the system since the ESP32-S3 isn't that powerful and we had some internal clarification within the group about the purpose of the board. This board is supposed to only really act as a relay for the system, accepting the compressed bitstream data from the ethernet daughter board and send it straight to a display of some kind. Since, we omitted the HDMI board, we are going to send the video data through the CP2102 to a laptop or computer and display the video that way. After identifying the flaws, we were able to pivot the design and the priorities for the PCBs to suit the needs of the project.

6.4.1 Ethernet Connection from Transceiver

The ethernet daughter that connects on our PCB utilizes an ethernet PHY transceiver, the LAN8702A, to enable network communication and gives us a physical ethernet connection. Via Reduced Media Independent Interface (RMII), the MCU communicates with the LAN8702A which allows the efficient transfer of ethernet frames. This interface typically operates at 50MHz, minimizing the number of required pins while providing full-duplex 10/100 Mbps Ethernet capabilities.

The LAN8702A connects to an ethernet connector via differential pairs for transmission and reception (TX/RX respectively). These pairs ensure the signal integrity, so we had to make sure that the trace lengths were matched while also using controlled impedance which for ethernet is typically around 100 Ω . We included decoupling capacitors to enhance our electromagnetic interference protection.

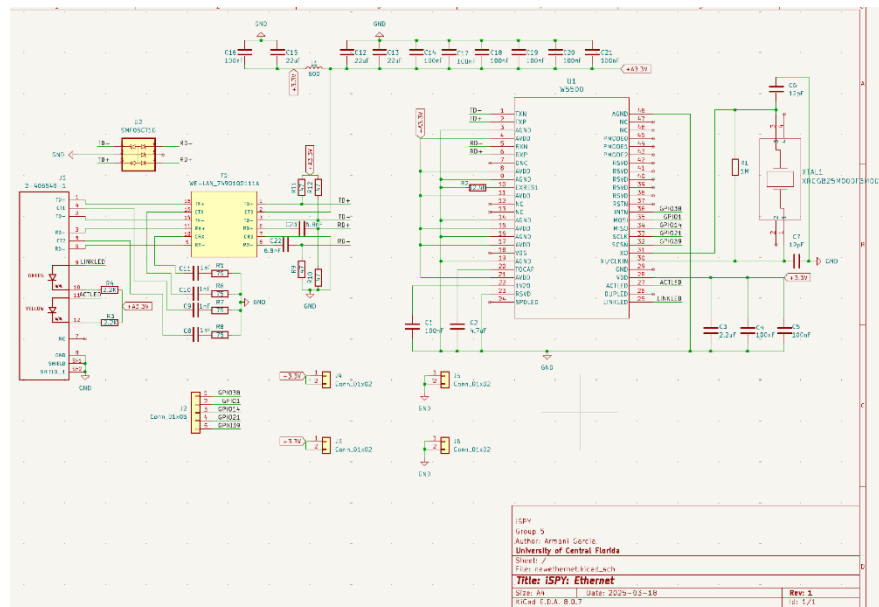


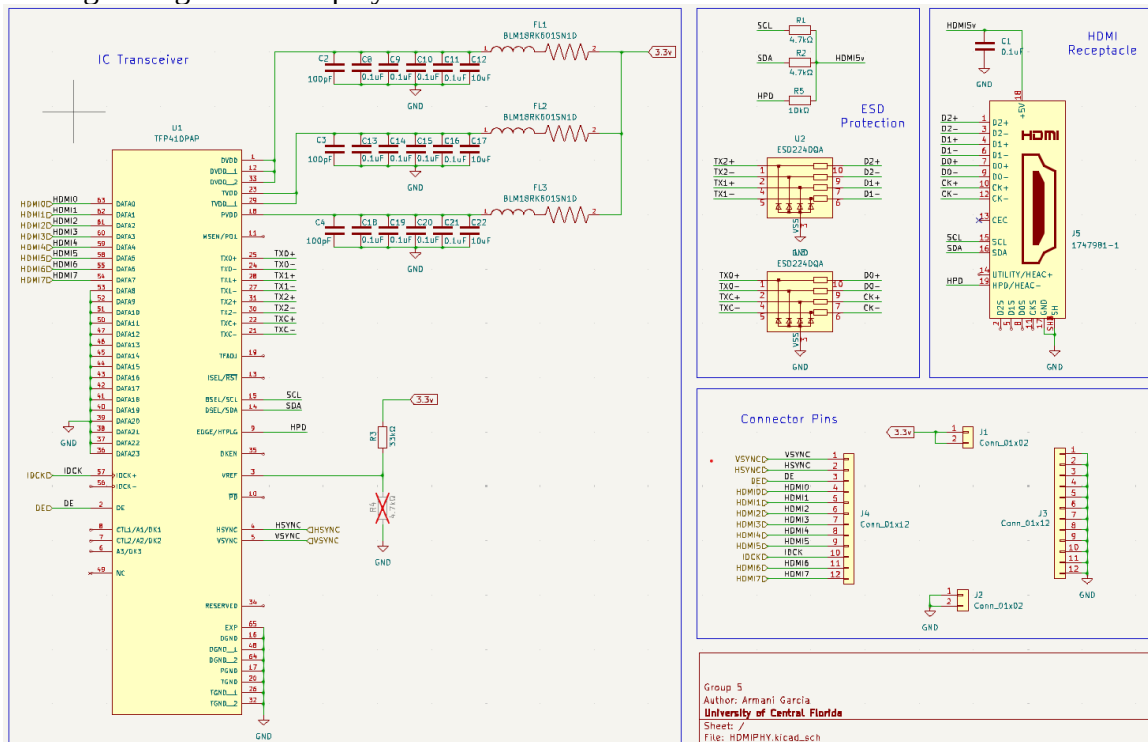
Figure 6.13. Schematics for Ethernet connections

Through further research into the chosen MCU and its capabilities, we later found out after the first iteration of constructing the PCB and testing, that the ESP32-S3 is not compatible

with the LAN8720, so we had to pivot and found that the W5500 was. Espressif, the creator of the ESP chips, had taken out the functionality of the Ethernet communication in the new models such as the ESP32-S3, which also included compatibility with the LAN8720, a much more powerful ethernet transmitter chip. Due to this we had to completely redesign this board to make sure everything was connected properly and so that left us with a much better designed PCB. With the W5500, we should be able to communicate from the optical transceiver to the ESP32 and allow us to receive data.

6.4.2 HDMI Connection to Display

The HDMI system is designed to transmit video signals from our MCU to an external display. The communication isn't one-to-one so we would need to utilize an HDMI transmitter that will take the form of the TFP410PAP. This unit converts the digital RGB video data into Transition Minimized Differential Signaling (TMDS) format. TMDS is a high-speed, differential signaling protocol ensuring reliable data transfer, minimizing noise and signal degradation, which is important when our entire project relies on clean video coming through to the display.



ignoring the Consumer Electronics Control (CEC) pin since no other HDMI devices are needed in the design. One of the most crucial aspects of this design was making sure that the differential pairs were length-matched and impedance-controlled to preserve our signal integrity. For HDMI data, the differential impedance is typically at 100 Ω like the ethernet pairs, so it is important to adjust the trace width and lengths to stay within the tolerance surrounding 100 Ω .

Although, we initially sought to display video via HDMI, we found during testing that the design of the HDMI daughter was not viable in the timeframe of the project. Therefore, we decided to withdraw the design for the final product and chose to display the video through an alternative route.

6.4.3 USB Connection for Programming

We initially had a micro-USB system that had two primary functions: powering the device when not plugged into the power supply and providing an interface when programming, debugging or USB communication with a host system. The micro-USB port allowed for the connection of a standard USB cable and can supply 5V power through the VBUS pin to the board. This 5V input is stepped down to 3.3V through voltage regulator in the power delivery system mentioned in Section 6.3.

The port used the STM32's embedded USB peripheral configured in Device Firmware Update (DFU) mode, which allows firmware to be uploaded with no additional hardware. This communication system also includes differential pairs in the form of D+ and D- data lines which are directly connected into the STM32's USB pins through ESD protection which ensures protection against voltage spikes. These data lines need to be carefully routed, but unlike the HDMI and ethernet differential pairs, we need to match the length of these traces and keep the differential impedance at around 90 Ω which is typical for USB data lines. We also utilize the BOOT0 pin on the STM32 and the reset circuit to allow the device to enter the DFU mode when booting.

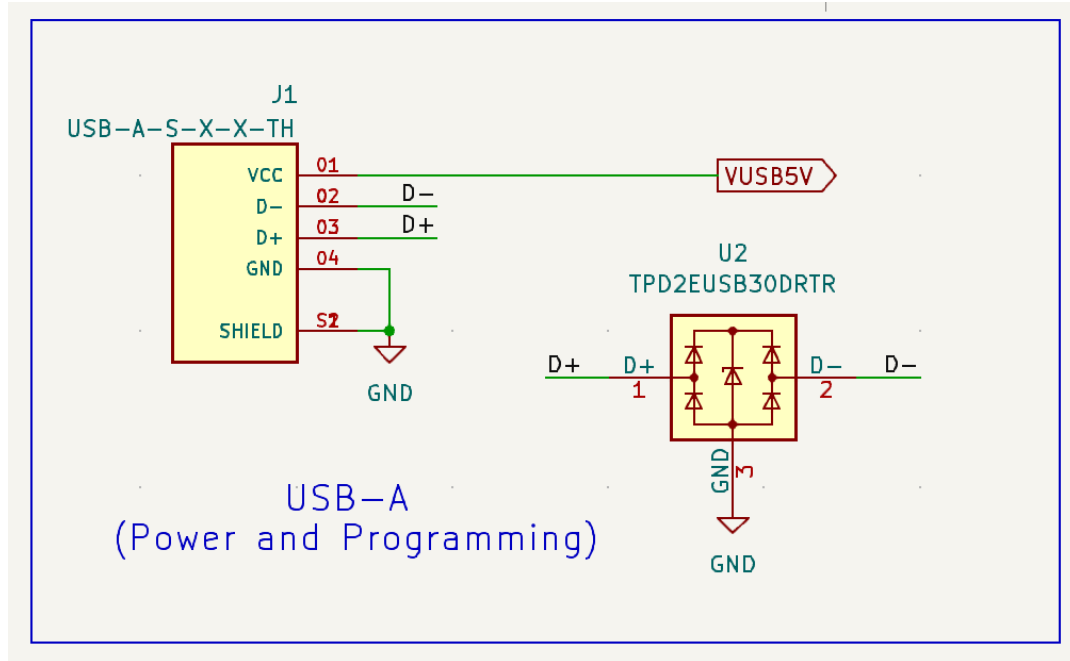


Figure 6.15. Schematics for USB-A connections

Since we had changed our MCU choice to the ESP32-S3, the connections from the USB-A/voltage regulator system differed slightly. We had to connect the D+ and D- signal lines to the respective pins on the MCU using the same method as mentioned previously, prioritizing the matched impedance and trace length. The other connections, like the 5V power line are made to the voltage regulator like before.

We also had naively neglected the inclusion of a programming chip like the CP2102 in the first revision. We had to solder extra wires to utilize an external CP2102 USB adapter since the pins necessary to use this adapter were occupied by the wrong connections, but in the second revision, we had traced those GPIO pins to a pin header so that we didn't have low quality solders that could result in a broken connection. Due to these changes we are able to effectively flash a program to the board and it allowed us to progress further into the project.

Chapter 7 – System Software Design

The software system is designed to handle the computer vision detection and recognition of individuals in iSPY. It also handles cleaning the image of each frame for the output that is shown on the display. It includes a counter that represents the number of people recognized and it will include a recording of frames for playback on our display.

7.1 Software Case Design

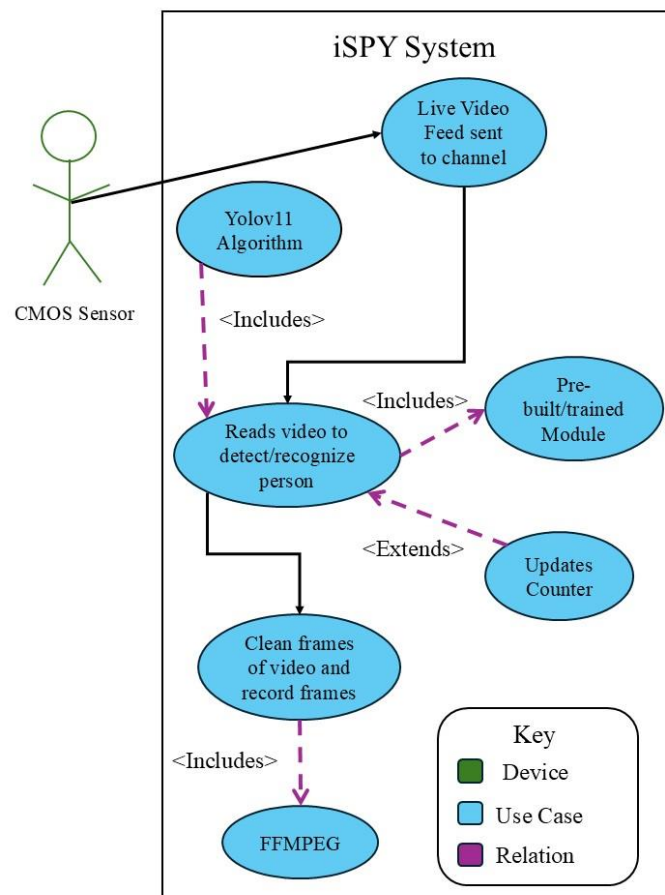


Figure 7.1. Use Case Diagram for iSPY system by Authors

In the above diagram the software being utilized is represented. The CMOS sensor, which is part of the camera module that takes in a live video feed will be connected to the computer vision program by the use of a channel like VideoCapture(0). Once the channel is set up this allows the computer vision program to connect to the CMOS sensor and allows the computer vision program to view what the camera module is seeing in the (Field of View) FOV.

The video frames are read by the computer vision pre-trained module/model to detect/recognize bodies of people that are in the given frames which includes the use of YOLOv11 algorithm as well as pre-built/trained module. Pre-built and trained module is used to detect/recognize bodies of individuals in order for the program to fully box out a person in the frame.

The counter will update every time a body is detected this will continuously update as new frames are sent to the program. After the detection/recognition process the video will be cleaned up meaning it will be dehazed, compressed and unblurred. In the cleaning of these frames from the video the use of FFmpeg-Python will be utilized which will allow the compression of the video as well as the filtering out of noise and deduction of haze in the frames. This will let the display show a cleaner image of the video frames. Also, allows recording of frames for playback.

In Figure 7.1. it includes the relation between the use case states and the external or internal components that will be used throughout the software program. The relation is either including the component or the component extends to the use case just depending on the use. Overall, this covers all of the software design and its use cases.

7.2 Software Flow Design

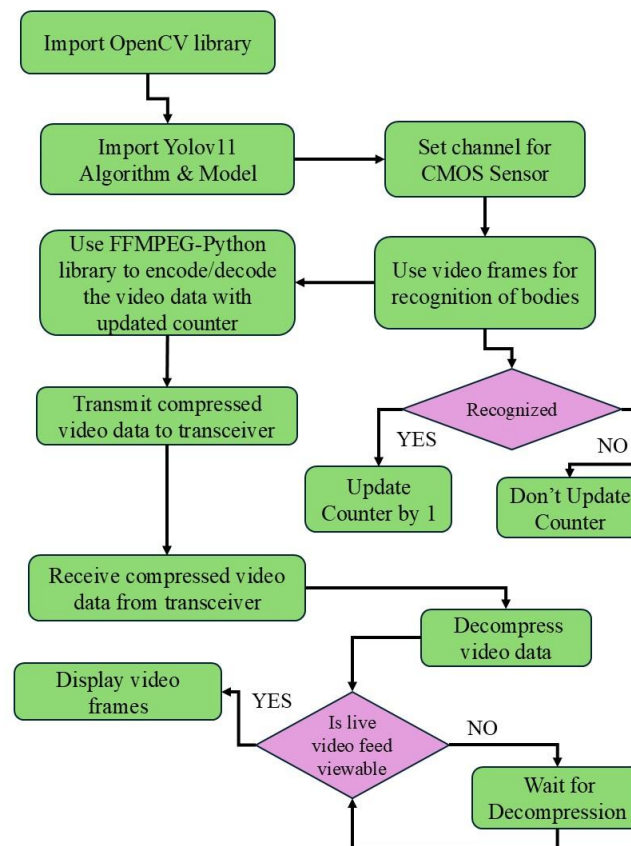


Figure 7.2. Software State Flowchart by Authors

The software state flowchart goes over the functionality of the software in our system from computer vision recognition to the cleaning of the frames and even the displaying of those frames. The software starts with importing the OpenCV library in python and importing the YOLOv11 algorithm as well as the model that will be utilized for recognition of bodies in the given video frame. In the software program there will be a line that sets the channel to connect to the CMOS sensor that is part of the camera module.

The video frames can then be used by a pre-trained model of YOLOv11 to recognize the bodies of people in the frames. If the body of the person is recognized, then a counter will be updated by 1 increment and if the body of the person is not recognized then the counter won't update. After this the video frames with the body recognition and updated counter will go through an encoding process which will use the h.264 codec. FFmpeg-Python will run the encoding process that will compress the video frames into a bitstream to be transmitted to the transceiver which will go through the FSO system.

FFmpeg-Python will also create a new video file that will take in the recorded video frames. The recorded video frames will include the recognition of people and the updated counter amount in each frame. The recorded video frames can be used as playback for previous frames that were already compressed with all the data from the computer vision program. This allows the iSPY system to be more like a real security camera system which all allows video playback in order to see what happened in prior video frames or surveillance feed.

Once the bitstream comes back into the transceiver, the bitstream is received. The use of FFmpeg-Python will take the bitstream and decode it into the video frames which is the decompression of the video frame data. After the video frames are decompressed the program will check if the live video feed is viewable. If the live video feed is viewable then it will send the video feed to the display and if the live video feed is not viewable the program waits until the decompression of video data is done.

7.3 Computer Vision Setup

In this section the setup of the computer vision part of the program will be detailed out with all necessary setup procedures and design. The design entails importing the libraries and pre-trained models. Setup for computer vision has many levels from the algorithm to taking in image or video. The setup will be shown in rough prototypes of our computer vision program in Chapter 8 section 3. The process would include channel setup for the CMOS sensor, scaling the image or frames, function for recognition and updating the counter for the amount of people recognized.

The channel is generally used for connections for the program to access some device whether that be a camera or image sensor to a webcam. The scaling of the image or frame is done for the program to actually be able to scan through the image or frame which allows the function for recognition to actually recognize a person then the counter can be updated based off of the recognition.

In the specific setup I focused on the algorithm likely being utilized in the iSPY system. The setup would be for the YOLOv11 algorithm. This is the latest computer vision algorithm that recently was released by Ultralytics which was mentioned before in Chapter 3. The decision was made to mimic the computer vision setup based on the likely usage of certain aspects such as algorithm to formatting of the recognition function. This setup would be dependent on factors in the overall structure of the software system, meaning that if the model chosen causes latency, we would have to change the model selection to another model that doesn't cause latency or that much latency. Other factors can determine the structure as well such as the image acquisition system setup or design.

7.3.1 Importing Library and Model

The first thing to do for the program for computer vision is to import the necessary library and model being used. When it comes to importing the library, we will import OpenCV as our library to use by this command `import cv2`. Although to import the YOLOv11 we will import YOLO from Ultralytics and choose the model to use. There are other ways to import YOLOv11 which are more non-direct like converting the model to ONNX format which is only one of the other ways. Once the model is chosen, we can use this model in our recognition process of bodies. YOLOv11 has five different models which all have their own accuracy rating and latencies. The models are scaled from lightweight to large weight which have been covered in Chapter 3 section 7.

7.3.2 Setting the Channel

The channel that will connect the CMOS sensor to our computer vision program will be set up before we run our recognition of bodies to indicate that there is a person in the video frame. In order to set up the channel we need to figure out what channel to use usually '0' is a webcam which means that '1' or '2' possibly is the sensor. Once we set the channel using `VideoCapture(1)` then the CMOS sensor will be connected to the program and allow the computer vision to see what the camera module is seeing in each video frame. This will allow the recognition function to be able to be used with the video frames from the camera module.

Since we used Raspberry Pi 5 as our microcontroller for handling the computer vision aspect the setup for the channel would be slightly different. The channel would utilize the `picam2` and PiCamera setup which is shown in Appendix E. The setup included sizing (resolution) of the input frames that the CMOS sensor (Pi Camera) would be capturing. Also to read any frames from the `picam` we needed to use a function known as `picam.array()` which captures the frames continuously.

7.3.3 Scaling Image or Frame

Before the recognition function, the image or frame is scaled down so that the image or frame can be used in the recognition function. The scaling is different depending on the algorithm. The scaling in Haar Cascade and HOG is done by the use of `detectMultiScale`

method which is shown in the code examples in Appendix E. However, when it comes to scaling in YOLOv11 it is done by putting the image or frame read by the computer vision program into the model of YOLOv11 being used such as yolo11n. The scaling is automatic in the YOLOv11 algorithm which allows the image to be automatically scaled down to 640 x 640 which is shown in Figure 3.10. The size in the figure represents the size of the image or frame that the YOLOv11 models resize to in order for the recognition function to work. This resizing is all done by passing the image or frame to the model which will take the image or frame and automatically resize it so that the model can actually work with it.

The Pi Cam setup allowed us to resize the main resolution we took in from the camera (sensor) itself. Although we still needed to adjust the scaling for YOLOv11 to handle the incoming frames from the Image Acquisition System which is shown in Appendix E under the sub-title Computer Vision Raspberry Pi Code. Even if the model automatically resizes it wasn't enough to handle the incoming frames (data) coming from the sensor to the Raspberry Pi itself.

7.3.4 Function for Recognition

When it comes to the main part of the program, which is the body recognition function, we had many aspects to create in order for the function to do the recognition. We started by defining a person which is optional and moved to how we define them with labeling which is required. Once the person is labeled by recognition of their body, we can put dimensions of a rectangle box around the person if their body is recognized by the program. We will be able to adjust the box by using a 2D array representing x and y. This x and y create a bounding box that goes from the center of the person to the outside depending on the sizing constraints. The result includes a number of people recognized in the video frames which were implemented by a counter. After we have established a result then we can send this into the FFmpeg portion of the program.

7.3.5 Updating Counter

Firstly, we needed to initialize the counter by setting the counter to 0 or null meaning that a person has not been recognized yet or that there is no one in the video frame. If the person is recognized in the video frame, then the counter updates by 1. The counter continues to update until the program is terminated or there are no people in view of the camera model. This number is displayed on the display in our system after going through the FFmpeg process.

We had the counter being determined by the confidence level and not the person. After noticing this we adjusted the counter by including object tracking in the computer vision program which is in the Computer Vision 1st Iteration located in Appendix E. The usage of object tracking allowed us to track the individual people in each frame which kept the count of how many people either entered or exited the frames.

7.4 Computer Vision Program

In the above setup process, I mentioned the focus on YOLOv11 algorithm. The YOLOv11 algorithm was simulated, and the simulation of this algorithm was tested as well as verified. I chose to use the YOLOv11 algorithm as our computer vision algorithm to be used in our computer vision program.

The model would be determined by other factors. The factors that were focused on determining the model were latency and accuracy. I first started with yolo11n which is the lowest lightweight model. The yolo11n model was pretty accurate although it had high latency. This meant that the yolo11n model would not be the most suitable for iSPY. The next model was yolo11s, which was accurate and didn't have much latency.

It is a little more advanced in the recognition of a person within the image. However, there was a loss in a few people recognized as a person. Mainly the ones in the distance that were not very clearly defined were not recognized by the yolo11s model. I noticed that yolo11s model was not a very balanced model do to it losing the recognition of people that were farther away in the image. Although the people that were closer were recognized and those that were blurred out were not recognized as well.

I went even further with the models of the YOLOv11 algorithm to yolo11m model. This model is a medium weight model meaning that it is in between lightweight and large weight models. The recognition using the yolo11m model was similar to the yolo11s model though it was able to recognize smaller objects like handbags. It was impressive on the bounding boxes and confidence scores that were determined by the medium weight model.

The other two models are large weight models and are even more advanced though they wouldn't change much except for the confidence score. I determined that YOLOv11 would be able to handle the overall recognition of people in our system. YOLOv11 was the best overall in the way of accuracy as well as not having that much latency like other algorithms. It also would be able to handle real-time recognition from a camera module that would be viewing people going in and out of frames.

Computer vision program is generated by the algorithm that is used to either detect or recognize. In iSPY our computer vision program handles real-time recognition of people. This real-time recognition can be done using more advanced methods or algorithms. It is going to take in live video feed and run the recognition process on each frame. The computer vision program recognized people in each frame continuously which meant that it needed to be set up to continuously read those frames and use a more rigorous algorithm. This is another reason for YOLOv11 being used as our overall computer vision algorithm for our system.

7.5 Video Compression Setup

In this section, we will go into detail about the process of setting up FFmpeg via Python. FFmpeg is what we used to compress and decompress our live video feed for transmission. We needed to establish libraries and make sure we have everything up to date.

7.5.1 Installation of Python and FFmpeg

Python needs to be on the board to even begin so a command was run to check if python was on the board and if the version was up to date. Python 3.11 was installed on the board. FFmpeg is a library that can be imported into Python code, but it needs to be installed and up to date. Commands were run to check if the packages were installed. Once Python and FFmpeg were properly installed, we could proceed with the design of the video compression software. FFmpeg is part of the library called subprocess. This library is pre-installed on the Raspberry Pi but it's important to check that everything is installed correctly.

I (Luke) recorded a small video to upload to Raspberry Pi to be used as a test video to run FFmpeg on. The video was roughly fourteen seconds long, 1920 by 1080 resolution, and recorded at 60 fps. I made sure to upload the uncompressed version of this video. The final size after transfer to the board was 37.6 MB. This was filmed on an iPhone with the file format .mov (important for later). I created a Python file to begin experimenting with FFmpeg. I had never worked with FFmpeg before, so I had no idea where to start. I had to do my own research to figure out how to properly feed a command into the code. I found a helpful article by Maab Saleem going over the ins and outs of how to compress a video.

7.5.2 Compression

There are many parameters to input when trying to use FFmpeg. The first input is “ffmpeg” followed by “-i input_file_name” [222]. This makes sure that it is taking in the proper file and not compressing a random video in the same directory. There are a few codecs available when using FFmpeg. The one that is used in this experimentation of code is H.264, which we did in depth research for in Chapter 3. We can access the codec using the command “-c:v libx264” [222]. The video being used for compression testing is a .mov file. By default, most video files are mp4. To convert the .mov file to a usable mp4 file that H.264 can properly use, the command “-pix_fmt yuv420p” is used [222]. The option to convert it to an mp4 is used for H.264 but I have it being converted back at the end to a .mov file for consistency. Constant Rate Factor (CRF) is another way of compressing the video. There are values between 0-51. Lower values offer a higher quality at the cost of larger file sizes. Higher values offer smaller file sizes at the cost of image quality. Setting the value to 0 results in lossless video compression (which we discussed in Chapter 3). For testing purposes, this was left at the default value of 23 using the command “-crf 23” [222].

The video that was uploaded for testing purposes was in 1080p resolution with 60 fps. This is more than we aim to use for our prototype, so the video is going to be compressed to 720p 30 fps for the test. We can change the resolution of our video with the command “-vf scale=1280:720” [222]. This automatically reduces the resolution which did in turn reduce

the size. 60 fps is unnecessary for our system as we want to be as efficient as possible. 30 fps is plenty to accurately see what is being displayed on a surveillance system. The command to change the frame rate is “-r 30” [222]. When the number is changed in the line of code, it will set the compression to the frame rate input.

We want to make sure the compression is fast. This is going to be compressing video feed in real time in the final product. We can specify the speed of the compression using the command “-preset fast” [222]. The word “fast” in the command can be changed based on the preference of the programmer. There are five presets: ultrafast, fast, medium, slow, and placebo [222]. Why not pick ultrafast every time? The quality is decreased when using ultrafast because it’s trying to go as fast as possible instead of being slow and making sure quality is ensured. In this example, the ultrafast preset is going to be used. There is a simple if statement to check if the file is in the directory. If the file is present, the code proceeds with the compression. The final compressed video is 7.9 MB. This size is roughly 21% of the original size.

Decompression is a similar process to compression when it comes to FFmpeg commands. First, the output file created when compressing the original video is now taken as the input file. Most of the commands are the same, however we want to set CRF to 0 so there is no data loss. The resolution is upscaled to 1080p and the frame rate is upscaled to 60 fps. The decompressed file’s final size was a staggering 234.5 MB. That’s nearly 30 times the size of the video! Thankfully, this is only happening because of the way we’re running this test. There is no cap on the bitrate, so it is generating an absurdly large bitrate. In the final product, it converts it back from bitstream to a lower bitrate which is much more efficient.

Chapter 8 – Prototype Construction

In this chapter we will dive into the details of the prototype construction of each sub-system and how it reflects the final product. This prototype discussion is important because it gives the audience an idea of what the final system looks like, it enables the readers to explore the authors' underlying ideas and approaches.

8.1 Image Acquisition System Prototype

The prototype of the imaging acquisition system was built as proof of concept, and it was presented during CREOL's second demo on November 21st, and the Mini Demo Video on November 26th.

As any prototype does, it gave us a rough idea of how the final product works. It was built with the same elements that are used at the end, the difference lies in the size of the lenses and the CMOS sensor, and their distances. Since the exact parts we need for this subsystem are not something that usually runs out of stock we decided to buy them during the winter break (after the semester is over).

This prototype was created with the goal of solely showcasing the impact of focal length on the field of view. The layout is relatively simple, which is not the case for the final product. The set-up demonstrated can be thought of as a really long camera, with the sensor first followed by the lenses.

The lenses used were Newport's KBX031 and KBC022AR.16. KBX03, is a biconvex (BCX) lens, uncoated, with a diameter of 12.7 mm, focal length of 38.1 mm, and made of BK-7 glass. KBC022AR.16, is a double concave (BCC) lens, with a diameter of 12.7 mm, focal length of -50 mm, also made with BK-7 glass, however, this lens is coated with BBAR for wavelengths in the range 0.65-1.0 μ m. These lenses were acquired in the Senior Design Lab, and as previously mentioned, they are only being used as placeholders. Their diameter size, focal length, and coating do not match the requirements of our system; however, the overall result shows a portion of what we want to accomplish in the future.

The CMOS sensor used for the set-up was Thorlabs' DCC1645C. It captures images in color, and it has an electronic rolling shutter like the sensor we chose for iSPY. Unfortunately, this specific sensor type has been discontinued and we only have access to it because the CREOL undergraduate lab still has them. We did search for sensors of the same brand, but the price is beyond our budget.

A schematic of the set-up can be seen in Figure 8.1.

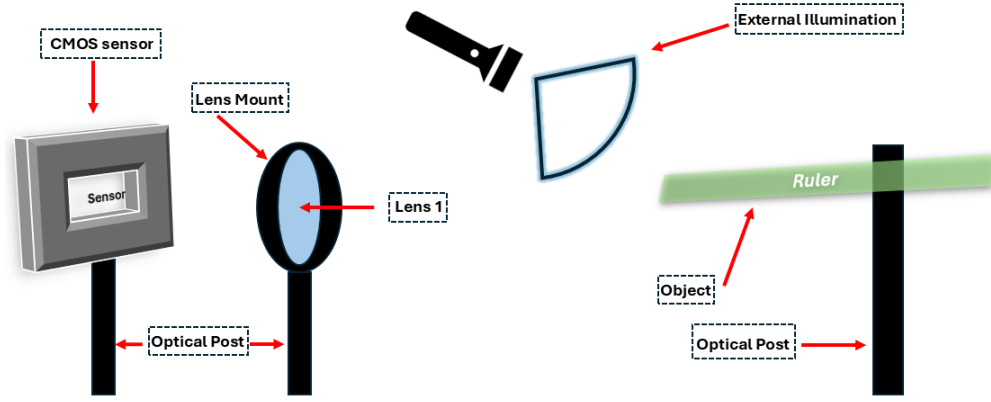


Figure 8.1. Diagram by Authors of the imaging acquisition prototype set-up with a single lens.

During the testing of our prototype, the room lights were turned off and the object we imagined was illuminated. This procedure a common practice in optics labs and its purpose is to reduce background noise caused by the room lights, to increase the contrast between the object and the background, and to eliminate unwanted reflections and shadows.

In the first step of this prototype set-up, a CMOS sensor was connected to a laptop at one end of the table, and an object was placed at the other end. The positive lens (KBX031) was then set in between the sensor and the object. The goal was to image the object, in this case, a ruler, onto the sensor.

Since only a single lens was used, its focal length is the same as the system's effective focal length, 38.1 mm. We could estimate the image location using the imaging equation (Equation 3.4). This equation is for thin lenses, so the results are not 100% transferable; however, it is used as reliable approximation.

$$\frac{1}{f} = \frac{1}{d_o} + \frac{1}{d_i}, \text{ where } f = 38.1\text{mm}, d_o = 290\text{mm}$$

$$\frac{1}{d_i} = \frac{1}{38.1} - \frac{1}{290}$$

$$d_i = 43.86\text{mm}$$

The diagonal size of the sensor used was 5.90mm. The field of view can be estimated using equation 3.1.

$$\text{FOV (in degrees)} = 2 \tan^{-1}\left(\frac{w/2}{f}\right), \text{ where } f = 38.1\text{mm}, w = 5.90\text{mm}$$

$$\text{FOV (in degrees)} = 2 \tan^{-1}\left(\frac{5.90/2}{38.1}\right)$$

$$\text{FOV} = 8.85^\circ$$

Figure 8.2 shows the outcome of this step. The details of the ruler are clear; however, we can only the bottom part and a little over 2.5 cm of it, the narrow field of view is noticeable.



Figure 8.2. Frame taken by the laptop connected to the CMOS sensor. A ruler was imaged onto the sensor, we can only see the bottom part of it.

For the second part of the set-up test, a second lens (KBC022AR.16) was added in between the first lens and the object. This addition drastically changed the effective focal length. To estimate the new system's effective focal length, we will use equation 8.1.

$$\frac{1}{f_{12}} = \frac{1}{f_1} + \frac{1}{f_2} - \frac{d_{12}}{f_1 f_2}$$

f_{12} is the effective focal length, f_1 is the focal length of the first lens, f_2 is the focal length of the second lens, and d_{12} is the distance between the two lenses.

$$f_1 = 38.1\text{mm}$$

$$f_2 = -50\text{mm}$$

$$d_{12} = 135\text{mm}$$

$$\frac{1}{f_{12}} = \frac{1}{38.1} + \frac{1}{-50} - \frac{135}{(38.1)(-50)}$$

$$f_{12} = 12.97\text{mm}$$

The image distance was then calculated, d_o has a new value as the distance between the lens (now, the last lens) and the object decreased.

$$\frac{1}{f} = \frac{1}{d_o} + \frac{1}{d_i}, \text{ where } f = 12.97\text{mm}, d_o = 150\text{mm}$$

$$\frac{1}{d_i} = \frac{1}{12.97} - \frac{1}{150}$$

$$d_i = 14.2\text{mm}$$

The image distance, d_i is in reference to the effective principal plane, somewhere between our lenses. In this case, rather than trying to find exactly where the principal plane was located, we adjusted the lens positions until we could image the object onto the sensor.

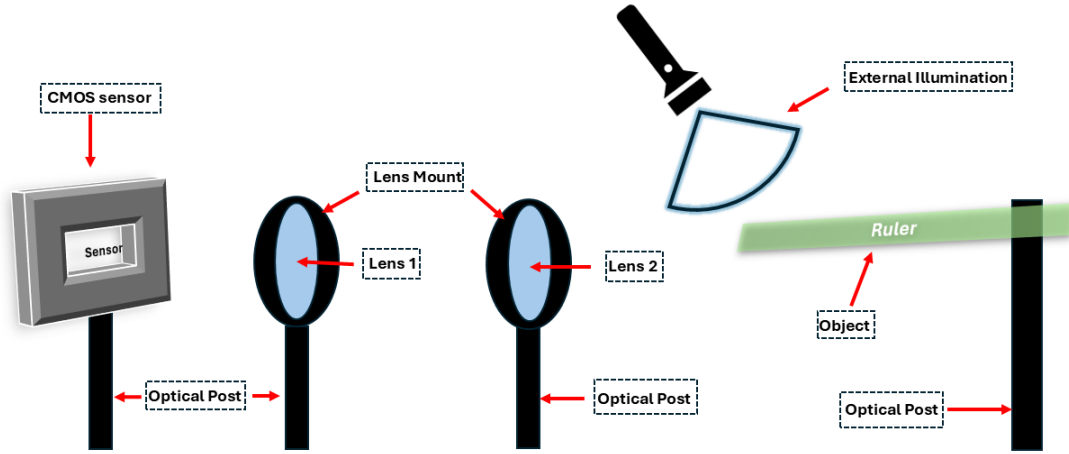


Figure 8.3. Diagram by Authors of the imaging acquisition prototype set-up with two lenses.

The CMOS sensor used was the same so to calculate the new field of view we only changed the focal length.

$$\text{FOV (in degrees)} = 2 \tan^{-1}\left(\frac{w/2}{f}\right), \text{ where } f = 12.97\text{mm}, w = 5.90\text{mm}$$

$$\text{FOV (in degrees)} = 2 \tan^{-1}\left(\frac{5.90/2}{12.97}\right)$$

$$\text{FOV} = 25.63^\circ$$

Figure 8.4 shows the end result of our system, and figure 8.5 shows the actual set-up that figure 8.3 represents.



Figure 8.4. Frame taken by the laptop connected to the CMOS sensor. A ruler was imaged onto the sensor, and we can see the top and bottom of it.

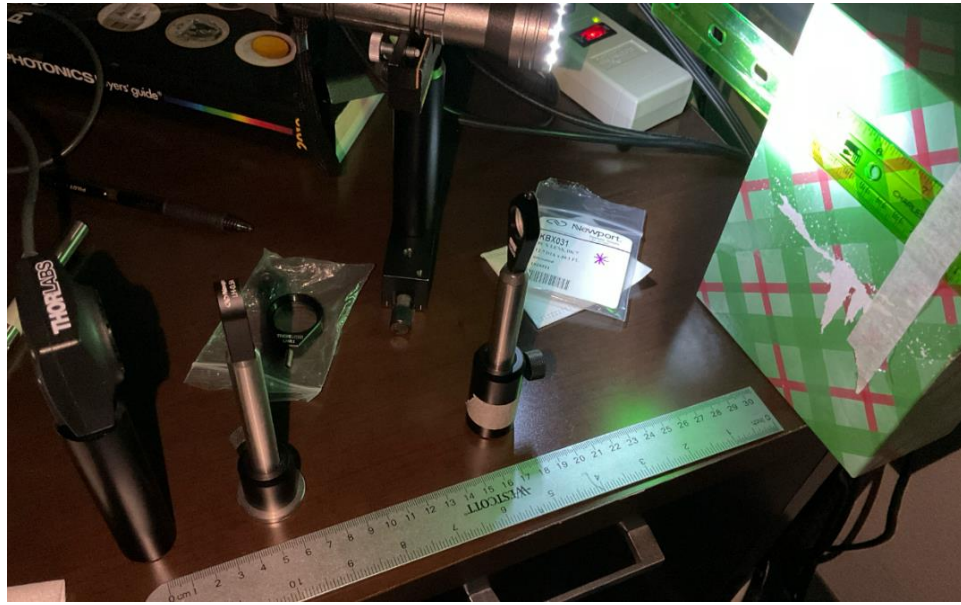


Figure 8.5. An actual image of how the prototype was put together.

A couple of details to highlight are the clear increase of the field of view, we can now see the top and bottom of the ruler, and the degradation of the image due to the addition of another lens. Imaging systems are simple when there is only one lens, but as soon as more lenses come into play the room for error expands exponentially. In figure 8.4 the details are not clear like in figure 8.2, and on the edge of the image we can see vignetting.

We knew such issues would arise, so while working on the lens system and thinking of how it would be mounted, we took into consideration such details. The final image acquisition system has a shorter focal length, and consequently, a greater field of view. The quality of the images increased, as we obtained better resolution and less aberrations, and it is possible to capture the images in a bright place. Such improvements are meant to be achieved in a compact form, increasing the challenge of this portion of the system. An in-depth discussion of the challenges faced, and the trials that took place to create the final optical design can be found in Chapter 9, section 1.

The final prototype, the one used for the Live Demo (April 15th, 2025) and the CREOL showcase (April 17th, 2025) uses the optics and sensor previously mentioned in Chapter 3 (section 3.1.2) and Chapter 6 (section 6.1).

Several trials were conducted to reach the final design. The distance between the elements was short, making the mounting of it all challenging. However, after weeks of testing, we were able to reach a successful structure. The final structure for the image acquisition system was inspired by Matryoshka dolls, also known as nesting dolls, in which we start with a larger piece and inside we find the smaller ones. Each piece is the mount for one element (pinhole and lenses). On figure 8.6, the cad model and the final structure can be seen.



Figure 8.6. Final structure of Image Acquisition System

Using this structure, we were able to reach the required distance between the elements. This was proven by carrying out a test to measure our FOV. We used grid paper and marked the extremities of what was captured by the camera. Since the size of the grid squares is known, we could find the real measurements and from that we used the formula below to obtain the FOV.

$$\theta_{V \text{ or } H} = 2 \tan^{-1} \left(\frac{\text{Scene dimensions}}{2d} \right)$$

Where d= distance from the camera and scene dimensions is the real distance (how many grids we can capture times the distance between them). The figure below displays how

the test was carried out. More information about this test can be found in our website in our midterm video for Senior Design II.

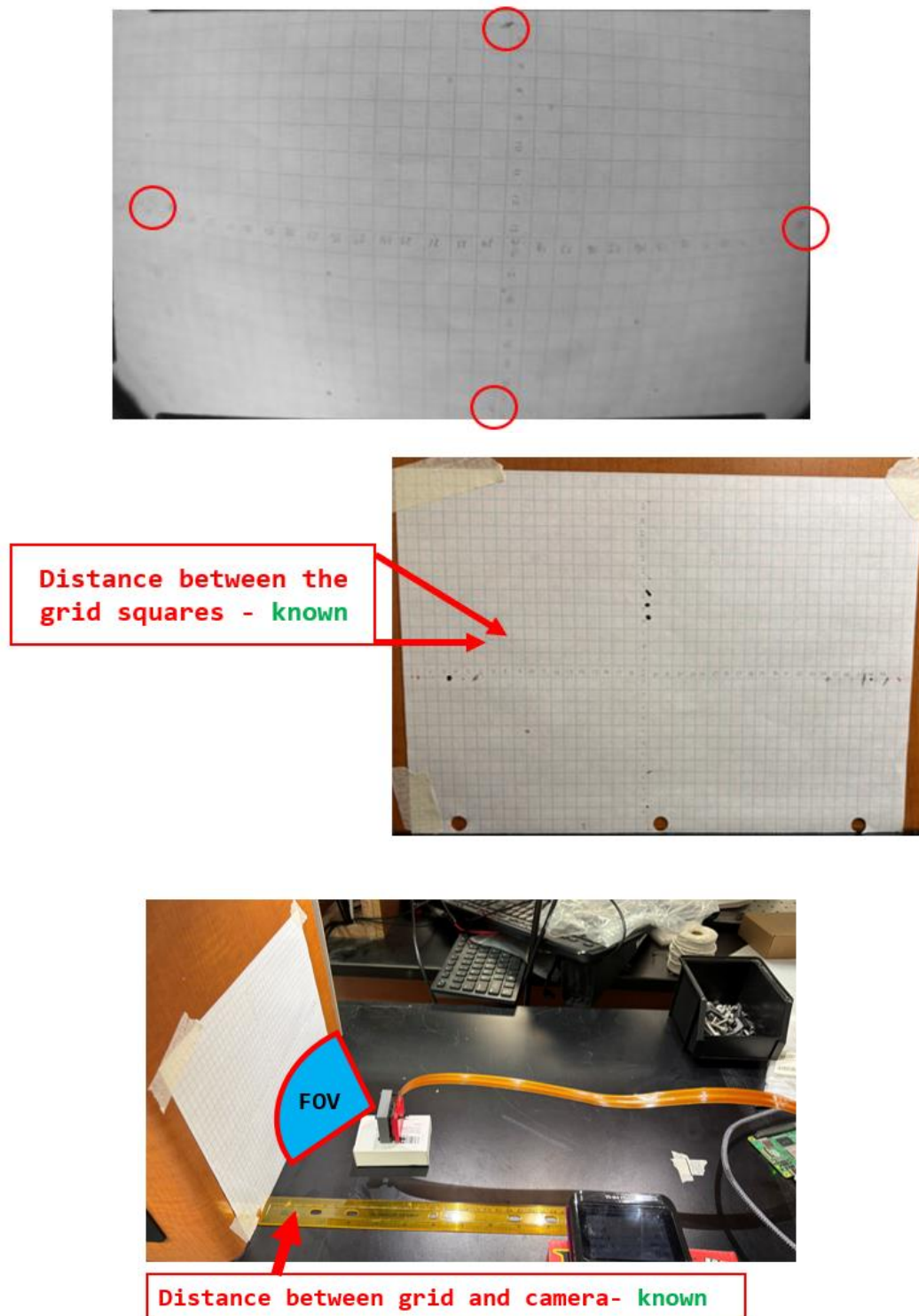


Figure 8.7. FOV test step by step

The following screenshots show how the image looks after it is captured by our system. As you can see, we are able to count everyone in the frame, and even the expected blur along the edges do not decrease our accuracy of detection. From the beginning, we wanted to make sure our image acquisition system was able to perform well with less elements than the current options on the market – and this came true.

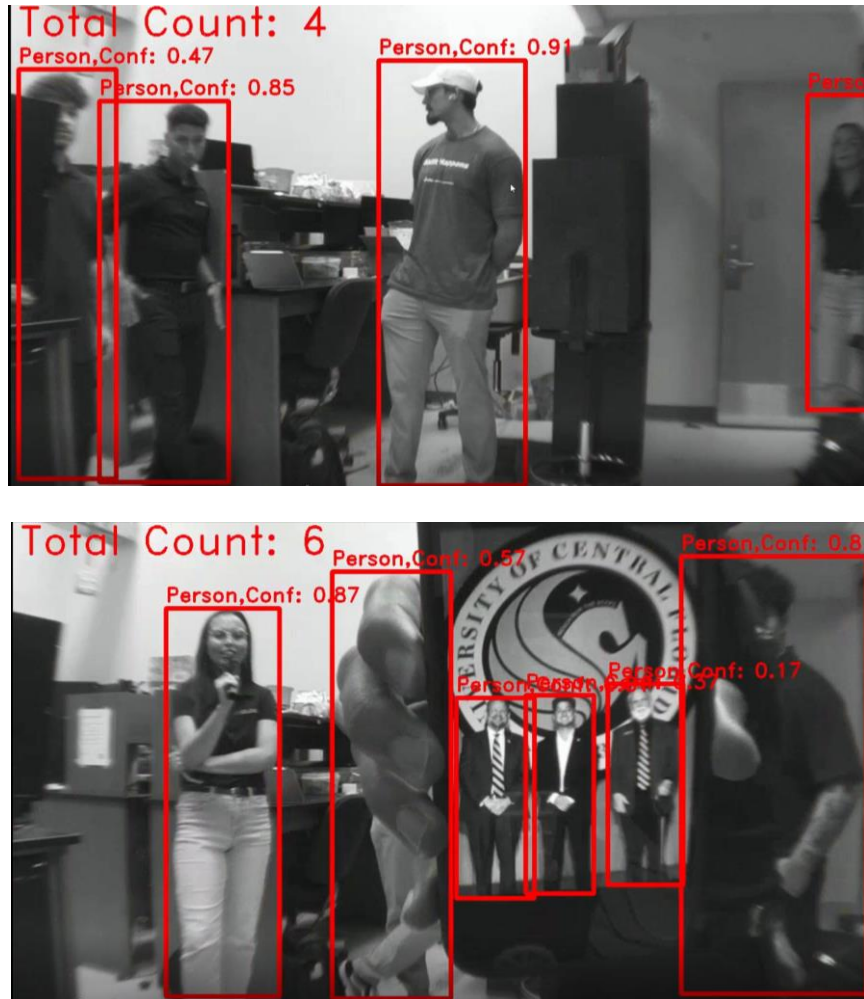


Figure 8.8. Live video capture by iSPY

The blur was expected due to the limitations of time and money. However, we adjusted our code accordingly. We understand that adding more elements to the system would increase the cost. However, optimizing the code would not. Therefore, we decided to follow this route.

8.2 Optical Communication Prototype

The optical communication prototype is to serve as a proof of concept for optical wireless communication of a signal of choice. Two prototypes were designed during senior design 1 and both were presented to faculty and students of CREOL, the College of Optics and Photonics, at the University of Central Florida.

The first prototype was designed in preparation for CREOL's midterm senior design demonstration. The goal for this prototype was to showcase the idea of using light to cause an electronic response since the main idea behind optical wireless communication is dependent upon whether an optical signal can be used to induce an electronic response at a receiver. To accomplish this goal, the objective of transmitting a binary optical signal by means of optical on-off keying was pursued.

Optical on-off keying can be understood as binary modulation of a light source – light on is logic 1 and light off is logic 0. To fulfill the objective, a light source, a detector, and an electronic response for feedback were critical elements for the first prototype. Due to the limited availability of materials, our critical elements of choice include a red LED, a photoresistor, and another LED as light source, detector, and feedback component, respectively. Together, the following circuits were constructed.

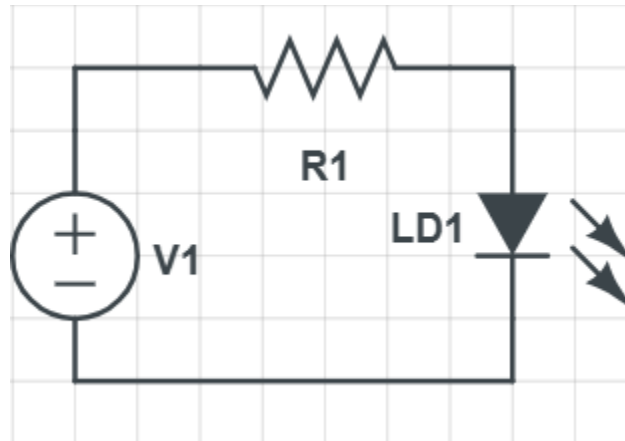


Figure 8.9. Transmitter for prototype #1.

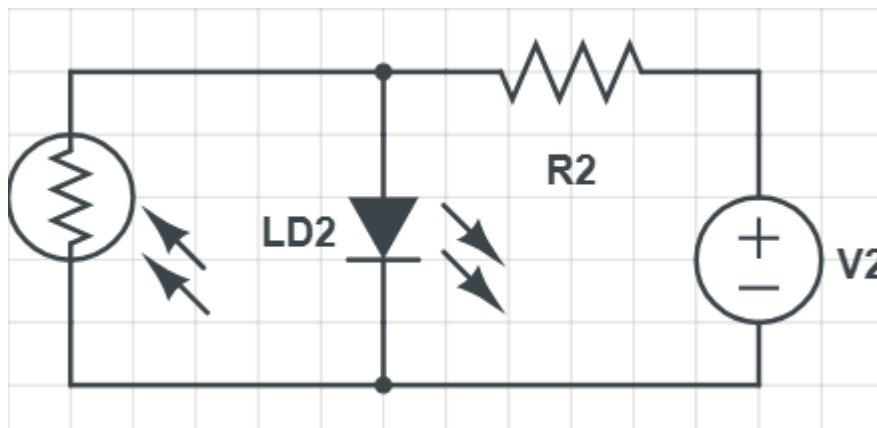


Figure 8.10. Receiver for prototype #1.

The transmitter circuit makes use of a simple LED circuit composed of a resistor and an LED. The resistor is in series with the LED to protect the LED. The LED has a forward voltage of 1.7 V. The signal source powering up the circuit is a function generator oscillating a 1 V_{PEAK} signal with a 2.5 V offset. This modulation technique gives the LED

enough modulation depth to appear “off” when the input signal is 2.5 V and “on” when the input signal is 3.5 V. The oscillation frequency from the function generator was set to 1 Hz to exaggerate the optical signal detected by the photoresistor.

The receiver circuit makes use of an LED and photoresistor in parallel with one another. This configuration enables the LED in parallel to be efficiently modulated “on” and “off” by the photoresistor. As the transmitter LED flickers “on” and “off”, the photoresistor flickers “on” and “off”. When the photoresistor is “on”, the photoresistor exhibits highest resistance; when the photoresistor is “off”, the photoresistor exhibits lowest resistance. Therefore, when the transmitter LED is “on”, the receiver photoresistor turns “off”, generating a short circuit bypassing the receiver LED and, thus, turning it “off”. As the transmitter LED flickers, the receiver LED flickers opposite the transmitter, completing the optical wireless link. Both circuits together are shown below.

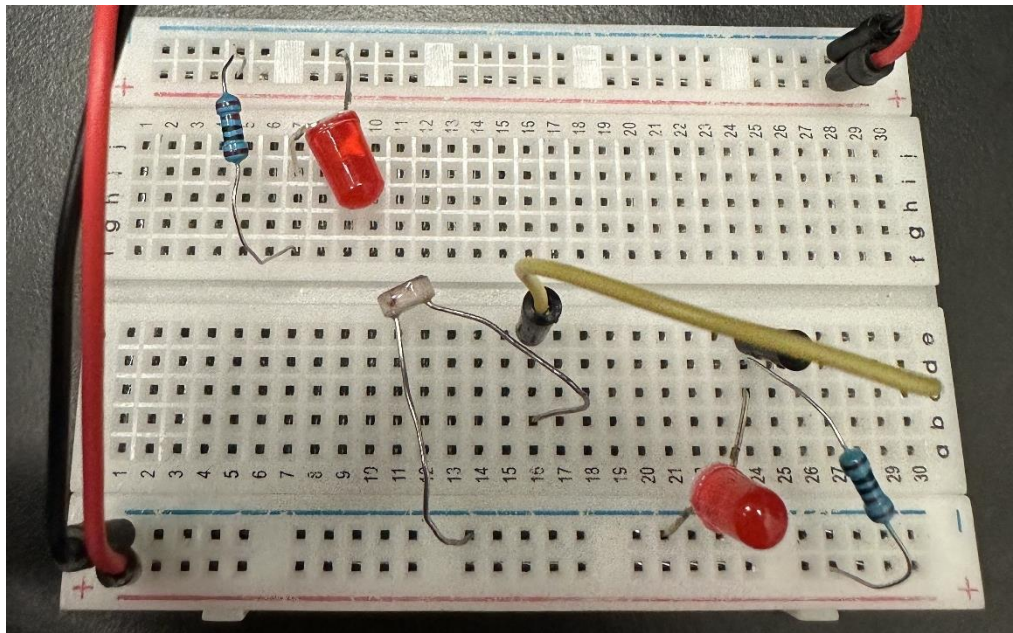


Figure 8.11. Completed prototype #1.

The second prototype was designed in preparation for CREOL’s final senior design demonstration. The goal for this prototype was to showcase the transmission of an audio signal using light. This prototype builds on the first by going beyond binary signals. To accomplish this goal, the objective of transmitting an amplitude modulated optical signal by means of optical on-off keying was pursued.

Unlike the first prototype, the transmitter was modulated by controlling the current passing through the LED. Likewise, the receiver is now a phototransistor and generates current with incident light. Since most optical wireless communication systems have separate transmitter and receiver circuits, this prototype makes use of two independent circuits as shown below.

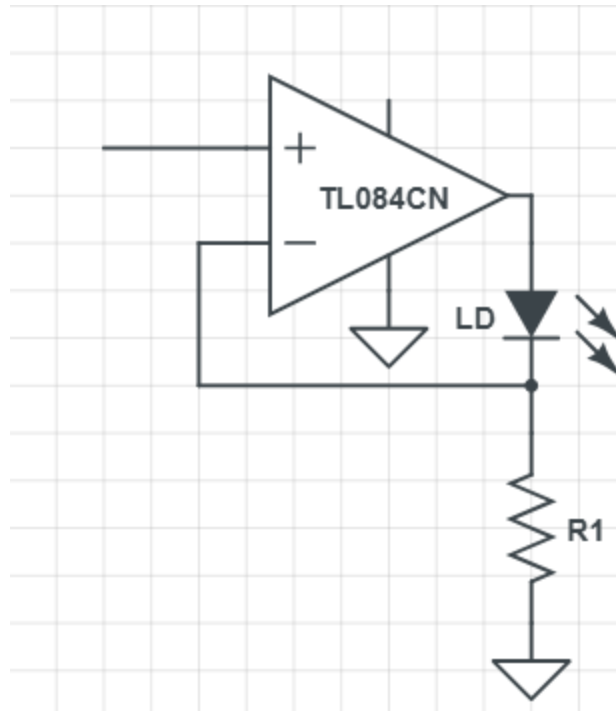


Figure 8.12. Transmitter for prototype #2.

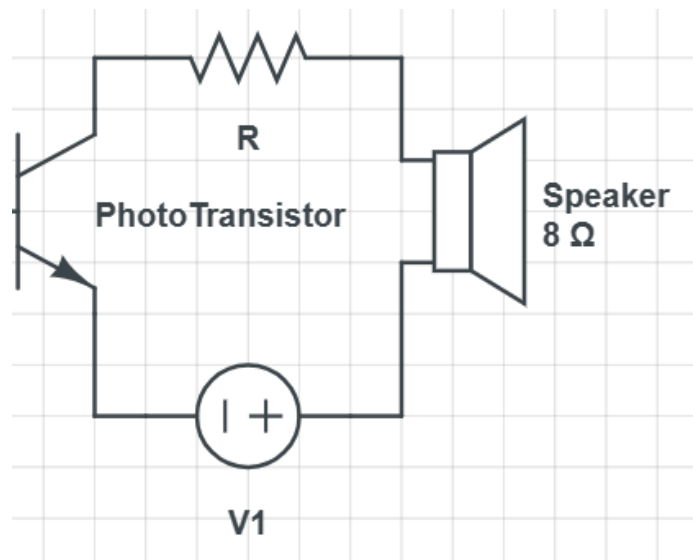


Figure 8.13. Receiver for prototype #2.

The transmitter circuit used an LED, an operational amplifier, and an auxiliary audio cable. The optical power output of an LED is directly proportional to the input current. To efficiently modulate the optical power output of an LED, the input current was controlled by a TL084CN operational amplifier operating as a voltage-to-current converter. The voltage-to-current converter is to transform the input audio voltage signal from the auxiliary audio cable from the source into an equivalent current signal. The input voltage difference from the source proved to be enough to provide a sufficient current difference to modulate the LED and transmit an analog optical signal in free space.

The receiver circuit used a phototransistor and a speaker. Even though the receiver circuit has an input voltage to power on the amplifier within the speaker and the phototransistor, the phototransistor served as a current modulator to change the load voltage across the speaker's built-in resistor. The current generated by the phototransistor depends on the incident optical power on the area of the phototransistor. To maximize this power, a lens was placed directly before the phototransistor, as shown below, to generate a spot of similar size to the phototransistor.

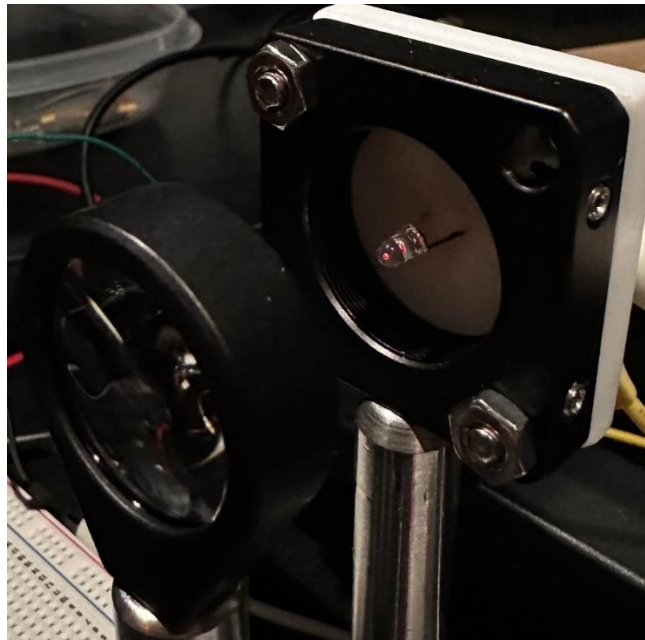


Figure 8.14. Focusing lens for prototype #2 receiver.

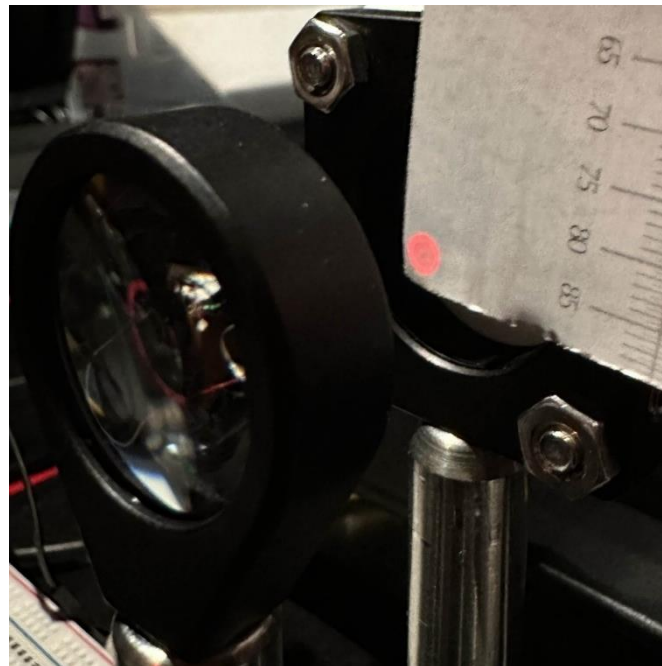


Figure 8.15. Focal spot overfilling prototype #2 receiver to maximize signal.

Both systems were put together on an optical breadboard. The transmitter and receiver circuits were taped to the breadboard. The LED transmitter and the phototransistor receiver were placed 8-inches apart. Together, the system looked like below.

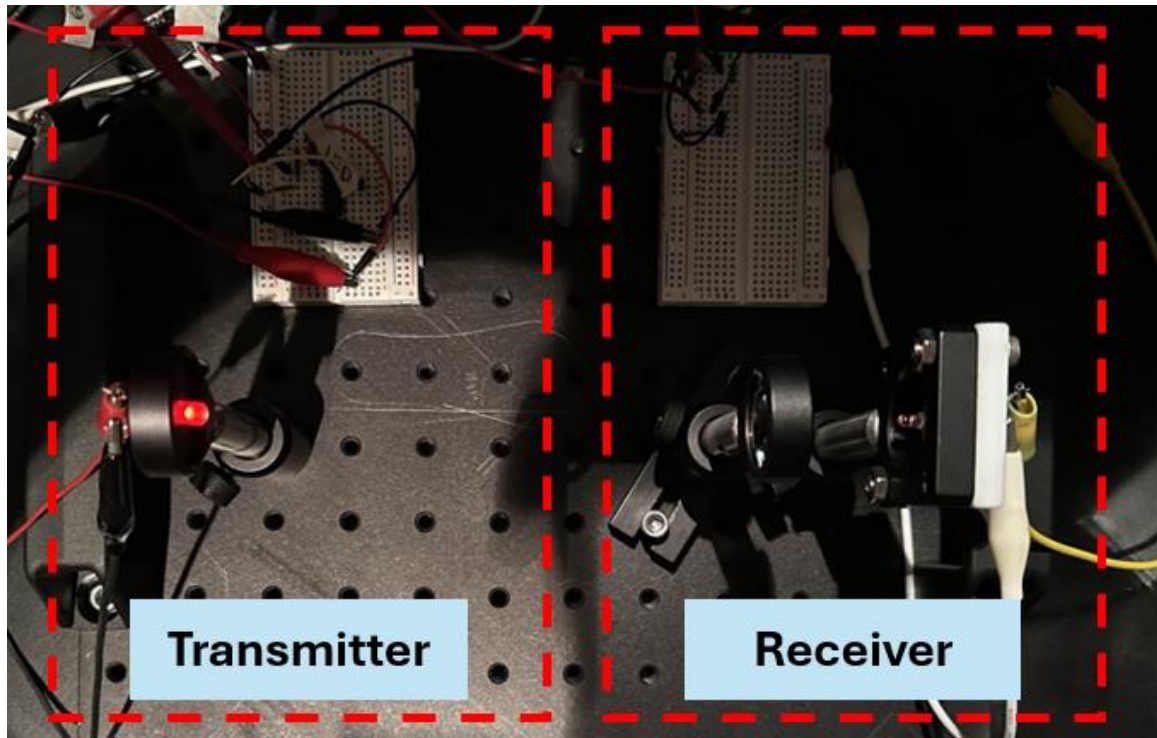


Figure 8.16. LED-based free-space optical communication prototype #2.

The third and final prototype integrated all the components mentioned in chapter 3 and chapter 6 for the optical communications sub-system. The equipment used to achieve this feat includes two media converters, two kinematic optical alignment mounts, two optical transceivers, three optical fibers, and two collimating lenses. The media converters were the brain of the communication sub-system – they were directly involved in converting the incoming electrical data from the Raspberry Pi to the correct electrical signals to modulate the optical transceivers – similar relationship exists on the back-end upon transmitting and receiving the signal. Placing the fiber-coupled kinematic mounts 0.5-meter apart, the physical distance spec was met. Then, after carefully aligning the optical system using the kinematic mounts and a power meter, a strong signal was coupled to receiving the fiber. Lastly, connecting the transmitted signal to a computer, proper video data was transmitted. To properly establish IP protocols, a feedback fiber was connected to the set, enabling the Raspberry Pi and the MCU to establish a closed connection, enabling the one-way communication of video data through free space. The set up being referred to in this discussion of prototype #3 is shown below.

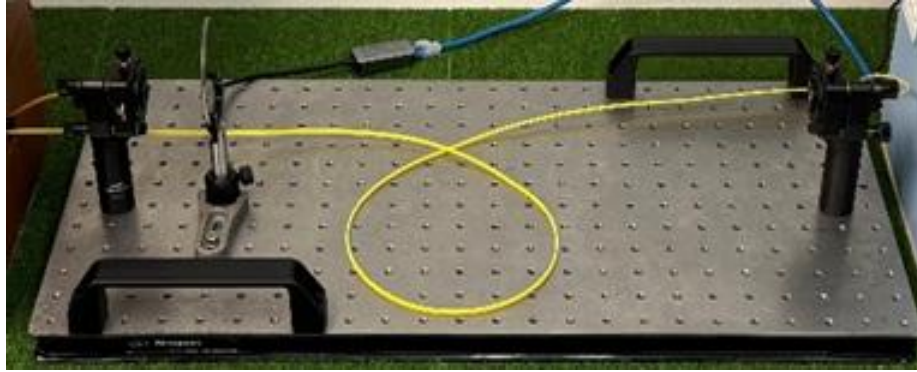


Fig. 8.17. Final FSO prototype – Prototype #3

8.3 Computer Vision Program Prototype

The prototype of the computer vision program was designed to prove the concept that our body recognition can work in a controlled environment. The prototype was designed with generic and basic structure to give an overall idea of roughly the final design. Although there would be differences in algorithm and model being utilized to do the recognition. The prototype would have three designs, one using the Haar Cascade algorithm, another using HOG and lastly YOLO which all have been discussed in Chapter 3.

The first prototype was designed with a Haar Cascade algorithm library set or database. It was to mainly show that the program can detect people in a generated image from Zemax which is shown in Figure 8.14. This would require utilizing the `haarcascade_fullbody.xml` classifier file which allows for the program to fully detect the full body of individuals. When reading the image, I used certain aspects to filter out blur basically for image correction.



Figure 8.18. Zemax Generated Image

The above figure shows the image used in the prototype for the computer vision program and with no modifications to the image. The two main modifications used happen to correct the edges and the blurriness of the image allowing for a much better result. Scaling would also be a factor in the prototype of the program as well.

The scaling of the image meaning the size of kernel or matrix of the convolution area would be adjusted to get more of an accurate detection. In the Haar Cascade prototype this would be very helpful when it came to testing the Haar Cascade algorithm in Chapter 9 section 2. The full prototype Haar Cascade code can be found in Appendix E.

The next prototype was designed with the HOG algorithm which would be used for the demo video. The prototype was run with a pre-trained model known as HOGDescriptor(). It also used the setSVMDetector() which took in the function that would be able to do a default people detector from the HOGDescriptor. It would use similar elements from Haar Cascade prototype such as defining a person by recognition of their body.

The setSVMDetector is similar to the Haar Cascade classifier though it is the classifier method that HOG utilizes. The image would be resized in order to be used for the detection in this prototype. In this HOG prototype it would use winStride and padding for more accurate detection.

The last prototype used as well as designed with YOLO algorithm and a lightweight model such as YOLOv11N and YOLOv10N. The YOLO algorithm prototype utilized two different model variations to be tested and prove the accuracy achievement. It used the Ultralytics repository or database to use both YOLO models. Both had similar setups which included model choice, boxing and labeling.

Both prototypes used had used a variable I created for simplicity in the code which was label. The label included the class identification of the object we focused on detecting and a confidence level. When it came to the confidence level it would have a boundary which was greater than .5. This allowed the program to focus on detecting objects at a confidence level higher than .5. There are many ways of gathering the results as far as boxing out objects and labeling. In the prototype a direct approach was used to handle the bounding boxes and the labeling.

YOLOv11 prototype had model set to yolov11n which is the most lightweight model. It used metrics that would handle confidence leveling, boxing and labeling. YOLOv10 prototype had model set to yolov10n which is the most lightweight model. However, the result is different from one version to the other which is discussed in Chapter 9 section 3.

As expected from the computer vision prototypes, issues did arise such as taking in the blurriness of the image which affected the structure of the prototypes in certain aspects. The final design of the computer vision program is able to handle the blurriness or haziness of video frames as a precaution. This enables the algorithm that we use to have clear and accurate detection of people by recognizing their bodies. After developing multiple

prototypes, they were all tested and evaluated to the most extreme conditions to determine which prototype, or algorithm would be utilized in our final design.

There are a few different key points to the prototypes that should be mentioned when it comes to their design. Haar Cascade prototype had to have image corrections done in order for the algorithm to do the required detection. It also required scaling factors and constraints on the size to have more detection capability. The HOG prototype had to have similar scaling and sizing constraints on the image for capability to detect people. It required image corrections as well. However, when it came to the YOLO prototypes, they both had no image corrections done and didn't require sizing or even scaling constraints for them to detect people in the given image. Although they required model choice for which model was being used for running the detection. They also required class identification and confidence factors that would be used for determining how confident the thing that is detected is a person or not a person.

The YOLOv11 prototype was adjusted to include a counter and object tracking for the amount of people in each frame. After my completed version of the YOLOv11 program I implemented it with Raspberry Pi 5 by including the usage of picam and PiCamera.

8.4 PCBs

Our PCB was designed, and placement was done within KiCad software application. A lot of adjustments are made within the process such as adjusting the width and length of wiring or switching out components and changing the MCU completely to better align with functionality. More labels are included in the revised board as well as the revised features. The PCB includes ground planes and polygon stitching so that the layers contain efficient connections. There were many mistakes and constraints in the process, precisely the differential pairs that require match trace lengths and controlled impedance. The placement of the components is important to make sure that the data lines aren't too long from the transmission to reception points. Since we're dealing with video data, we want the transmission lines to be clean and void of noise and signal interference, so those impedance calculations were of utmost importance. Capacitors are placed close to relevant devices and the regulator is placed further away to reduce noise.

Each subsystem of the overall design is separated onto individual daughter boards in order to reduce the strain on the system. This was advice given to us by our advisors with the sole purpose of being able to easily replace any individual parts in the case that something breaks or burns out. We can see each board's trace layout below.

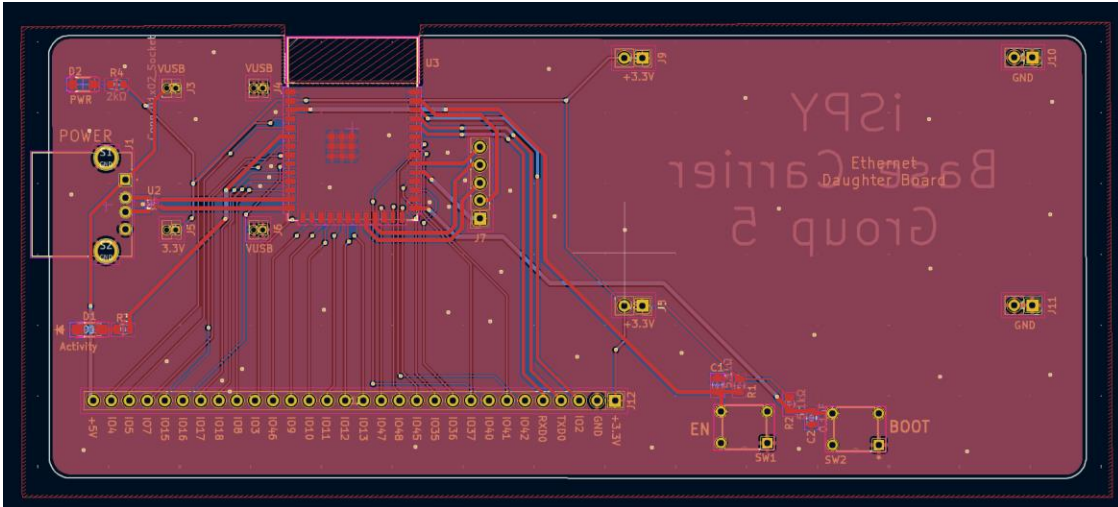


Figure 8.19. PCB #1: Base Carrier

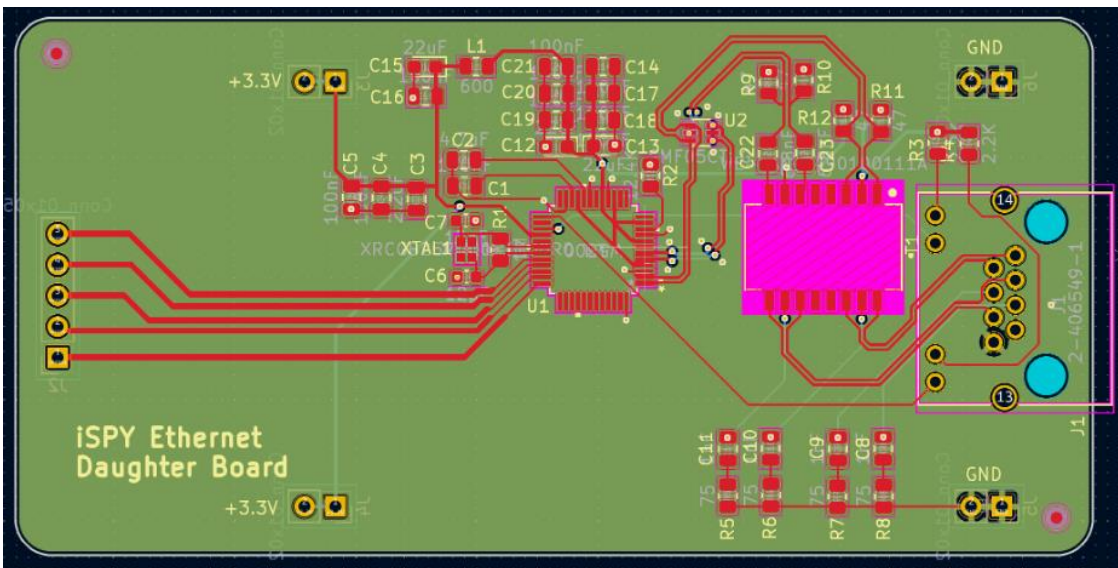


Figure 8.20. PCB #2: Ethernet Daughter Board

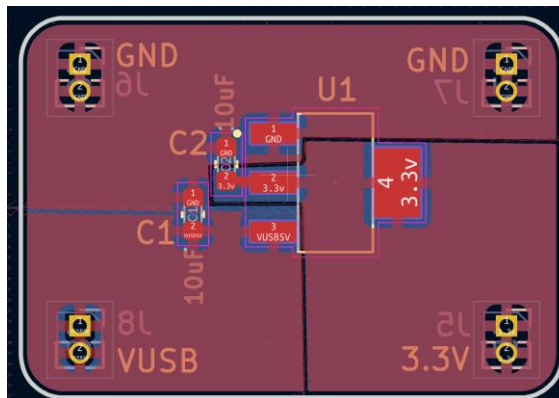


Figure 8.21. PCB #3: Voltage Regulator Daughter Board

After receiving the first revision of boards from JLCPCB, we had to do a bit of soldering to get them to complete form. This came with its own set of issues as we found out that there was a diode on the 5V power line on the base carrier blocking power from traveling the rest of the system. We also had some more major issues with the actual routing of the board itself which were all changed in the second revision of the boards. We changed some of the pin locations as well as removing that diode that was causing issues. We also changed the dimensions of the board to accommodate the standard 2.54mm pin headers instead of the 1.27mm pin headers that we had implemented earlier, which was incredibly inconvenient to work with. We also included a breakout pin header to include all of the unused GPIO pins from the MCU so that we can add anything, if necessary, which also allows us to use the external CP2102 USB-UART adapter for quick programming. With these changes made in the second revision, we have the boards constructed and ready to test and integrate.

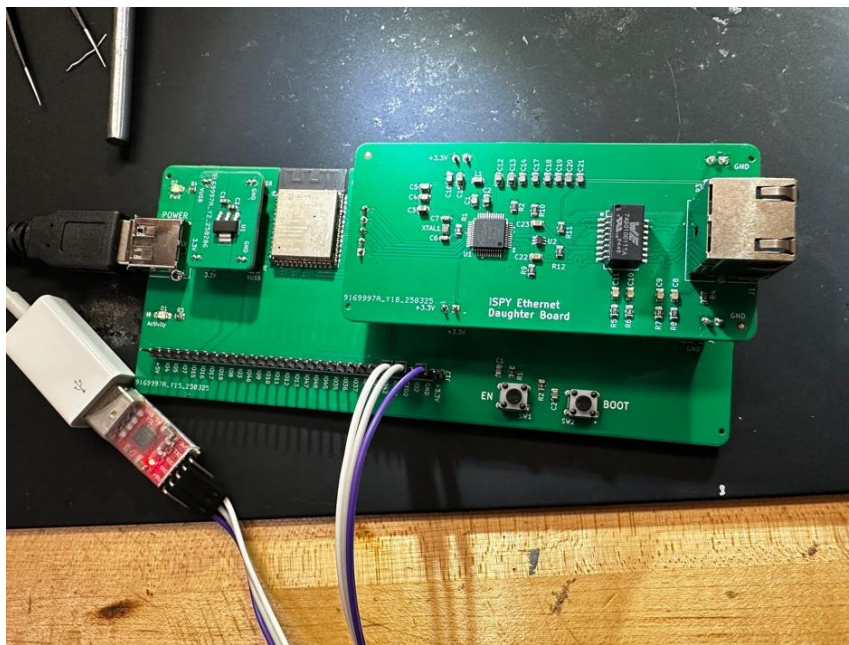


Figure 8.22. Fully Constructed PCB

The base carrier and voltage regulator boards worked as expected and were fully functional but since the ethernet board was not, we needed to order another. Although we had a final iteration of the ethernet daughter board on the way, it would not make it in time for the final demo, so we had to look for an alternative that would help us have a fully testable system in the meantime. We had found a W5500 breakout board from Amazon that we could use as a substitute for the Ethernet Daughter board to test functionality within the system.

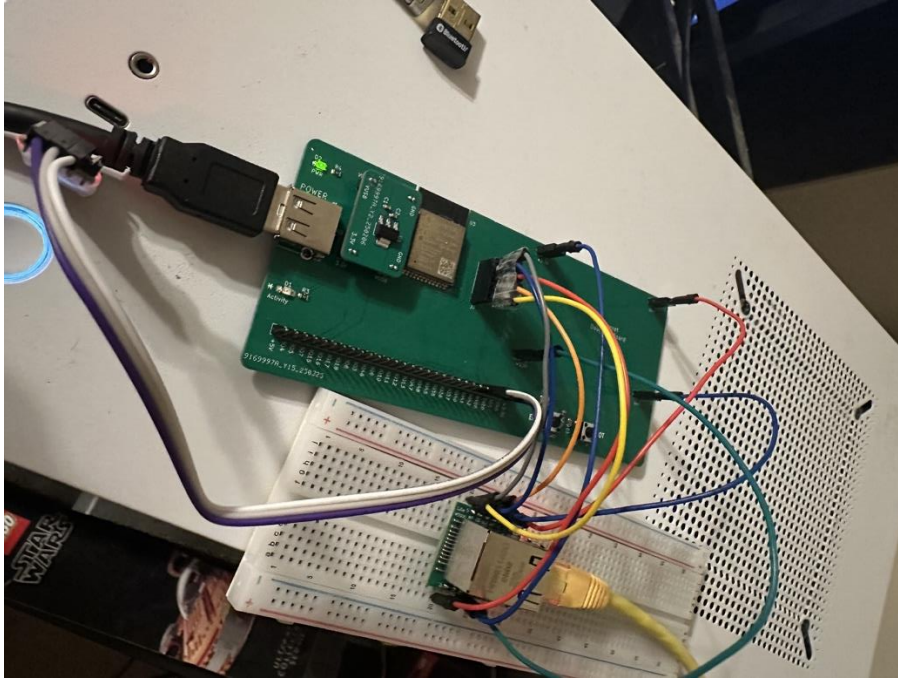


Figure 8.23. Constructed PCB Using Breakout Board

8.5 Raspberry Pi Video Compression Prototype

The Raspberry Pi 4 Model B is the board we use to handle the more intensive tasks. The Raspberry Pi board comes with just the board. The board is capable of connecting to a display via a micro-HDMI port. We needed to purchase a micro-HDMI port so that we could work with the board. However, the board comes without an OS installed. It is installed via a micro-SD card. We needed buy a micro-SD card for the board and also a micro-SD card reader so that we could upload the files onto the board. After installing the raspberry OS to the board, the board is now able to run. The board can be seen below with all labeled connections.

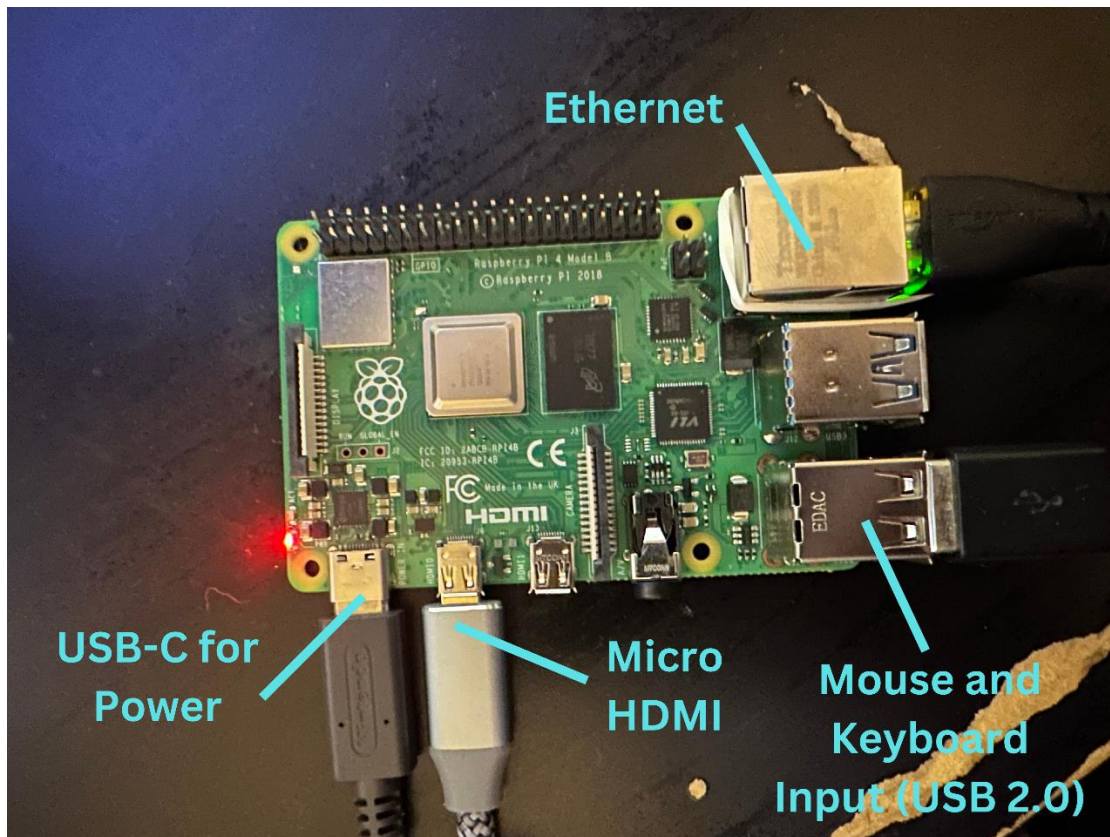


Figure 8.24. Raspberry Pi Module

Python and FFmpeg come ready on the board so the only thing to do was to write the code and test compressing and decompressing video files. A sample video was used at around 37.6 MB with 1080p and 60fps. The original file was compressed to around 7.9 MB with 720p and 30fps. The compressed file was decompressed at around 234.5 MB by upscaling to 1080p and 60fps. The reason for such a large size is the uncapped bitrate.

Name	Size	Modified
decompressed.mov	234.5 MiB	11/24/2024
output.mov	7.9 MiB	11/24/2024
Test.py	2.9 KiB	11/24/2024
Video_Demo.mov	37.6 MiB	11/24/2024

Figure 8.25. Final file sizes of compression and decompression testing

Further into the project, testing for transmission of the live video feed took place. We were having issues using the Raspberry Pi 4 Model B. We went with the Raspberry Pi 5 with 8 gigabytes of RAM so that we could have more processing power for ultralytics and FFmpeg. The Raspberry Pi 5 was set up in a similar configuration as shown in Figure 8.23. However, the ethernet was hooked up to a computer to establish a local link between the Raspberry Pi and the computer. This allowed for transmission of data between the two locally. The computer used an ethernet to USB-C adapter to establish a connection. A live

feed from the Raspberry Pi camera module was used for transmission. The system was successfully able to transmit data locally over the ethernet. This would later translate into the integration of the optical communications prototype. The image before transmission (Figure 8.25) and the image after transmission (Figure 8.26) can be found below. The blue hue in the first image is due to the camera capturing the image of the TV in poor lighting.

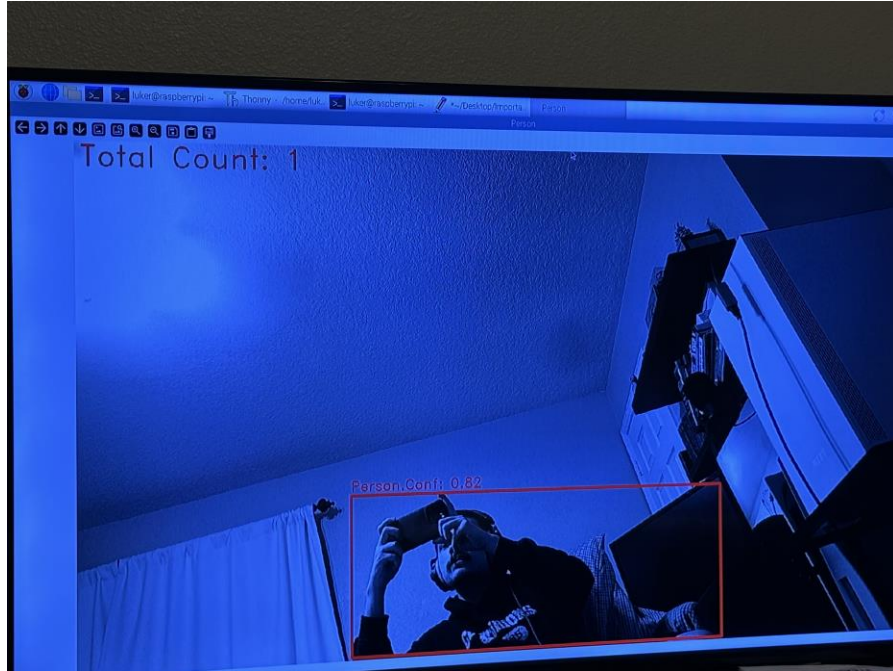


Figure 8.26. Video feed before transmission on Raspberry Pi

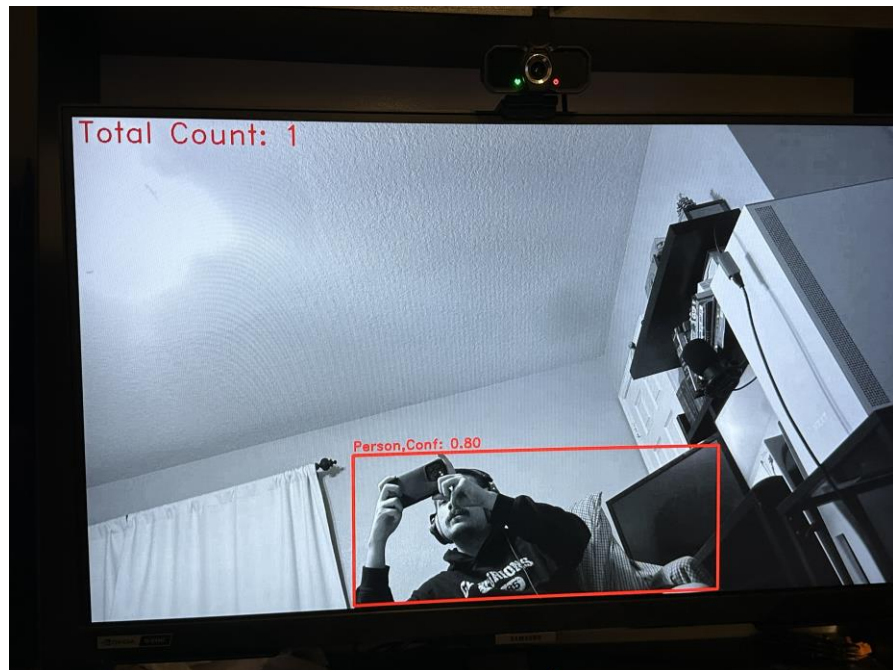


Figure 8.27. Video feed after transmission on computer

8.6 Final Setup

The final setup can be seen in the images below.

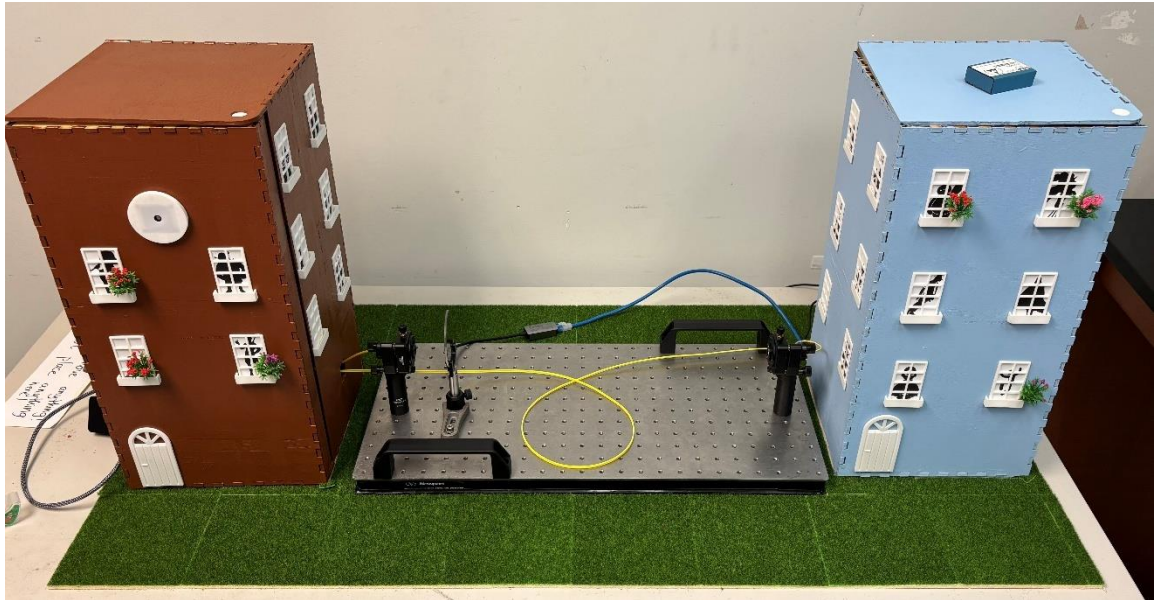


Figure 8.28. iSPY's final setup

We decided to tell a story with our final display and that is the reason why we chose to construct two buildings: the brown building was suffering with unwanted visitors, and the blue building residents wanted to help. However, they wanted to see what was happening before they had to leave their home. Therefore, a secure video surveillance system (iSPY) was implemented.

In the first building, the brown one, the image acquisition system, the raspberry pi and the transmitter were housed. We also attached a monitor to it so everyone could see how the initial video looked like.

The second building housed our PCB, receiver and another monitor – the one we could see the video after it was transmitted. Between the buildings we had our FSO system, and we were able to prove the transmission of information in free space by putting obstacles between the transmitter and receiver during our demos.

Chapter 9 – System Testing and Evaluation

In this chapter we will explore the testing done for each sub-system of iSPY and our takeaways from these trials. For a project such as this one, a lot of trial and error takes place, and we made sure to record this process as it shows the process that happened until we reached our final results. Even though this was the most timing-consuming part of the project so far, we gained more insight into the system and experience.

9.1 Image Acquisition System Simulation

The first part of the testing began with the calculations found in Chapter 6, Section 6.1. We first found the required focal length of our system and that was the starting point for everything else. Since we are trying to accomplish a lens system that has a wide field of view, the effective focal length is the most important parameter.

As mentioned in Chapter 3, Section 1.4, the plan was to follow the frameworks already used in the industry. However, as you will see this was not very successful as our limitations were many. Therefore, we had to move back to the basics, from paraxial approximation to adding one lens at a time and trying different combinations. More than 200 iterations took place, and 40 of them were recorded. In this section we will explore the most important ones, the lessons learned for each of them, the optimization process, and how we reached our final design.

The first round of trials: Following the instructions from the book “Designing Optics Using Zemax” the first round of trials took place. As mentioned in Chapter 3, Section 3.1.4, the simulations that took place following their steps did work successfully; however, the design relies on custom-made lenses which go beyond the scope of this project due to our time and financial constraints.

The second round of trials: For the second round of trials, I went back to the paraxial world. This term is used in optics when we are talking about approximations made with thin lenses, in which we treat light as rays. This approach is highly used when designing a lens system, since it is simplified, we can focus on basic parameters, such as focal length, before anything else. This approach could also be seen as a prototype of the lens system since it is a rough design of the final product.

The following figure shows how the paraxial approximation was done. The lenses are displayed as thin lines. The effective focal of this system was 3.16mm, the focal length of lens 1 is 6mm while the focal length for lens 2 is of 5mm. This combination was successful at obtaining the effective focal length within the range we needed; therefore, I proceeded to search for off-the-shelf lenses that would work similarly. The F-number for this system 6.0, and the entrance pupil was 0.53mm – this trend of large F-number, and consequently, a small entrance pupil is consistent throughout the simulations. Since the iSPY is designed to be placed in a well-lit location a small entrance pupil is needed.

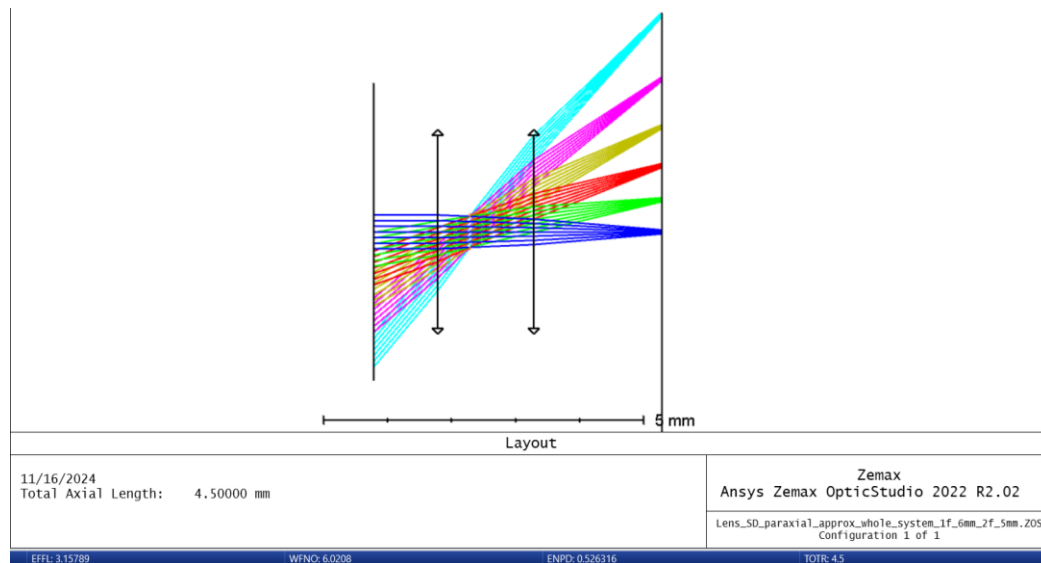


Figure 9.1. Lens design – Paraxial approximation by Authors (Zemax output).

As you can notice, originally the aperture stop was placed in between the lenses, there was nothing limiting the light into the first lens. We later saw this was a big issue.

Third round of trials: for the next round of trials, I downloaded the Zemax files with the lenses' information and I ran simulations. However, when searching lenses' manufacturers websites, it was not possible to find a lens with a diameter greater than 7.4mm (to cover the sensor entirely) and focal length as short as the ones from the second round of trials (5 and 6mm). Therefore, I had to work with lenses with longer focal lengths, increasing the effective focal length.

One of the hardest parts was figuring out how many lenses I needed. In the process of researching for this project I learned that security cameras you can buy off the shelf usually have 4 to 5 lenses, so this was my starting point.

As you can see in Figure 9.2, the results were not great. Even though images are being formed they were severely aberrated, and the focal length was a lot higher than we wanted. From the first to the second trial in the figure, you can see how we added fields (shown as different colored rays). Since the goal is to image a wide field of view, we needed to make sure that light ("rays") from all the directions were reaching the last surface – the position we placed our sensor.

This round of trials was focused on understanding how different lenses would work together, it was conducted with flexibility, emphasizing experimentation over strict parameters. As one can see, several types of lenses were simulated, the position of the aperture stop changed, etc. The biggest takeaway from this set of simulations was that increasing the number of lenses does not necessarily make the imaging system better.

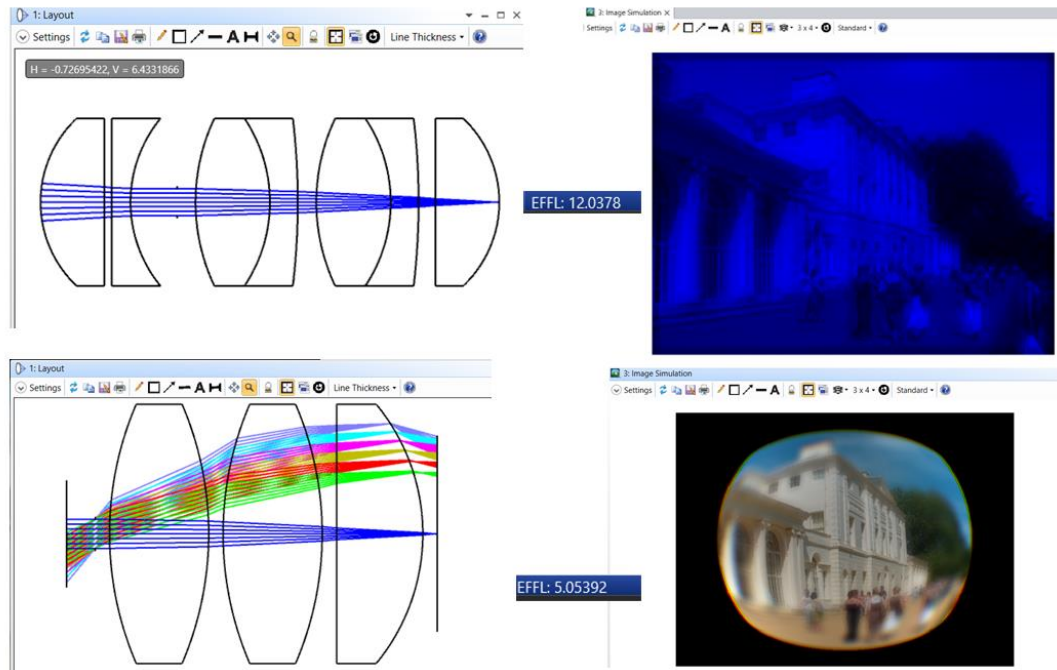


Figure 9.2. Simulations from the third round of trials (Zemax output).

Fourth round of trials: now, having a better understanding of how the lenses worked together and using the last trial in figure 9.2 as reference, I began to focus on decreasing the focal length – our most important parameter.

First, I tested something that resembles an inverted telephoto lens system, negative lens (or lenses) followed by positive lenses. Even though the quality of the image simulated increased, the focal length also increased, and this is not ideal. An issue happened when using a meniscus lens, there was not a great variety of sizes (or focal length) for this type of lens; therefore, the simulation failed. This trial is displayed in Figure 9.3.

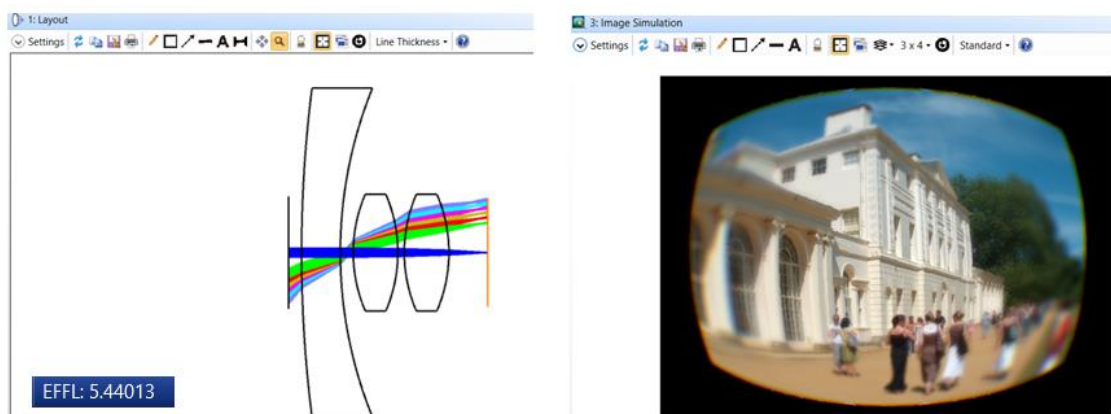


Figure 9.3. Negative meniscus followed by positive lenses (Zemax output).

A different combination of lenses was tested to obtain the inverted telephoto system configuration. This time, a double concave was placed in front of two biconvex and convex plano lenses.

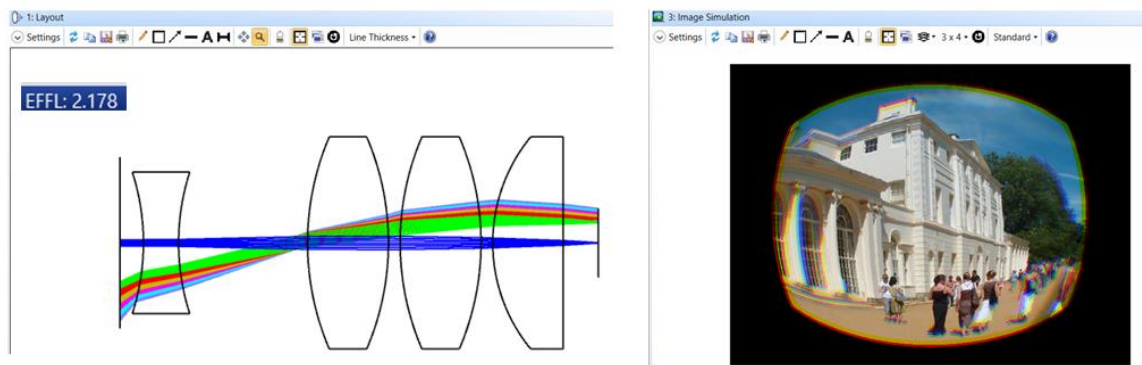


Figure 9.4. Second attempt on inverted telephoto lens system (Zemax output).

The focal length was successfully reduced; however, this design suffered from chromatic aberrations. Since the light is going through so many surfaces, this is something expected. The image simulation shows how color, especially on the edges, is imaged. Unfortunately, along with the chromatic aberration we also had barrel distortion. Keeping this system would not be in our best interest since both types of aberrations are very apparent. We understand that it would be impossible to design a perfect system, but at the same time we want the imperfections as much as possible.

Another trial to reduce the focal length was then tested, this one involved the use of Fresnel lenses. Fresnel lenses are great at collimation; however, during my research I also found that sometimes they are used to decrease the focal length of a system. The trial was successful at reducing the focal length; however, it also reduced the quality of image. Fresnel lenses are usually not used for imaging for such a reason. I also tried to insert the lens in between the other two lenses but in that scenario the detector would have to be right after the last lens – making it something extremely difficult to accomplish when building the physical mount for the system.

To reduce the chromatic aberration, I simulated an achromatic lens in the system. Unfortunately, this is also the case where we do not have control over its focal length, and we can only test (and purchase) the ones on the shelf. Even though the trial went well, the focal length increased by a large amount and the image resolution decreased.

In the next image the MTF graph shows how this lens system is not performing well around the edges of our frame. The dark blue line indicates the 0° field, which means that, on axis, our system is performing close to the diffraction limit (black line). However, the graph also shows the drop in performance at different angles. In other words, it is not resolving the image at higher spatial frequencies.

Introducing an achromatic lens in our system decreased chromatic aberration as expected, but it also introduced a problem – the resolution of our image. We knew tradeoffs would take place in this system, we understood there would be compromised in some parameters, and that is the reason why so many trials took place. To achieve the best possible design with the given constraints, we had to analyze the outcome of several configurations.



Figure 9.5. Attempt on inverted telephoto lens system with an achromatic lens (Zemax output).

Fifth round of trials: During this last round of trials, we could see that we had reached the limit with regular (not aspheric) lenses. We tried to add more lenses to decrease the focal length and chromatic aberration. However, more surfaces did not improve our design – in fact, it deteriorated our images and made it more expensive.

And that is when we decided to use aspheric lens. We understand that they are more expensive than non-aspheric lenses; however, the price difference between one aspheric and three to four (what we had in our previous designs) non-aspheric lenses was not too much. By using aspheric lens, we reduced the size of our system, and decreased aberrations especially chromatic aberration and barrel distortion. With this new design, we were able to achieve equivalent and even better results than the ones previously obtained.



Figure 9.6. Final lens design for iSPY (Zemax output).

Figure 9.6 displays our final lens design. It is composed of one aspheric lens, and one convex-plano lens. The image simulation shows the decrease in barrel distortion and chromatic aberration. The effective focal length is 3.66mm.

The aperture stop of this design was moved to the front of the first lens, we wanted to control light entering in the system from the very beginning. This stop is also our entrance pupil – and the measured entrance pupil diameter is of 0.46mm. Since we have a large F-number (of 8) this makes sense.

A concern was raised if we would get enough light into the system, so we tried to decrease the F-number (increase the entrance pupil diameter).

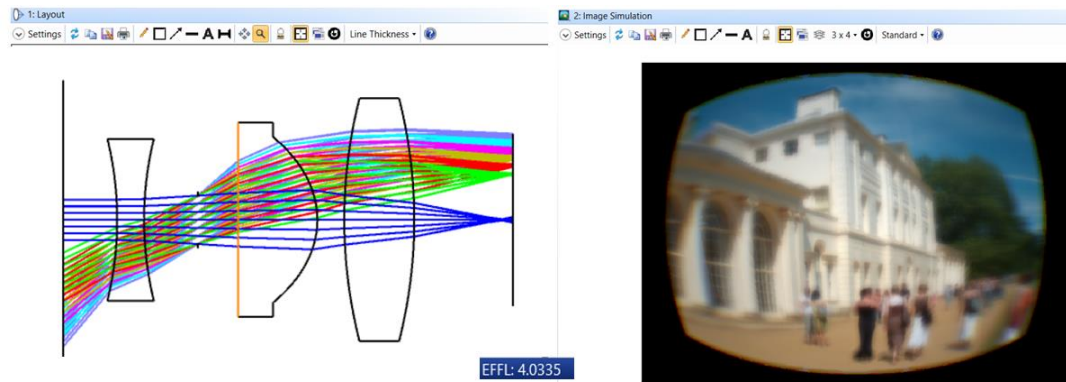


Figure 9.7. Larger entrance pupil diameter (Zemax output).

Figure 9.7 shows this attempt at increasing the entrance pupil diameter. And as expected, as we increase the aperture, we let more light in, and that causes more aberration.

Therefore, we kept the design from Figure 9.6 and to test that enough light makes into our sensor we carried out the experiment found in Chapter 6, Section 6.1.

This final optical design was kept, and no alterations were made during Senior Design II. In chapter 8, you can see the structure made to mount the optics.

9.2 Optical Communication Testing

The optical communication prototype was built over the span of several days. Testing was conducted throughout the assembly process and in the final stages of prototyping. This form of procedural testing enabled changes to be made on demand without wasting resources and time. Ultimately, the optical communication prototype was also showcased to faculty and peers from CREOL, the College of Optics and Photonics, at the University of Central Florida giving the team an opportunity to get valuable feedback and improve the prototype design.

As mentioned in section 8.2, the prototype used as proof of concept for the critical optical communications component was composed of two isolated circuits for the optical transmitter and optical receiver respectively, an analog audio input signal, an analog output audio signal, speakers, and an imaging lens. The prototype showcased is shown on figure 8.13.

The prototype prepared using the diagrams presented in section 8.2 yielded the above – two isolated circuits connect only via light. The transmitter and receiver circuitry were prototyped on separate breadboards and connected to separate power supplies. This limited and eliminated any possible electrical interference that could have yielded a false positive result from the prototype. Originally, the circuits were connected using the same power supply; however, the circuits were powered by different DC output ports. Even though the output ports were different, a faint sound could be heard coming out of the speakers. This sound was attributed to electrical interference and the issue was resolved.

The transmitter of choice was a red LED (~633 nm wavelength) with a forward bias voltage of 1.7 V measured using a digital multimeter. Since LED devices are not voltage modulated, the current signal from the TL084CN quad operational amplifier was used. This current signal was derived by configuring the operational amplifier as a voltage-to-current converter, as discussed in section 8.2. When the inverting input to the amplifier is connected to an analog audio input, the varying analog voltage is converted to an equally varying current signal for the LED. Interestingly and unknowingly, the LEDs used for the prototype had reversed anode and cathode leads. This led to the burnout of two LEDs throughout testing. Regardless, the varying current signal was observed using a digital multimeter by varying input voltage to constant voltage inputs.

The receiver made use of a very simple circuit as demonstrated in section 8.2. The prototype features a phototransistor with peak responsivity for 1100 nm infrared light. Initially the phototransistor was not available for prototyping and a photoresistor was used in its place. Since the photoresistor is conductive when exposed to light, it had the potential to modulate the current on the receiver circuitry by varying the voltage drop across itself. The initial photoresistor circuit was as simple as the one shown in section 8.2. Then it progressed to a voltage-to-current converter circuit. Neither proved good. The phototransistor proved to be a much more effective receiver. As a photoresistor is exposed to light, it generates a current that varies the load voltage across the built-in resistance of the speaker, modulating the sound output. Even though the photoresistor responsivity is not at its peak at near 633 nm wavelengths, it generates enough current to drive the circuit with and without a voltage source.

The sound quality produced by the speakers connected to the prototype yielded good bass sounds. Interestingly, treble and mid sounds typical of any sound system were almost indistinguishable from the noise amplified by the speakers. This can be attributed to the limited LED optical power modulation depth enabled by the transmitter circuit. Since the transmitter circuit was making an equivalent amplification of current based on the low input voltage from the source music, then the LED optical power was limited to small current variations. This can be improved by amplifying the input voltage signal from the music source before the voltage-to-current converter.

Additionally, the sound quality could have been degraded by the electrical connections used throughout either circuit. Degraded wires introduce noise. Dirty wire connection leads introduce noise. Faulty speaker circuitry introduces noise. Regardless, noise was amplified and present regardless of test conditions.

The 35 mm focal length lens used for the prototype was used to increase the radiance on the phototransistor. Initially, the phototransistor was placed immediately after the LED to capture as much light as possible, as shown below.

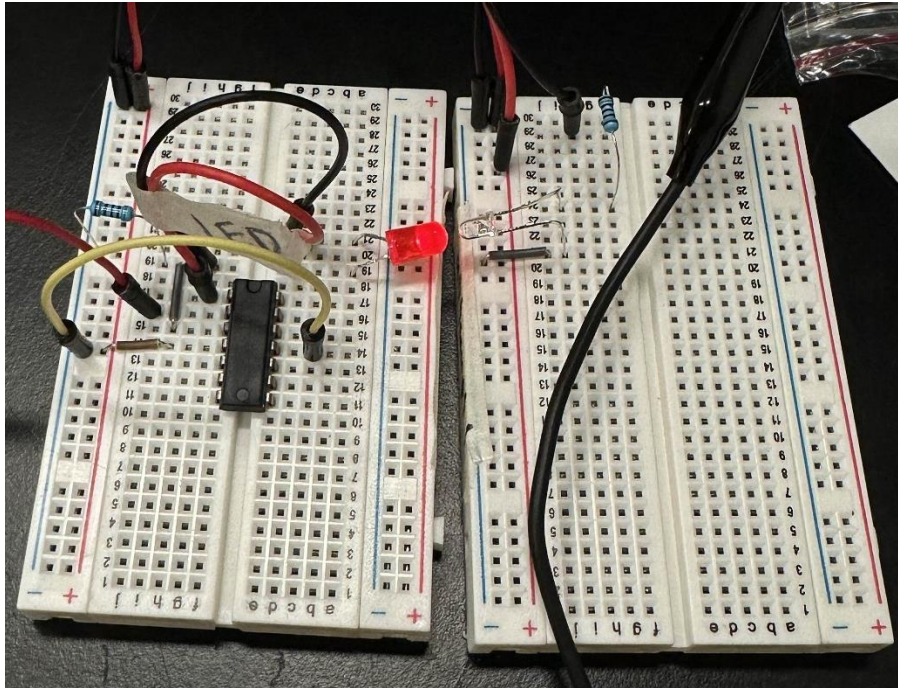


Figure 9.8. First attempt at an LED-based free-space optical communication prototype.

Even though audio signals were transmitted and received effectively in this configuration, the generated audio outputs from the speaker were weak. The bass sounds were weak compared to when using the lens. The lens focal length was chosen with regards to the 8" workspace and to have significant distance between the LED and focal point of the lens. The latter was necessary because the LED signal had to be imaged on the photoresistor to maximize the incident radiance. Overall, the prototype was successful at transmitting and reproducing audio signals.

In preparation of prototype #3 discussed in section 8.2, optical communication was put to the test in two different ways: fiber testing and free space testing – both differing by the transmission medium. Fiber testing was an all-in optical fiber evaluation that tested the optical transceiver at their optimal efficiency using SMFs. Since optical transceivers are made to work well with optical fibers, an all-in fiber test enabled proper verification that the transceivers functioned as intended. An example all-in fiber testing is shown below.

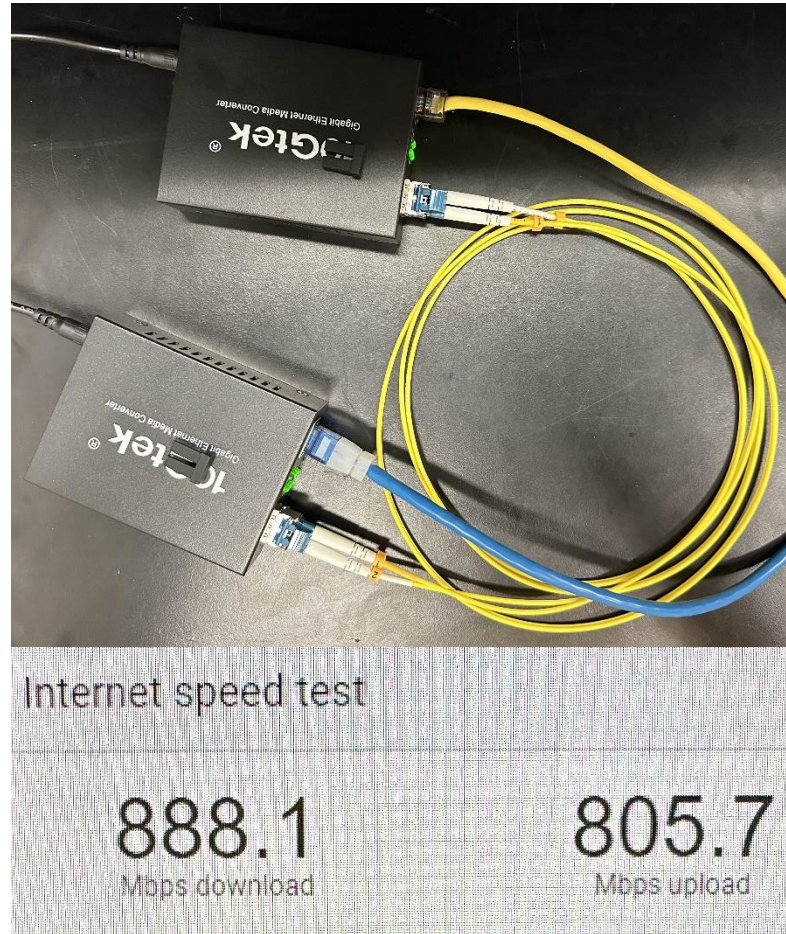


Figure 9.9. First all-in fiber of testing of optical transceivers using SMF-28e fiber

The fiber testing was successful in proving the transceivers worked to meet the transmission speed spec, yielding effective computer connectivity to a personal laptop. Besides, the optical communication sub-system is a mostly optical wireless communication system. Hence, the next and final test is a free-space test. The free space evaluation removed the duplex optical fiber from the fiber evaluation, replacing it with a simplex fiber instead. These simplex fibers were test connected to the kinematic mounts to enable optical alignment by hand. Placing the transmitter and receiver mount, first, a few inches apart, then 0.5-meter apart, the test was conducted. First tests were to couple light onto the distal, receiving fiber. This was not easy feat as it took a while to couple light into an optical fiber the farther the lenses moved from each other. Besides, a free space link was established with very low loss, around the magnitude discussed in chapter 6 – near-zero loss. With a great free-space optical communication link, it was time to test data transmission in free-space. After integrating all the necessary sub-systems together, video packets were transmitted, received, and displayed on a separate laptop. These results proved the success of the project. For reference, the image below details the testing location for the first few video transmissions using free-space optics.

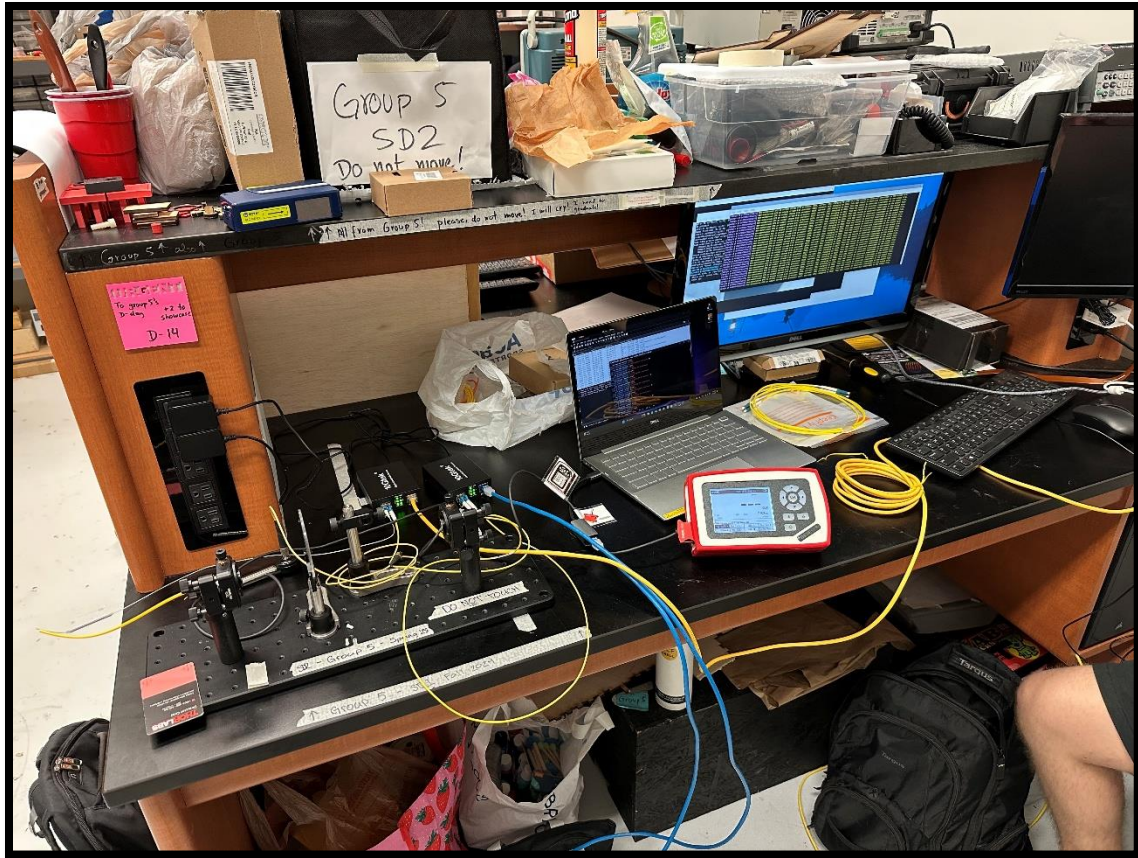


Figure 9.10. Test bench for the transmission of video packets via free-space optics

With these tests ultimately yielding positive results, the group agreed to continue integrating all other sub-systems of the project and transmitting near-real-time video footage over the free-space optical link.

9.3 Computer Vision Program Simulations

In the testing of the computer vision program a total of four simulations were tested. When it came to the testing, I utilized the setup mentioned in Chapter 7 section 3 with minor changes depending on the algorithm used. Since we are shooting for accuracy and efficiency when it comes to our detection of people in video frames, I wanted to focus the test on many different algorithm variations. Also, I decided to work with different classifiers and models to see their functionality as well as how they could handle the recognition in a rough image that had blurred edges. Although some individuals show up with some blur and at different distances, this was a known factor for the design of iSPY since people were (and will be in future usage) in view of the camera module at varying distances from the camera module.

All the tests used the same Figure 8.14 to keep the input consistent among and across the tests that were run. The multiple algorithms that were incorporated in these simulations of our computer vision system were deeply discussed and detailed in previous chapters. The

first simulation was for the Haar Cascade algorithm which was a baseline for the other simulations that were run. Next simulation that was tested was for HOG algorithm which was a bit more advanced in detecting people than the Haar Cascade simulation. The last simulation that was tested was for the YOLO algorithm which I used two different models for seeing results of detection between the two.

All simulations concluded that we are able to detect people. Although the accuracy of the simulations would vary in how accurate they detect. In the results bounding boxes and labeling were utilized to show the detection. I would test until the results gave at least one or no outliers. Every result shows the progression in the testing process for clarity on how the detection result would change with every simulated program.

9.3.1 Haar Cascade Simulation

The first program that was simulated was a very basic computer vision algorithm. This program was used as a baseline for the other programs that would be simulated. In the simulation of the Haar Cascade algorithm I would set it up by creating a variable (path) that took in the xml file in the Haar Cascade database which pertains to detecting a full body of a person. I created another variable that would be for the cascade classifier which took in the variable mentioned earlier.

After I created another variable that would read in the image being used for all the testing. I defined a function that contained the method for scaling as well as a for loop that generated for the bounding boxes and labeling. During the test of this program at first, I did not do any corrections to the image. I also didn't adjust the scaling and left it with general scaling like minimum neighbors be set to 2. The other thing I would add is a flag in the scaling method and a minimum size for the scaling. I realized that the image was blurry and that would affect the detection which is why I added Gaussian blur. Also, Haar Cascade uses the gray scale of the image, so I added the conversion from BGR to gray scale which was an input in the equalize histogram and Gaussian blur.

Once I added image corrections and adjusted the scaling metrics from 2 to 1 for minimum neighbors, minimum size from (10,10) to (20,20) and added the flags to cascade scale image. It helped with the detection in the input image.

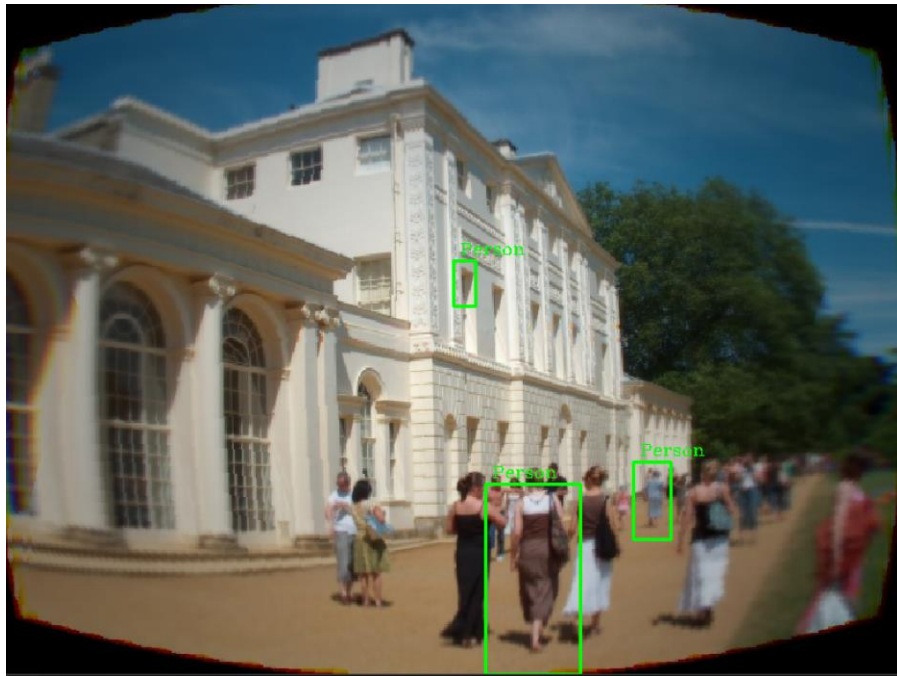


Figure 9.11. Resulting Image from Haar Cascade algorithm program

In Figure 9.9 the result shows detection for a person in the given image using the Haar Cascade algorithm and the full body xml file. The figure also shows that there is one outlier that exists. This result is after the adjustments were made to the scaling and blurriness of the image. However, the image is still blurry on the edges and corners leaving those people that are close to them undetected.

9.3.2 HOG Simulation

The second program that was simulated was a bit more advanced computer vision algorithm. This program built off the previous one. In the simulation of the HOG algorithm, I would set up a variable that would represent the HOG descriptor. I utilized the SVM classifier that HOG uses and made a variable for reading the image similar to the first program tested. I had to include resizing in order for HOG to work with the image. I included Gaussian blur though I did this at the beginning and not after I tested it.

After making sure that the image correction was included, I would implement the scaling method. Although this time I had to use window stride and padding for scanning the image in the detection process. The rest of the program would be the same as the prior program. I adjusted the resize a few times to finally end up using (650,500) which represents width and height. I also changed the window stride a couple times to end up with (5,5) and I changed the padding to settle on (8,8).

No adjustments were made to the Gaussian blur and the bounding boxes as well as the labeling. I found out what the window stride and padding does exactly in the program. Both the window stride and padding help with the accuracy when detecting in the image by scanning it. The improvements made using HOG could be seen in the results.

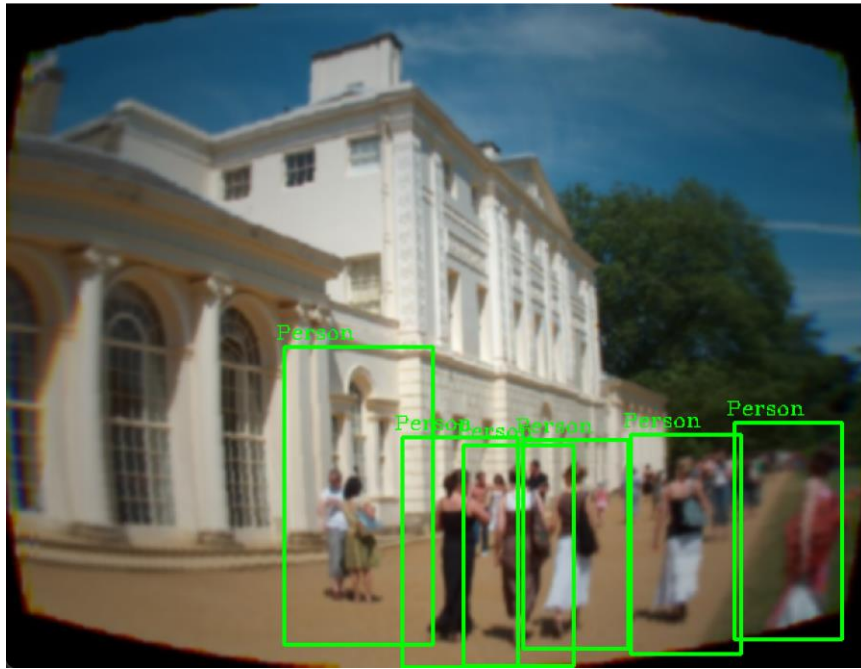


Figure 9.12. Resulting Image from HOG algorithm program

In Figure 9.10 the result shows the detection of people. The figure shows that there is no outlier and that more people are detected. The edges and corners are still blurred although there was detection of a person in the lower right corner of the image this could be from the way the image was scanned or scaled. However, I know that the result has overlapping bounding boxes which could be adjusted for a cleaner display of detection.

9.3.3 YOLO Simulation

The last program that was simulated was even more advanced than the others. I would run two models to see the difference in the results. The first one I worked on was YOLOv11 with yolo11n model. I implemented importing YOLO from Ultralytics then I would read in the image like the others. The other similarity was the bounding boxes and the labeling although they were structured differently by using two pairs of x and y. So, the bounding boxes used x1, x2 and y1, y2 for boxing out the person detected in the image.

I included a class identification for the labeling of the person and confidence score. I also added confidence score to be displayed for showing how confident the algorithm is in its detection of a person or that the object is a person. I included a variable that represents the model of YOLOv11 that I utilized in the program.

There are additions that could be added such as scaling and adjustments to the bounding boxes to image corrections. All of these could improve the results for both models of YOLO used in the simulation of YOLO algorithm.

I did not make any additions due to the fact that I wanted to see the raw results from the two models of YOLO algorithm when simulated or tested. The second one I worked on was YOLOv10 with yolov10n model. The structure for the YOLOv10 program would be like that of YOLOv11. I would not add additions to the YOLOv10 program leaving it the same as YOLOv11 so I could get the raw result.

Both models were more advanced in the structure to process the image. The advancement that YOLO models have is very evident by testing both models with extreme conditioning. I changed the structure for the YOLOv11 N model when generating the bounding boxes and labeling. Even with the changes made the result showed no or little difference in the detection of a person.



Figure 9.13. Resulting Image from YOLOv11 algorithm

The figure above shows the results for YOLOv11 utilizing the model yolo11n which is the lowest lightweight model. The result shows no overlapping bounding boxes making the display of the detection cleaner and more coherent to those viewing them. I was very impressed by the fact that the result shows two bounding boxes around the two individuals on the left side of the image. In Figure 9.10 the two people were in a single box grouped together though in this one they are separated by their own boxes. I was also impressed by the detection itself since it was able to detect near and far as well as blurry to non-blurry. The confidence level even changed depending on how clear the person was in the image.



Figure 9.14. Resulting Image from YOLOv10 algorithm

In the figure above it shows the results for YOLOv10 with yolov10n which again is the lowest lightweight model for YOLOv10. The result shows a little overlapping between the two people detected in the middle although that can be fixed with additional functionality used in the code. Also, it only recognized one person over on the left in the image when clearly, we can see two are standing there right next to each other. It didn't detect those people that are far away either, making it seem as if they can't be detected because they are too blurry. The program did do a decent amount of detection even without detecting every single person.

After testing and evaluating the results I compared them together to make a final decision on which one of the algorithms would be viable for our system. I thoroughly tested each to see their handle load and their latency factors on the detection process. If their latency was over thirty seconds to a minute that would be an issue for our system since we want to detect immediately once someone comes into the view of our camera module. Another issue would be the number of people the algorithm was able to detect and box out. Even though the YOLO algorithms included the confidence score that was not a deal breaker.

The confidence score was a bonus which was a stretch goal mentioned in Chapter 2 section 3 under stretch goals. It was implemented as a test of its own which ended up working in both YOLO algorithm simulations. Overlapping bounding boxes was a consideration as well even though it is a fixable issue and problem to have in computer vision detection. One of the most common problems is overlapping boxes which can be solved by utilizing non-maximum suppression (NMS) in the program. This only keeps the most confident bounding box when there are overlapping boxes.

When considering the use of Haar Cascade as our algorithm to utilize for our system I looked for the latency and the amount of detection made on the image. Overall, the latency

of Haar Cascade didn't factor in since the program instantly gave results. The detection of Haar Cascade wasn't that great though it was good since it was able to detect two people and have one outlier. This clarified that Haar Cascade wouldn't be able to handle the real-time detection we are aiming for in our system.

The HOG algorithm was an improvement in the detection of people. Its latency again wasn't really a factor since it was able to give the result a bit after the program ran. The detection in the resulting image showed no outliers and more people being detected by the program which is indicated by the bounding boxes. Although it is not a clear and clean detection. Also, for real-time body recognition the algorithm wouldn't be able to handle recognizing the person in real-time. It would struggle cause of posture, and it would need fully visible bodies in order to recognize them. In our system since we aim for real-time recognition or detection, we need an algorithm that could handle many different frames or views such as blurry or clear bodies moving in as well as out of the view of our camera module. This meant that HOG would not be a suitable fit as our computer vision algorithm to utilize in our system for iSPY.

YOLO algorithms seemed to be the most advanced and the most improved in detection. When I tested the two different YOLO algorithm models it seemed as though the latency was a factor in displaying the results after the program ran. YOLOv10 was able to recognize more accurately than HOG though it didn't have a lot of people recognized. I utilized the model yolov10n which is a lightweight model. It seemed that the model selection caused a longer latency to gather the resulting image. In iSPY we want to recognize all the bodies of people that are either in or out of the FOV of the camera module. This means that YOLOv10 with model yolov10n could be a possible choice for our system.

I would keep YOLOv10 as the choice for iSPY until I tested YOLOv11. YOLOv11 is the most current algorithm out so I decided to put it to the test. YOLOv11 was able to recognize almost every person in the image that was either close or far and even visible to non-visible. It accurately boxed out every individual person recognized in the image. I used the lightweight model yolo11n which once again caused latency to gather the result. Though the result was very impressive and more accurate in detection this would make it the algorithm to use for iSPY.

The results were very rough in recognition and the processing of the image. It was simply an overall test to see which simulated algorithm works best with our structure for our system. It showed the pros and cons of each algorithm that was simulated. The test also gave clarity on which algorithms are more of a suitable match for the recognition and would possibly be able to handle real-time recognition. The iSPY system had certain focal points in accuracy, efficiency and less latency for body recognition. Since the computer vision program was geared towards achieving all these focal points I wanted to push the simulations to their full potential. I would adjust as needed to get more accuracy or increase the efficiency of the recognition. I also wanted to determine which algorithms had more latency for body recognition.

9.4 Computer Vision 1st Iteration

After the initial test I decided to implement the counter that would count how many people were in each frame. When I included the counter the first time I had it based on the confidence level which is not the accurate count since it is not based off of people. However, it was based off of how confident the model was that the object was a person. If the confidence score was higher, such as .75 or .85, which is equivalent to 75% or 85% confidence then the counter would reflect that by incrementing by 1.

I overcame this obstacle by including object tracking which was mentioned in Chapter 7 Section 3 Sub-Section 5. The inclusion of object tracking involved the use of the track method, checking track IDs for each bounding box and plotting the tracks. Although this would work, another issue arose, which was that it was creating bounding boxes around objects that were not a person meaning it was including bottles, chairs and other objects. I dealt with this issue by model class selection meaning I selected the index that corresponded to the person class for the model. Also included a model property called name which would be set to person. If the name was person, then it would only box out people in the frame as well as count them since the only objects boxed were people this is shown in Appendix E.

9.5 Computer Vision 2nd Iteration

I modified my first iteration by including an aesthetic appeal to make it similar to a security camera or system. The modifications included changing the color scale of the frames to gray scale and converting gray scale to BGR (Blue Green Red) using the gray scaled frames. The reason for utilizing the gray scale frames and doing the conversion on those gray scaled frames is to create an overlay. The overlay is to display the labels and bounding boxes in red coloration which is shown in Appendix E.

9.6 Computer Vision Raspberry Pi 5 Code

The only way the program would work with Raspberry Pi 5 was to implement the code to work with a PiCamera module which could be connected to Raspberry Pi. I included the necessary libraries to use the PiCamera which involved picam2. The way to view live video feed from the camera module which is utilized in our Image Acquisition System had to be through the setup of the camera module such as size and formatting. Also, I added the array that captured the frames one by one which is shown in Appendix E.

9.7 Computer Vision AI Kit Code

When it came to major optimizations the use of a Hailo AI Kit would be attached to the Raspberry Pi 5. The way to work with the AI Kit was by setting up the kit which dealt mainly with running certain commands in the Linux terminal on the Pi itself. After the setup process was finished, I had to optimize the program to use the Hailo AI Kit to boost up the computer visions frames per second (fps) to be real-time which is 30 fps. I would work with DeGirum PySDK which works with the Hailo AI Kit although it is recommended to use HailoSDK. The reason HailoSDK wasn't used is due to the fact that it is more complex to utilize, and it has a pipeline structure.

I installed DeGirum onto the Raspberry Pi and imported the library that would be able to load the required as well as necessary information to load the computer vision model. The model contained a model path, inference host address and a zoo URL. The model path was a folder inside the directory (zoo URL) and the inference host address was whether the model was being run on the cloud or local machine.

Model path held three files that were needed in processing the model which are a json file (yolov11n.json), Hailo Execution File (yolov11n.hef) and a label file (coco.yaml). I had to change the way tracking worked in the program as well. The tracking method I previously used wouldn't work since the hef file was only for object detection and not tracking. Also, DeGirum didn't have a tracking model. The tracking method that worked was Single Online Real-Time Tracking (SORT). I did this by including a variable set to the SORT tracking model and I adjusted the way I got the bounding boxing as well as the labeling for the SORT method. We decided not to utilize the Hailo AI Kit after spending time trying to integrate it with our system with no outcome and do to time constraints which we could not build a full functioning model to use for the AI Kit. We came to the decision to not utilize Hailo AI Kit in our finished project.

9.8 Video Compression and Transmission

The prototype demonstrated in Chapter 8 was used for most of the testing involved with compression and transmission of data. This section will highlight the testing methods, problems, and solutions taken to ensure a consistently working product. There were many problems that needed to be addressed and many inconsistencies that had to be dealt with.

The first hurdle was making sure we could transmit a live feed across the local link between the Raspberry Pi 5 and the computer. This took much trial and error because it is not as simple as connecting the ethernet and having a direct connection. IP addresses played a big role in getting a connection that would work. The IP address of normal routers and internet connections starts with 192. The IP address needed for a local connection starts with 169. The IP address of the port we transmit to on the computer was 169.254.74.47.

To establish a proper connection, the static IP of the raspberry Pi must match the IP address of the port up to a certain point. At first, the IP address of the Raspberry Pi was configured to be the same as the computer. This didn't work in establishing a link. It was later found out that the subnet had to be the same but the last number of the IP address had to be different. The IP address was changed to 169.254.74.100. The command "ping ip_address" can be used to send packets to the desired IP (replace ip_address with desired IP) to see if there is an established connection. Packets were able to be sent to the Raspberry Pi but there were no packets able to be sent to the computer. The solution to this problem was disabling the firewall on the computer. It likely detected the incoming packets as hostile due to it being from an unverified source. After the firewall was disabled, a successful connection was established between both devices.

Now that a successful connection was established between both devices, the testing of transmission of data using FFmpeg could start. It was difficult to get the correct FFmpeg command to properly transfer the video feed of the camera. A makeshift solution was used in which a command was run to set up a video feed on a separate window and FFmpeg would take that window as input as the camera feed. This approach was taken because the name of the camera in devices could not be found at the time. However, this led to problems in transmission. The transmission worked well except that it would display the feed at 2x speed on the receiving end. It would also crash the FFmpeg instance when it caught up with the live feed on the Raspberry Pi 5. It was later found out that the window displaying the camera basically acted as mp4 file and when the FFmpeg transmission caught up, it would treat it as the end of the mp4 file and end the instance.

This was later solved by eventually finding the device name of the Raspberry Pi Camera Module and using that in the FFmpeg command. Once found, it was treated as a live feed and would transmit at normal speed with little to no delay and no crashing. It was then integrated into the YOLO code being used so that we could transmit the live feed with body recognition. This was not difficult to implement and ended up working quite well. There was a large delay of around 5-6 seconds but this was because the YOLO code was running at around 2-3 fps. The YOLO instance was already heavy on the processing side of the Raspberry Pi and transmission on top of that resulted in a lot of latency. Without the YOLO code running, the webcam transmission runs at 4k resolution at 30fps.

With that working, it was time to integrate it into the PCB. The C code was written to be flashed onto the PCB. The C code took in the packets being received via the ethernet module and wrote it to the serial monitor. There was another C code that pipelined the serial monitor data into an FFmpeg command. It was done this way because FFmpeg could not directly read from the serial monitor. There was a lot of noise due to the way it was set up with wires. This resulted in corrupt transmissions and unfortunately could not be solved within the timeframe of the project. Below are images of transmissions utilizing the PCB. Details can be seen in the window of the receiving side but the colors are corrupted and there is pixelation.

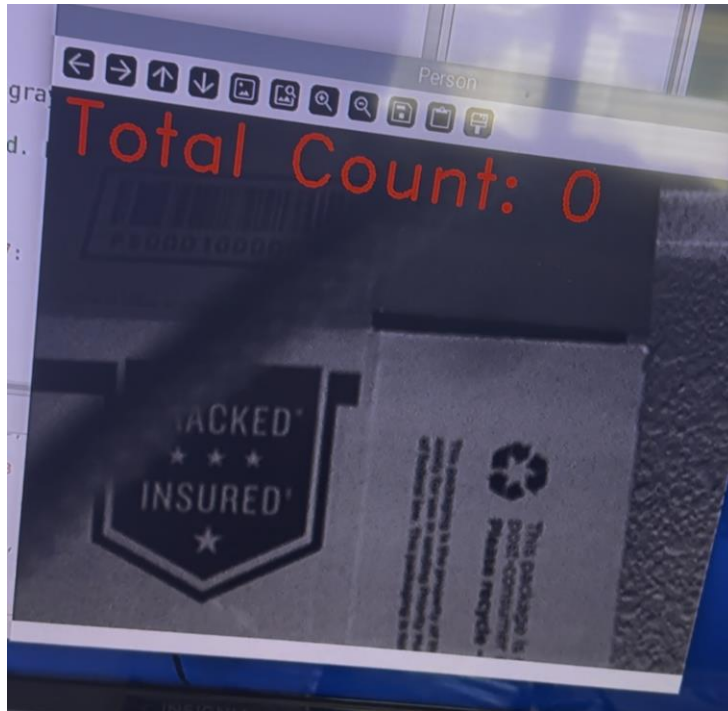


Figure 9.15. Window before transmission (Example 1)

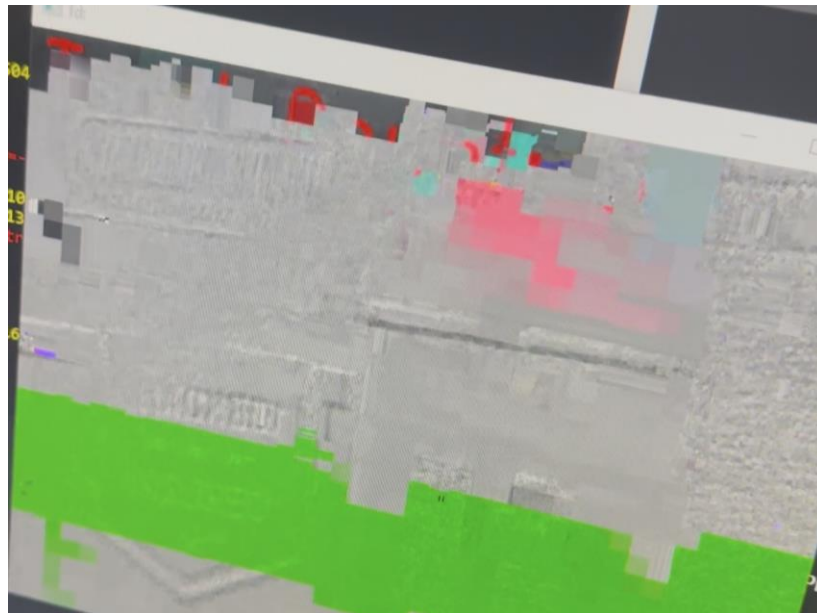


Figure 9.16. Window after transmission (Example 1)

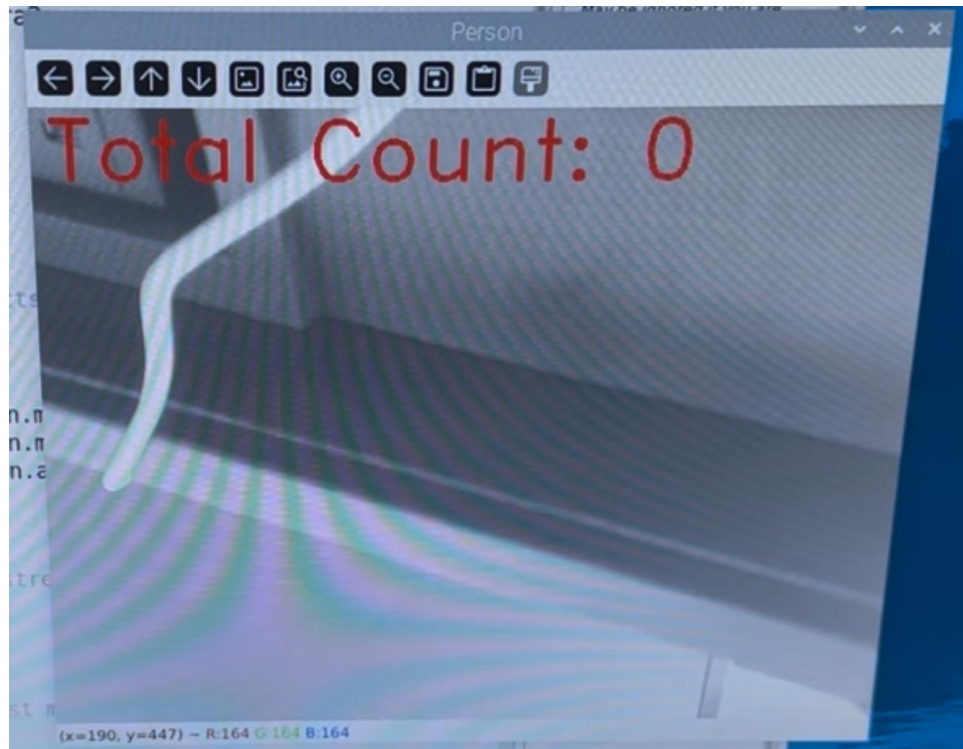


Figure 9.17. Window before transmission (Example 2)

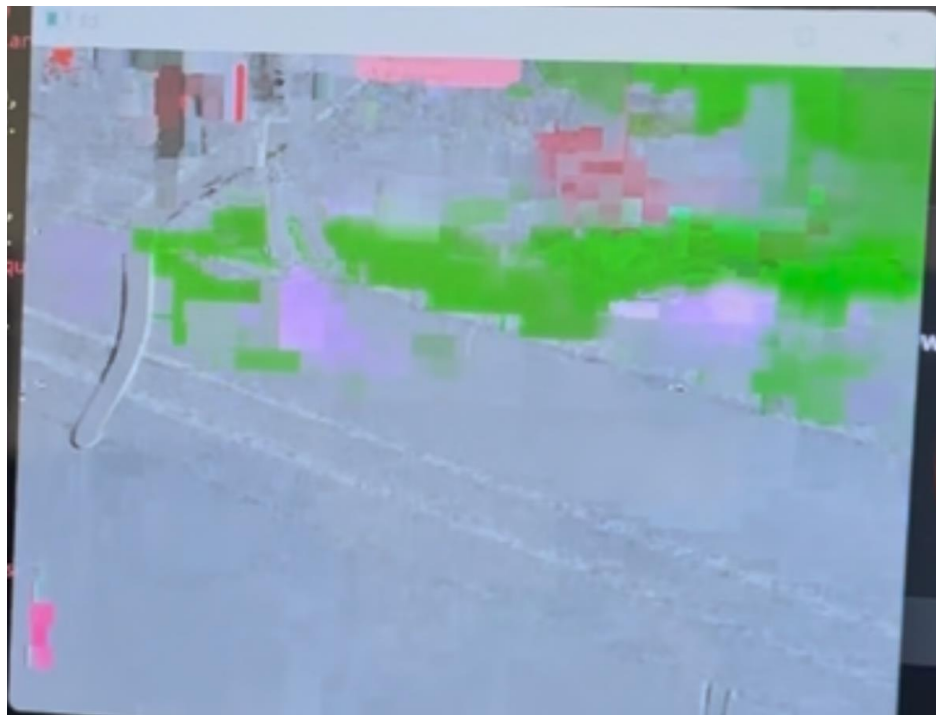


Figure 9.18. Window before transmission (Example 2)

Chapter 10 – Administrative Content

The success of the project equally depends on the administration of its many parts and engineers. A detailed discussion on the group’s financial budget and distribution of work is necessary to understand the decision-making process. This section discusses the project’s budget, bill of materials, distribution of work, and overall milestones.

10.1 Budget and Bill of Materials

The iSPY project is to be entirely self-funded by its group members. It’s the authors goal to minimize costs and limit expenditures to less than \$1000. The table below highlights the distribution of costs across the different sub-systems and all the components that make up iSPY.

Table 10.1. Itemized bill of materials.

Sub-System	Component	Part Name	Budget	Quantity	Unit Cost	Real Cost
Imaging	Camera Module	Raspberry Pi Camera Module	\$80.00	2	\$31.77	\$63.54
	Imaging Optics	Pinhole	\$300.00	1	\$78.00	\$0.00
		Aspheric Lens		1	\$139.92	\$0.00
		Spherical Lens		1	\$35.00	\$0.00
Communication	Transceiver	SFP1G-EX-55	\$100.00	2	\$27.00	\$54.00
	Media Converter	Open-SFP to RJ46 Converter	\$50.00	2	\$18.99	\$1.00
	Collimating Optics	F280FC-1550	\$300.00	2	\$181.00	\$0.00
	Optical Fiber	SMF-28e	\$25.00	2	\$6.65	\$13.30
PCB	Base Carrier PCB	iSPY	\$60.00	5	\$17.00	\$17.00
	Daughter Boards	HDMI		5	\$52.15	\$52.15
		Ethernet		5	\$40.25	\$40.25
		Voltage Regulator		5	\$10.20	\$10.20
	Misc.	~	~	90	\$115.00	\$115.00
Power	External Power Supply	Meanwell LRS-50-5	\$30.00	5	\$17.01	\$34.02

10.2 Distribution Worktable

The distribution worktable below details the primary and secondary members assigned to each responsibility. Each group member is assigned as the primary member for a specific sub-system and as secondary member for two or more responsibilities.

Table 10.2 Distribution worktable.

Sub-System	Responsibilities	Primary Member	Secondary Member
FSO	FSO Lens Design and Implementation	Alejandro	Gabryella
	Laser and Detector Selection and Implementation		Gabryella
	Optoelectronics integration with PCB Housing		Robert
Imaging	Design And Align Lenses for Image Acquisition System	Gabryella	Alejandro
	Selection of Appropriate Image Sensor		Alejandro
	Optimization of Image Quality		Luke
	Integration of Image Acquisition System		Luke
Object Detection	Computer-Vision Software Design and Implementation	Robert	Luke
	Appropriate MCU Selection		Armani
	Machine Learning Algorithm Development		Luke
Image Processing	Image Correctness	Luke	Armani
	Output Video Processing and Display		Robert
	Compression Software Design and Implementation		Robert
Electrical Hardware	PCB Design and Fabrication	Armani	Luke
	PSU Design and Implementation		Robert
	Interface Design for Data Transmission		Robert
	Signal Conditioning and Processing		Robert
Website Management	Organize and update the group's website with assignments before deadline	Luke	Robert

10.3 Milestones

The success of the project depends on individual and collective achievements. To create a good dynamic amongst group members and to maintain pace, the collective milestones encountered in Senior Design 1 and Senior Design 2 are highlighted below.

Table 10.3. Senior Design 1 Milestones - Project design phase.

Senior Design I Milestones		
Task	Status	Deadline
First group meeting	Completed	August 15, 2024
Project selection/Brainstorming	Completed	August 25, 2024
Group meeting to discuss D&C document and committee formation	Completed	August 27, 2024
Idea presentation to Dr. Kar	Completed	August 29, 2024
Submit Divide and Conquer 10 pages report	Completed	September 6, 2024
Committee Formation	Completed	September 6, 2024
Divide and Conquer Meeting with Advisors	Completed	September 9, 2024
Fix Divide and Conquer report	Completed	September 27, 2024
Upload Divide and Conquer Report on website	Completed	September 27, 2024
Midterm Demo (CREOL)	Completed	October 17, 2024
Upload 75 Page Report	Completed	October 25, 2024
75 Pages Report Meeting with Advisors	Completed	October 29, 2024
Fix 75 Pages Report	Completed	November 8, 2024
Upload 75 Pages report on website	Completed	November 8, 2024
Final Demo (CREOL)	Completed	November 21, 2024
SD1 Final Report	Completed	November 26, 2024
Upload Mini Demo Video on website	Completed	November 26, 2024
Project Research	Completed	November 2024

Table 10.4. Senior Design 2 Milestones - Project creation phase.

Senior Design II Milestones		
Task	Status	Deadline
Project Research	Completed	January 2025
Prototype Development	Completed	January 2025
Testing	Completed	February 2025
Troubleshooting	Completed	February 2025
CDR	Completed	February 2025
Midterm Demo Meeting	Completed	February 2025
Conference Paper	Completed	March 2025
Prototype Development	Completed	March 2025
Testing and Optimizing	Completed	April 2025
Final Demo	Completed	April 2025
CREOL Showcase	Completed	April 2025

10.4 Acknowledgements and Declaration

Thanks to the collaborative spirit of the Authors, this project is a success. Thank you to those closest to us and their continued support. Big thank you to our review committee and all external advisors. This project represents the Authors' drive to learn something and explore a field that is increasingly more abundant in the telecommunication industry.

Declaration: We hereby declare that we have not copied more than 7 pages from the Large Language Model (LLM). We have utilized LLM for drafting, outlining, comparing, summarizing, and proofreading purposes.

We have also used LLM to create the cover page image to illustrate our project; however, this is the only AI generated image in our project.

Chapter 11 – Conclusion

In conclusion, the iSPY project was developed to provide a secure and efficient surveillance system utilizing Free-Space Optical (FSO) communication and computer vision. As a group, we came from different backgrounds of study. Two of us are in computer engineering, two of us are in photonic science and engineering, and one of us is in electrical engineering. We were able to come together and combine our skills and knowledge to further advance the future of security.

The goal when designing the surveillance system was to design a system that could not be intercepted or traced. What sets apart iSPY compared to other surveillance systems is the free space data transmission. FSO allows us to transfer data securely without vulnerabilities commonly associated with other forms of wireless transmission. The only way to obtain the information being transferred via free space is to be in direct line of transmission. FSO transmissions also have high potential for high-speed and high-bandwidth transmission. This ensures that a live video feed can be properly displayed with high quality and little latency.

Another innovation that sets apart iSPY from other surveillance systems are computer vision algorithms. The system can incorporate object detection using a live feed over just simply recording a live feed. These algorithms can run in real time to track and identify individuals in a given area. This is especially useful in urban areas or even military operations. Being able to identify individuals on a live feed allows for the opportunity to react to potential threats before they become a reality. iSpy combines FSO technology with computer vision algorithms to achieve a superior advantage over other surveillance systems.

The hardware of iSPY has been carefully designed by the team to achieve the most efficient and high-quality performance. There are three primary subsystems in the hardware of iSPY: image acquisition system, FSO system, and electrical hardware. The image acquisition system aims to achieve a wide field of view in a compact (using less lenses than the cameras on the market) and cheaper form, and a high-quality imaging to ensure that the live feed is suitable for security purposes. The compression of the live feed is handled using H.264. The feed is ready for transmission after compression. The FSO system transmits the feed via bitstream and ensure that it is transmitted safely and securely. The customized PCB receives the transmitted data and perform decompression to properly display the live feed. This process utilizes the Raspberry Pi 4 Model B to run OpenCV and FFmpeg and the custom PCB to decompress. This orientation achieves the most optimal performance to achieve the best performance possible.

Throughout the development of iSPY, we have overcome many challenges. It was important to make sure that our PCB worked well and was effective at performing the tasks we need of it. MCU selection was an integral process to the success of the project. Many MCUs were considered before a decision was made. There were many ideas and products that were ultimately not chosen due to finding options that outweighed others. However, even after facing many challenges, our team came out with a deeper understanding of what

needed to be accomplished. We came out with a better design because of the mistakes and challenges along the way.

There are so many potential applications of iSPY in our society. Urban areas could have security and comfort knowing that there are measures in place to identify and prevent suspicious activities. In military applications, it provides a safe and secure transmission of data that cannot be intercepted. All of these things makes iSPY a valuable tool when it comes to surveillance.

The success of developing iSPY would not be possible without the efforts of our team. Our combined knowledge of different fields of engineering enabled us to tackle any obstacle in our way. Challenges that were faced alone could be made simple by collaborating with another team member.

iSPY represents the future in the field of surveillance systems. The combination of FSO systems and computer vision algorithms offers a unique and advantageous solution to modern security systems. In the future, we will continue to refine and build upon what we've accomplished with this project. This report serves as documentation for our journey to this point and shows both the challenges and victories that lead us to this moment. iSPY will contribute much to society and be the foundation for the future of surveillance.

Appendix A – References

- [1] W. G. Schulz, “Eye Spy: Optics, Photonics, and Espionage,” *Photonics Focus*, vol. 5, no. 2, pp. 19–21, Mar. 2024.
- [2] G. Chen, “Advancing Mobile Sensing with Resilient Distributed Computing,” *Photonics Focus*, vol. 5, no. 2, pp. 19–21, Mar. 2024.
- [3] S. M. Kelly, “That security camera and smart doorbell you’re using may have some major security flaws | CNN Business,” CNN. Accessed: Aug. 30, 2024. [Online]. Available: <https://www.cnn.com/2024/03/09/tech/smart-home-cameras-hackers-security/index.html>
- [4] N. Vigdor, “Somebody’s Watching: Hackers Breach Ring Home Security Cameras,” *The New York Times*, Dec. 15, 2019. Accessed: Aug. 30, 2024. [Online]. Available: <https://www.nytimes.com/2019/12/15/us/Hacked-ring-home-security-cameras.html>
- [5] S. Ikeda, “Russian Agents Hacking Residential Surveillance Cameras to Gather Intel in Ukraine,” CPO Magazine. Accessed: Aug. 30, 2024. [Online]. Available: <https://www.cpomagazine.com/cyber-security/russian-agents-hacking-residential-surveillance-cameras-to-gather-intel-in-ukraine/>
- [6] J. Pattison Tuohy and R. Henderson, “How To Keep Your Home Security Cameras Safe,” US News & World Report. Accessed: Aug. 30, 2024. [Online]. Available: [//www.usnews.com/360-reviews/services/security-cameras/how-to-keep-your-security-cameras-safe](https://www.usnews.com/360-reviews/services/security-cameras/how-to-keep-your-security-cameras-safe)
- [7] “Features – Koruza.” Accessed: Sep. 05, 2024. [Online]. Available: <http://www.koruza.net/features/>
- [8] “Wireless Optical Data Transmission Technology KORUZA: Upgrade Announcement! - IRNAS.” Accessed: Sep. 05, 2024. [Online]. Available: <https://www.irnas.eu/wireless-optical-data-transmission-technology-koruza-upgrade-announcement/>
- [9] “MOSTCOM – PathFinder Digital Wiki.” Accessed: Sep. 05, 2024. [Online]. Available: <https://wiki.pathfinderdigital.com/wiki/mostcom/>
- [10] R. Awati, “What is field of view (FOV)?,” TechTarget. Accessed: Sep. 01, 2024. [Online]. Available: <https://www.techtarget.com/whatis/definition/field-of-view-FOV>
- [11] D. Kim, S. Chang, and H.-S. Kwon, “Wide Field-of-View, High-Resolution Endoscopic Lens Design with Low F-Number for Disposable Endoscopy,” *Photonics*, vol. 8, no. 4, Art. no. 4, Apr. 2021, doi: 10.3390/photonics8040089.
- [12] Threat.Media, “What is a Bitstream?,” [Online]. Available: <https://threat.media/definition/what-is-a-bitstream/#:~:text=A%20bitstream%20is%20a%20sequence,any%20other%20type%20of%20file>
- [13] L. Hardesty, “Explained: Neural networks,” MIT News | Massachusetts Institute of Technology. Accessed: Sep. 05, 2024. [Online]. Available: <https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414>
- [14] D. C. O’Shea and J. L. Bentley, *Designing Optics Using Zemax OpticStudio*, 1st ed., vol. PM367. SPIE, 2024.
- [15] “CMOS Sensor Size Comparison Guide and Lens Calculator,” Commonlands Optics. Accessed: Sep. 14, 2024. [Online]. Available: <https://commonlands.com/blogs/technical/cmos-sensor-size>

- [16] R. Paschotta, "Field of view." Accessed: Sep. 03, 2024. [Online]. Available: https://www.rp-photonics.com/field_of_view.html
- [17] T. Don, *Camera Sensor Size Explained*, (Jan. 25, 2017). [Online Video]. Available: https://www.youtube.com/watch?v=cBV8wpNmnpY&ab_channel=Channel8
- [18] S. d'Entremont, *Does sensor size affect aperture? The TRUTH about CROP FACTOR.*, (Apr. 11, 2024). [Online Video]. Available: https://www.youtube.com/watch?v=Apum3Ezqo0U&ab_channel=Simond%27Entremont
- [19] "Digital Camera Sensor Sizes: How it Influences Your Photography." Accessed: Sep. 14, 2024. [Online]. Available: <https://www.cambridgeincolour.com/tutorials/digital-camera-sensor-size.htm>
- [20] J. M. Geary, *Introduction to Lens Design: with practical ZEMAX examples*, 1st ed. Richmond, Virginia: Willmann-Bell, Inc, 2022.
- [21] B. Saleh, Ed., *Introduction to Subsurface Imaging*. Cambridge: Cambridge University Press, 2011. doi: 10.1017/CBO9780511732577.
- [22] S. d'Entremont, "HOW MANY MEGAPIXELS DO YOU NEED? YOU MAY BE SURPRISED! - YouTube." Accessed: Sep. 16, 2024. [Online]. Available: https://www.youtube.com/watch?v=ThpQWhOfKO4&ab_channel=Simond%27Entremont
- [23] "1080p vs. 2K vs. 4K Security Cameras: Understanding the Differences | Mammoth Security Inc." Accessed: Sep. 16, 2024. [Online]. Available: <https://mammothsecurity.com/blog/1080p-vs-2k-vs-4k-security-cameras>
- [24] N. James and G. Hollows, "Resolution," Edmund Optics. Accessed: Sep. 16, 2024. [Online]. Available: <https://www.edmundoptics.com/knowledge-center/application-notes/imaging/resolution/>
- [25] "What are Frames Per Second (FPS)?" Accessed: Sep. 18, 2024. [Online]. Available: <https://www.isarsoft.com/knowledge-hub/fps>
- [26] Elvia, "Complete Guide to Security Camera CMOS VS CCD Image Sensors," Reolink Blog. Accessed: Sep. 20, 2024. [Online]. Available: <https://reolink.com/blog/security-camera-cmos-vs-ccd-image-sensors/>
- [27] "Electronic shutter vs mechanical shutter," Canon Europe. Accessed: Sep. 18, 2024. [Online]. Available: <https://www.canon-europe.com/pro/infobank/electronic-vs-mechanical-shutter/>
- [28] IPVM Team, "Shutter Speed / Exposure Tutorial," IPVM. Accessed: Sep. 20, 2024. [Online]. Available: <https://ipvm.com/reports/exposure-setting-tutorial>
- [29] "Rolling Shutter vs Global Shutter sCMOS Camera Mode," Oxford Instruments. Accessed: Sep. 20, 2024. [Online]. Available: <https://andor.oxinst.com/learning/view/article/rolling-and-global-shutter>
- [30] C. Yang, "Global Shutter vs. Rolling Shutter in Embedded Vision," TechNexion. Accessed: Sep. 20, 2024. [Online]. Available: <https://www.technexion.com/resources/global-shutter-vs-rolling-shutter-in-embedded-vision/>
- [31] AlliedVisionTV, "Sensor module vs. camera module." Accessed: Sep. 21, 2024. [Online]. Available: https://www.youtube.com/watch?v=Oi4S7SwumGs&ab_channel=AlliedVisionTV

- [32] B. McCurrach, “A Quick Overview Of Industrial Camera Interfaces,” *Quality Magazine*. Accessed: Sep. 21, 2024. [Online]. Available: <https://www.qualitymag.com/articles/97108-a-quick-overview-of-industrial-camera-interfaces>
- [33] AlliedVisionTV, “Which camera interface is the right one for my application?” Accessed: Sep. 21, 2024. [Online]. Available: https://www.youtube.com/watch?v=Co2PR1wDRLk&ab_channel=AlliedVisionTV
- [34] P. Kumar, “What is a MIPI Interface? How does MIPI Camera Work?,” *e-con Systems*. Accessed: Sep. 29, 2024. [Online]. Available: <https://www.e-consystems.com/blog/camera/technology/what-is-a-mipi-camera-how-does-mipi-camera-work/>
- [35] “Fixed focal length lenses for machine vision applications,” *Opto Engineering Basics*. Accessed: Sep. 28, 2024. [Online]. Available: <https://www.opto-e.com/en/basics/fixed-focal-length-lenses>
- [36] P. Kumar, “Fixed-focus lens vs. autofocus lens: How to choose the best lens for your application,” *e-con Systems*. Accessed: Sep. 29, 2024. [Online]. Available: <https://www.e-consystems.com/blog/camera/technology/fixed-focus-lens-vs-autofocus-lens-how-to-choose-the-best-lens-for-your-application/>
- [37] “Numerical Aperture (NA),” *MEETOPTICS*. Accessed: Sep. 28, 2024. [Online]. Available: <https://www.meetoptics.com/academy/numerical-aperture?srsltid=AfmBOorbSLQJmJttjanMOndBKgcGwBhzVMk5bEIVihX9W5EDwPb9ngw3>
- [38] C. Olson and J. Bentley, *Field Guide to Lens Design*, vol. FG27. SPIE, 2012. [Online]. Available: <https://doi.org/10.1117/3.934997>
- [39] M. J. Riedl, “Diffraction Limit,” *SPIE*. Accessed: Sep. 29, 2024. [Online]. Available: https://spie.org/publications/spie-publication-resources/optipedia-free-optics-information/tt48_38_diffraction_limit
- [40] Ansys Learning, *The Airy Disk and Diffraction Limit — Lesson 4*, (2023). Accessed: Sep. 29, 2024. [Online Video]. Available: <https://www.youtube.com/watch?v=EbN7rNe98k4>
- [41] J. Chouinard, “What Is a Lens Optical Format? | Sensor Size vs Sensor Format,” *1stVision Inc. - Machine Vision Articles*. Accessed: Sep. 15, 2024. [Online]. Available: <https://www.1stvision.com/machine-vision-solutions/2017/08/how-does-a-lens-optical-format-relate-to-machine-vision-cameras.html>
- [42] “Image Circle,” *VS Technology*. Accessed: Sep. 15, 2024. [Online]. Available: <https://vst.co.jp/en/glossary/image-circle/>
- [43] F. C. Wippermann, E. Beckert, P. Dannberg, B. Messerschmidt, and G. Seyffert, “Disposable low-cost video endoscopes for straight and oblique viewing direction with simplified integration,” in *Design and Quality for Biomedical Technologies III*, SPIE, Feb. 2010, pp. 25–34. doi: 10.1117/12.840692.
- [44] S. Sparrold, *Optics Tutorial - 10 - Achromatic Doublets*, (Jul. 29, 2013). Accessed: Oct. 18, 2024. [Online Video]. Available: <https://www.youtube.com/watch?v=DspKhNiygDU>
- [45] “Abbe-Number,” *Edmund Optics*. Accessed: Oct. 18, 2024. [Online]. Available: <https://www.edmundoptics.com/knowledge-center/glossary/abbe-number/>

- [46] “Why Use an Achromatic Lens?” Accessed: Oct. 18, 2024. [Online]. Available: <https://www.edmundoptics.com/knowledge-center/application-notes/optics/why-use-an-achromatic-lens/>
- [47] “Meniscus lens,” VY Optoelectronics. Accessed: Oct. 18, 2024. [Online]. Available: <https://www.vyoptics.com/optical-meniscus-lens.html>
- [48] “Meniscus Lenses,” Altechna - Custom Laser Optics & Technologies. Accessed: Oct. 18, 2024. [Online]. Available: <https://www.altechna.com/products/lenses/meniscus-lenses/>
- [49] “Meniscus Lenses Selection Guide: Types, Features, Applications,” GlobalSpec. Accessed: Oct. 18, 2024. [Online]. Available: https://www.globalspec.com/learnmore/optics_optical_components/optical_components/meniscus_lenses
- [50] E. L. Dereniak and T. D. Dereniak, *Geometrical and Trigonometric Optics*, vol. 1. Cambridge University Press, 2008.
- [51] “All About Aspheric Lenses.” Accessed: Oct. 06, 2024. [Online]. Available: <https://www.edmundoptics.com/knowledge-center/application-notes/optics/all-about-aspheric-lenses/>
- [52] “Advantages of Fresnel Lenses,” Edmund Optics. Accessed: Oct. 18, 2024. [Online]. Available: <https://www.edmundoptics.com/knowledge-center/application-notes/optics/advantages-of-fresnel-lenses/>
- [53] H. Grigonis, “What is a Telephoto Lens? (Why and When you Need One),” ExpertPhotography. Accessed: Oct. 18, 2024. [Online]. Available: <https://cms.expertphotography.com/telephoto-lens-camera-equipment/>
- [54] J. E. Greivenkamp, “Elementary Optical Systems - Section 12 - Objectives.” College of Optical Sciences - The University of Arizona, 2019. [Online]. Available: <https://wp.optics.arizona.edu/jgreivenkamp/wp-content/uploads/sites/11/2019/08/502-12-Objectives.pdf>
- [55] “N-BK7 and BK7.” Accessed: Oct. 18, 2024. [Online]. Available: <https://www.mindrum.com/glass/n-bk7-and-bk7/>
- [56] M. N. Polyanskiy, “Refractiveindex.info database of optical constants,” *Sci. Data*, vol. 11, no. 1, p. 94, Jan. 2024, doi: 10.1038/s41597-023-02898-2.
- [57] “N-SF11 Plano-Convex Lenses,” Thorlabs. Accessed: Oct. 18, 2024. [Online]. Available: <https://www.thorlabs.com>
- [58] P. Sahoo, S. K. Das, and J. Paulo Davim, “3.3 Surface Finish Coatings,” in *Comprehensive Materials Finishing*, M. Hashmi, Ed., Oxford: Elsevier, 2017, pp. 38–55. doi: 10.1016/B978-0-12-803581-8.09167-0.
- [59] “An Introduction to Optical Coatings,” Edmund Optics. Accessed: Oct. 19, 2024. [Online]. Available: <https://www.edmundoptics.com/knowledge-center/application-notes/lasers/an-introduction-to-optical-coatings/>
- [60] D. Ristau, “OPTICAL COATINGS | Thin Film Optical Coatings,” in *Encyclopedia of Modern Optics*, R. D. Guenther, Ed., Oxford: Elsevier, 2005, pp. 360–369. doi: 10.1016/B0-12-369395-0/00880-0.
- [61] “Coatings, Anti-Reflective, Single Layer MgF2,” Esco Optics, Inc. Accessed: Oct. 19, 2024. [Online]. Available: <https://escooptics.com/products/coatings-anti-reflective-single-layer-mgf2>

- [62] “Comparison of Optical Aberrations.” Accessed: Oct. 19, 2024. [Online]. Available: <https://www.edmundoptics.com/knowledge-center/application-notes/optics/comparison-of-optical-aberrations/>
- [63] “Coma.” Accessed: Oct. 19, 2024. [Online]. Available: <https://site.physics.georgetown.edu/~vankeu/webtext/Aberrations/Coma.html>
- [64] N. James and G. Hollows, “Distortion,” Edmund Optics. Accessed: Oct. 19, 2024. [Online]. Available: <https://www.edmundoptics.com/knowledge-center/application-notes/imaging/distortion/>
- [65] “Introduction to Modulation Transfer Function.” Accessed: Nov. 10, 2024. [Online]. Available: <https://www.edmundoptics.com/knowledge-center/application-notes/optics/introduction-to-modulation-transfer-function/>
- [66] B. Saleh, “4. Physical Optics of The Eye.” Mar. 2024.
- [67] Junyi Chai, H. Zeng, A. Li, and E. W. T. Ngai, “Deep learning in computer vision: A critical review of emerging techniques and application scenarios,” *Mach. Learn. Appl.*, vol. 6, p. 100134, Dec. 2021, doi: 10.1016/j.mlwa.2021.100134.
- [68] N. O’Mahony *et al.*, “Deep Learning vs. Traditional Computer Vision,” in *Advances in Computer Vision*, vol. 943, K. Arai and S. Kapoor, Eds., in *Advances in Intelligent Systems and Computing*, vol. 943, Cham: Springer International Publishing, 2020, pp. 128–144. doi: 10.1007/978-3-030-17795-9_10.
- [69] Y. Jia *et al.*, “Caffe: Convolutional Architecture for Fast Feature Embedding,” in *Proceedings of the 22nd ACM international conference on Multimedia*, Orlando Florida USA: ACM, Nov. 2014, pp. 675–678. doi: 10.1145/2647868.2654889.
- [70] “7 Best Computer Vision Development Libraries in 2024,” Labellerr. Accessed: Sep. 24, 2024. [Online]. Available: <https://www.labellerr.com/blog/top-7-computer-vision-development-tools-2024/>
- [71] “BoofCV.” Accessed: Sep. 24, 2024. [Online]. Available: https://boofcv.org/index.php?title=Main_Page
- [72] “7 Best Computer Vision Libraries in Python,” ProjectPro. Accessed: Sep. 24, 2024. [Online]. Available: <https://www.projectpro.io/article/computer-vision-libraries/772>
- [73] “SimpleCV Tutorial — Tutorial.” Accessed: Sep. 24, 2024. [Online]. Available: <https://tutorial.simplecv.org/en/latest/>
- [74] “About,” OpenCV. Accessed: Sep. 25, 2024. [Online]. Available: <https://opencv.org/about/>
- [75] “OpenCV: Introduction to OpenCV-Python Tutorials.” Accessed: Sep. 25, 2024. [Online]. Available: https://docs.opencv.org/4.x/d0/de3/tutorial_py_intro.html
- [76] “Body Detection with Computer Vision | Latest | Instrument.” Accessed: Sep. 26, 2024. [Online]. Available: <https://www.instrument.com/latest/body-detection-with-computer-vision>
- [77] A. Tam, “Using Haar Cascade for Object Detection,” *MachineLearningMastery.com*. Accessed: Sep. 26, 2024. [Online]. Available: <https://machinelearningmastery.com/using-haar-cascade-for-object-detection/>
- [78] A. Rosebrock, “OpenCV Haar Cascades,” *PyImageSearch*. Accessed: Sep. 26, 2024. [Online]. Available: <https://pyimagesearch.com/2021/04/12/opencv-haar-cascades/>

- [79] N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, San Diego, CA, USA: IEEE, 2005, pp. 886–893. doi: 10.1109/CVPR.2005.177.
- [80] A. Drake, “Unveiling the Potential of Histogram of Oriented Gradients (HOG) in Computer Vision,” Comet. Accessed: Oct. 06, 2024. [Online]. Available: <https://www.comet.com/site/blog/unveiling-the-potential-of-histogram-of-oriented-gradients-hog-in-computer-vision>
- [81] A. Tam, “Extracting Histogram of Gradients with OpenCV,” MachineLearningMastery.com. Accessed: Oct. 06, 2024. [Online]. Available: https://machinelearningmastery.com/opencv_hog/
- [82] S. AI, “Histogram of Oriented Gradients (HOG) — Simplest Intuition,” Medium. Accessed: Oct. 06, 2024. [Online]. Available: <https://medium.com/@skillcate/histogram-of-oriented-gradients-hog-simplest-intuition-2392995f8010>
- [83] “Histogram of Oriented Gradients explained using OpenCV.” Accessed: Oct. 06, 2024. [Online]. Available: <https://learnopencv.com/histogram-of-oriented-gradients/>
- [84] A. Mirkhan, “YOLO Algorithm: Real-Time Object Detection from A to Z,” Kili Technology. Accessed: Oct. 13, 2024. [Online]. Available: <https://kili-technology.com/data-labeling/machine-learning/yolo-algorithm-real-time-object-detection-from-a-to-z>
- [85] Melanie, “You Only Look Once (YOLO): What is it?,” Data Science Courses | DataScientest. Accessed: Oct. 13, 2024. [Online]. Available: <https://datascientest.com/en/you-only-look-once-yolo-what-is-it>
- [86] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, “YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information,” Feb. 29, 2024, *arXiv*: arXiv:2402.13616. Accessed: Oct. 15, 2024. [Online]. Available: <http://arxiv.org/abs/2402.13616>
- [87] Ultralytics, “YOLOv9.” Accessed: Oct. 15, 2024. [Online]. Available: <https://docs.ultralytics.com/models/yolov9>
- [88] “YOLOv9: Advancing the YOLO Legacy.” Accessed: Oct. 15, 2024. [Online]. Available: <https://learnopencv.com/yolov9-advancing-the-yolo-legacy/>
- [89] “YOLOv10: Real-Time End-to-End Object Detection.” Accessed: Oct. 15, 2024. [Online]. Available: <https://arxiv.org/html/2405.14458v1>
- [90] “YOLOv10: Paper Explanation and Inference Results.” Accessed: Oct. 15, 2024. [Online]. Available: <https://learnopencv.com/yolov10/>
- [91] Saiwa, “YOLOv10 | explanation, features, and implementation,” Medium. Accessed: Oct. 15, 2024. [Online]. Available: <https://medium.com/@saiwadotai/yolov10-explanation-features-and-implementation-c67b49e44120>
- [92] S. Arya, “YOLOv10: Revolutionizing Real-Time Object Detection,” Analytics Vidhya. Accessed: Nov. 03, 2024. [Online]. Available: <https://www.analyticsvidhya.com/blog/2024/07/yolov10-for-realtime-object-detection/>
- [93] Ultralytics, “YOLO11 🚀 NEW.” Accessed: Oct. 17, 2024. [Online]. Available: <https://docs.ultralytics.com/models/yolo11>

- [94] “YOLO11: Faster Than You Can Imagine!” Accessed: Oct. 17, 2024. [Online]. Available: <https://learnopencv.com/yolo11/>
- [95] Z. Shariff, “YOLOv11 Is Officially Out! What You Need To Know!,” Medium. Accessed: Oct. 17, 2024. [Online]. Available: <https://medium.com/@zainshariff6506/yolov11-is-officially-out-what-you-need-to-know-6738c5d25be1>
- [96] N. Singh, “Meet YOLO-NAS: An Open-Sourced YOLO-based Architecture Redefining State-of-the-Art in Object Detection,” MarkTechPost. Accessed: Oct. 18, 2024. [Online]. Available: <https://www.marktechpost.com/2023/05/08/meet-yolo-nas-an-open-sourced-yolo-based-architecture-redefining-state-of-the-art-in-object-detection/>
- [97] Ultralytics, “YOLO-NAS (Neural Architecture Search).” Accessed: Oct. 18, 2024. [Online]. Available: <https://docs.ultralytics.com/models/yolo-nas>
- [98] “YOLO-NAS: New Object Detection Model Beats YOLOv6 & YOLOv8.” Accessed: Oct. 18, 2024. [Online]. Available: <https://learnopencv.com/yolo-nas/>
- [99] R. Vaidya, “Deci’s YOLO-NAS: Next-Generation Model for Object Detection.,” CodeX. Accessed: Oct. 18, 2024. [Online]. Available: <https://medium.com/codex/decis-yolo-nas-next-generation-model-for-object-detection-8ccb7f2013a7>
- [100] A. S. Gillis, “What is a codec?,” Unified Communications. Accessed: Oct. 22, 2024. [Online]. Available: <https://www.techtarget.com/searchunifiedcommunications/definition/codec>
- [101] “Lossy vs Lossless Compression: Differences & Advantages | Adobe.” Accessed: Oct. 22, 2024. [Online]. Available: <https://www.adobe.com/uk/creativecloud/photography/discover/lossy-vs-lossless.html>
- [102] Z. Powell, “Lossy vs Lossless Compression: Comprehensive Analysis - ShortPixel Blog.” Accessed: Oct. 23, 2024. [Online]. Available: <https://shortpixel.com/blog/lossy-vs-lossless/>
- [103] D. Molenhouse, “Understanding the Difference between Interframe & Intraframe Compression.” Accessed: Oct. 23, 2024. [Online]. Available: <https://info.verkada.com/Interframe-vs-Intraframe-Compression/>
- [104] ~ Prateek Joshi, “What Is Entropy Coding?,” Perpetual Enigma. Accessed: Oct. 23, 2024. [Online]. Available: <https://prateekvjoshi.com/2014/12/06/what-is-entropy-coding/>
- [105] “Network Bandwidth: What Is Bandwidth in Networking? - IT Glossary | SolarWinds.” Accessed: Oct. 23, 2024. [Online]. Available: <https://www.solarwinds.com/resources/it-glossary/network-bandwidth>
- [106] “Video Codecs - List of the best codecs and how they work.” Accessed: Oct. 23, 2024. [Online]. Available: <https://getstream.io/glossary/video-codecs/>
- [107] “FFV1,” *Wikipedia*. Sep. 06, 2024. Accessed: Oct. 23, 2024. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=FFV1&oldid=1244343241>
- [108] “Pulse Code Modulation - Modulation, Types, Advantages and Disadvantages, Applications,” BYJUS. Accessed: Oct. 23, 2024. [Online]. Available: <https://byjus.com/physics/pulse-code-modulation/>

- [109] “Pulse Code Modulation - an overview | ScienceDirect Topics.” Accessed: Oct. 23, 2024. [Online]. Available: <https://www.sciencedirect.com/topics/engineering/pulse-code-modulation>
- [110] “Modulation Basics – Wavelength Electronics.” Accessed: Oct. 23, 2024. [Online]. Available: <https://www.teamwavelength.com/modulation-basics/>
- [111] “Error Correcting Codes - Reed-Solomon codes.” Accessed: Oct. 23, 2024. [Online]. Available: <https://www.tutorialspoint.com/error-correcting-codes-reed-solomon-codes>
- [112] “Error Correcting Codes - Hamming codes.” Accessed: Oct. 23, 2024. [Online]. Available: <https://www.tutorialspoint.com/error-correcting-codes-hamming-codes>
- [113] “About FFmpeg.” Accessed: Oct. 23, 2024. [Online]. Available: <https://ffmpeg.org/about.html>
- [114] “GStreamer: features.” Accessed: Oct. 23, 2024. [Online]. Available: <https://gstreamer.freedesktop.org/features/>
- [115] “OpenH264.” Accessed: Oct. 23, 2024. [Online]. Available: <https://www.openh264.org/>
- [116] “JCodec -.” Accessed: Oct. 23, 2024. [Online]. Available: <http://jcodec.org/>
- [117] B. Louis, “Xuggler: The Easy Way to Work with Media Files in Java,” Xu 2Gle. Accessed: Oct. 23, 2024. [Online]. Available: <https://www.xuggler.com/xuggler/>
- [118] “Java Media Framework.” Accessed: Oct. 23, 2024. [Online]. Available: <https://www.oracle.com/java/technologies/javase/java-media-framework.html>
- [119] “PyAV Documentation — PyAV 9.0.2 documentation.” Accessed: Oct. 23, 2024. [Online]. Available: <https://pyav.org/docs/stable/>
- [120] *moviepy: Video editing with Python*. Python. Accessed: Oct. 23, 2024. [Online]. Available: <https://zulko.github.io/moviepy/>
- [121] “ESP SoCs | Espressif Systems.” Accessed: Oct. 23, 2024. [Online]. Available: <https://www.espressif.com/en/products/socs>
- [122] “esp32-s3_datasheet_en.” Espressif. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf
- [123] “ESP32-S3,” DigiKey Electronics. Accessed: Oct. 23, 2024. [Online]. Available: <https://www.digikey.com/en/products/detail/espressif-systems/ESP32-S3/15822445>
- [124] “STM32H7 - Arm Cortex-M7 and Cortex-M4 MCUs (480 MHz) - STMicroelectronics.” Accessed: Oct. 23, 2024. [Online]. Available: <https://www.st.com/en/microcontrollers-microprocessors/stm32h7-series.html>
- [125] “Buy STM32H7S3L8 - eStore - STMicroelectronics.” Accessed: Oct. 23, 2024. [Online]. Available: <https://estore.st.com/en/products/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus/stm32-high-performance-mcus/stm32h7-series/stm32h7r3-7s3/stm32h7s3l8.html>
- [126] R. P. Ltd, “RP2040 specifications,” Raspberry Pi. Accessed: Oct. 23, 2024. [Online]. Available: <https://www.raspberrypi.com/products/rp2040/>
- [127] “Raspberry Pi RP2040 Microcontroller,” CanaKit. Accessed: Oct. 23, 2024. [Online]. Available: <https://www.canakit.com/raspberry-pi-rp2040.html>
- [128] R. P. Ltd, “Raspberry Pi 4 Model B specifications,” Raspberry Pi. Accessed: Oct. 23, 2024. [Online]. Available: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>

- [129] “raspberrypi-4-product-brief.” Raspberry Pi. [Online]. Available: <https://datasheets.raspberrypi.com/rpi4/raspberrypi-4-product-brief.pdf>
- [130] “Raspberry Pi 4 1GB Model B,” CanaKit. Accessed: Oct. 23, 2024. [Online]. Available: <https://www.canakit.com/raspberrypi-4.html>
- [131] R. P. Ltd, “Buy a Raspberry Pi Zero 2 W,” Raspberry Pi. Accessed: Oct. 23, 2024. [Online]. Available: <https://www.raspberrypi.com/products/raspberrypi-zero-2-w/>
- [132] “raspberrypi-zero-2-w-product-brief.” Raspberry Pi. [Online]. Available: <https://datasheets.raspberrypi.com/rpizero2/raspberrypi-zero-2-w-product-brief.pdf>
- [133] “Raspberry Pi Zero 2 W,” CanaKit. Accessed: Oct. 23, 2024. [Online]. Available: <https://www.canakit.com/raspberrypi-zero-2-w.html>
- [134] R. P. Ltd, “Buy a Raspberry Pi 3 Model B+,” Raspberry Pi. Accessed: Oct. 23, 2024. [Online]. Available: <https://www.raspberrypi.com/products/raspberrypi-3-model-b-plus/>
- [135] “raspberrypi-3-b-plus-product-brief.” Raspberry Pi. [Online]. Available: <https://datasheets.raspberrypi.com/rpi3/raspberrypi-3-b-plus-product-brief.pdf>
- [136] “Raspberry Pi 3 Model B+ 1.4 GHz,” Chicago Electronic Distributors. Accessed: Oct. 23, 2024. [Online]. Available: <https://chicagodist.com/products/raspberrypi-model-3-b-1-4-ghz>
- [137] R. P. Ltd, “Buy a Compute Module 4,” Raspberry Pi. Accessed: Nov. 23, 2024. [Online]. Available: <https://www.raspberrypi.com/products/compute-module-4/>
- [138] “Raspberry Pi Compute Module 4 Data Sheet.” Raspberry Pi. [Online]. Available: <https://datasheets.raspberrypi.com/cm4/cm4-datasheet.pdf>
- [139] “STM32MP157A - MPU with Arm Dual Cortex-A7 650 MHz, Arm Cortex-M4 real-time coprocessor, 3D GPU, TFT/MIPI DSI displays, FD-CAN - STMicroelectronics.” Accessed: Nov. 23, 2024. [Online]. Available: <https://www.st.com/en/microcontrollers-microprocessors/stm32mp157a.html>
- [140] “i.MX 6ULL Applications Processor | Single Arm[®] Cortex[®]-A7 @ 900 MHz.” Accessed: Nov. 23, 2024. [Online]. Available: <https://www.nxp.com/products/processors-and-microcontrollers/arm-processors/i-mx-applications-processors/i-mx-6-processors/i-mx-6ull-single-core-processor-with-arm-cortex-a7-core:i.MX6ULL>
- [141] “Buy STM32MP157A - eStore - STMicroelectronics.” Accessed: Nov. 23, 2024. [Online]. Available: <https://estore.st.com/en/products/microcontrollers-microprocessors/stm32-arm-cortex-mpus/stm32mp1-series/stm32mp157/stm32mp157a.html>
- [142] “i.mx 6ull - Google Shopping.” Accessed: Nov. 23, 2024. [Online]. Available: https://www.google.com/search?sca_esv=191e8f881edcc710&rlz=1C1CHBF_enUS910US910&sxsrf=ADLYWIKoPJLPHcqHAgJNjU67x7SjVPwYrw:1732378104052&q=i.mx+6ull&udm=28&fbs=AEQNm0Aa4sjWe7Rqy32pFwRj0UkWd8nbOJfsBGG B5IQQO6L3J03RPjGV0MznOJ6Likin94pT_oR1DTSof42bOBxoTNxGeB1pS5_mu b79WlyOO98XhEgJ5ByfFkeFUd9hlGBCmUZgrzLESyrkqGBL7osK5ZH3-0drzcH2VIMmwLRButbs-cMHhV8leQ--AYHsqw2WSrddKuHyCX6iowNOKZDTcJOO0G_14g&ved=1t:220175&ictx=111&biw=958&bih=958&dpr=1

- [143] “STM32MP157AAC3,” DigiKey Electronics. Accessed: Nov. 23, 2024. [Online]. Available:
<https://www.digikey.com/en/products/detail/stmicroelectronics/STM32MP157AAC3/10238135>
- [144] “i.MX6ULL Data Sheet.” NXP. [Online]. Available:
<https://www.nxp.com/docs/en/data-sheet/IMX6ULLCEC.pdf>
- [145] “STM32MP157AAC3 Data Sheet.” STMicroelectronics. [Online]. Available:
<https://www.st.com/resource/en/datasheet/stm32mp157a.pdf>
- [146] “STM32H753VIT6 Data Sheet.” STMicroelectronics. [Online]. Available:
<https://www.st.com/content/ccc/resource/technical/document/datasheet/group3/01/21/00/39/12/c9/4c/f1/DM00388325/files/DM00388325.pdf/jcr:content/translations/en.DM00388325.pdf>
- [147] “STM32H753VIT6,” DigiKey Electronics. Accessed: Nov. 24, 2024. [Online]. Available:
<https://www.digikey.com/en/products/detail/stmicroelectronics/STM32H753VIT6/8257854>
- [148] - ScholarSpace -, “Research Guides: All About Images: What is Resolution?” Accessed: Nov. 24, 2024. [Online]. Available:
<https://guides.lib.umich.edu/c.php?g=282942&p=1885350>
- [149] D. Brunner, “Frame Rate: a Beginner’s Guide,” The TechSmith blog. Accessed: Nov. 24, 2024. [Online]. Available: <https://www.techsmith.com/blog/frame-rate-beginners-guide/>
- [150] “Surveillance Storage Calculator | Seagate US,” Seagate.com. Accessed: Nov. 24, 2024. [Online]. Available: <https://blog.seagate.com/video-storage-calculator/>
- [151] Monkey see, monkey do, *How to record video from a webcam using Python & OpenCV*, (Jul. 07, 2021). Accessed: Nov. 24, 2024. [Online Video]. Available: <https://www.youtube.com/watch?v=frYotHLJ-Rc>
- [152] “One-way communication: Definition and applications.” Accessed: Oct. 24, 2024. [Online]. Available: <https://www.oneteam.io/en/blog/what-is-one-way-communication>
- [153] “Communication; principles for a lifetime,” *Ref. Res. Book News*, vol. 16, no. 3, 2001.
- [154] “One-way Communication: Definition, Examples, Benefits & Process,” Clearinfo. Accessed: Oct. 24, 2024. [Online]. Available:
<https://clearinfo.in/blog/what-is-one-way-communication/>
- [155] “What is two-way communication? Importance of two-way communication.” Accessed: Oct. 24, 2024. [Online]. Available:
<https://thebusinesscommunication.com/two-way-communication-and-importance/>
- [156] A. removed at request of original Publisher, “1.2 The Communication Process,” Sep. 2016, Accessed: Oct. 24, 2024. [Online]. Available:
<https://open.lib.umn.edu/communication/chapter/1-2-the-communication-process/>
- [157] “Everything You Need to Know About Coaxial Cable | Lenovo US.” Accessed: Oct. 24, 2024. [Online]. Available: <https://www.lenovo.com/us/en/glossary/coaxial-cable/>

- [158] “Oliver Heaviside and the coaxial cable,” Waveform. Accessed: Oct. 24, 2024. [Online]. Available: <https://www.waveform.com/blogs/main/146807175-oliver-heaviside-and-the-coaxial-cable>
- [159] “HeavisideUKPatent1407.pdf.” Accessed: Oct. 24, 2024. [Online]. Available: <https://cdn.shopify.com/s/files/1/0358/5537/files/HeavisideUKPatent1407.pdf?3837882249986977540>
- [160] “What is radio frequency (RF, rf)? - Definition from WhatIs.com,” Networking. Accessed: Oct. 24, 2024. [Online]. Available: <https://www.techtarget.com/searchnetworking/definition/radio-frequency>
- [161] “RF cable and RF wireless transmission: how does it work?” Accessed: Oct. 24, 2024. [Online]. Available: <https://www.microcontrollertips.com/rf-wireless-rf-cable-transmission/>
- [162] “What is an optical module? Optical module wiki | FS Community,” Knowledge. Accessed: Oct. 24, 2024. [Online]. Available: <https://community.fs.com/article/what-is-an-optical-module.html>
- [163] G. Keiser, *Optical fiber communications*, 4. ed. New York, NY: McGraw-Hill, 2011.
- [164] “The Advantages and Disadvantages of Optical Fiber | FS Community,” Knowledge. Accessed: Nov. 09, 2024. [Online]. Available: <https://community.fs.com/article/the-advantages-and-disadvantages-of-optical-fibers.html>
- [165] M. Uysal and H. Nouri, “Optical wireless communications — An emerging technology,” in *2014 16th International Conference on Transparent Optical Networks (ICTON)*, Graz, Austria: IEEE, Jul. 2014, pp. 1–7. doi: 10.1109/ICTON.2014.6876267.
- [166] M. Kong *et al.*, “Toward self-powered and reliable visible light communication using amorphous silicon thin-film solar cells,” *Opt. Express*, vol. 27, no. 24, p. 34542, Nov. 2019, doi: 10.1364/OE.27.034542.
- [167] M. Kong *et al.*, “Survey of energy-autonomous solar cell receivers for satellite–air–ground–ocean optical wireless communication,” *Prog. Quantum Electron.*, vol. 74, p. 100300, Nov. 2020, doi: 10.1016/j.pquantelec.2020.100300.
- [168] “Laser Communications Relay Demonstration (LCRD) Overview - NASA.” Accessed: Nov. 16, 2024. [Online]. Available: <https://www.nasa.gov/directorates/stmd/tech-demo-missions-program/laser-communications-relay-demonstration-lcrd-overview/>
- [169] “Laser Communications: Empowering More Data Than Ever Before - NASA.” Accessed: Nov. 17, 2024. [Online]. Available: <https://www.nasa.gov/technology/laser-communications-empowering-more-data-than-ever-before/>
- [170] “Quick guide on laser beam divergence measurement.” Accessed: Nov. 17, 2024. [Online]. Available: <https://www.gentec-eo.com/blog/laser-beam-divergence-measurement>
- [171] D. R. Paschotta, “diffraction-limited beams.” Accessed: Nov. 17, 2024. [Online]. Available: https://www.rp-photonics.com/diffraction_limited_beams.html

- [172] D. R. Paschotta, “free-space optical communications.” Accessed: Nov. 18, 2024. [Online]. Available: https://www.rp-photonics.com/free_space_optical_communications.html
- [173] “Optical Transceivers.” Accessed: Oct. 24, 2024. [Online]. Available: <https://www.broadcom.com/products/fiber-optic-modules-components/networking/optical-transceivers>
- [174] “What is an SFP? All things optical transceivers - YouTube.” Accessed: Oct. 24, 2024. [Online]. Available: <https://www.youtube.com/watch?v=WqTNDUatOho>
- [175] “The Most Comprehensive Guide Of Optical Modules - Asterfusion Data Technologies.” Accessed: Oct. 24, 2024. [Online]. Available: <https://cloudswit.ch/blogs/the-most-comprehensive-guide-of-optical-modules/>
- [176] “Optical Transceivers.” Accessed: Oct. 24, 2024. [Online]. Available: <https://www.amphenol-cs.com/optics/transceivers.html>
- [177] “1.25 GBd MMF Transceiver with DMI for Gigabit Ethernet, SFP, MSA, AFBR-5718PZ.” Accessed: Oct. 24, 2024. [Online]. Available: <https://www.broadcom.com/products/fiber-optic-modules-components/networking/optical-transceivers/sfp/afbr-5718pz>
- [178] “Optical transceivers - turning data into light,” Smartoptics. Accessed: Oct. 24, 2024. [Online]. Available: <https://smartoptics.com/knowledgebank-post/optical-transceivers-turning-data-into-light/>
- [179] C. Bunce, “Transceiver Form Factors: The History,” Pro Optix. Accessed: Oct. 24, 2024. [Online]. Available: <https://www.prooptix.com/news/transceiver-form-factors/>
- [180] C. Bunce, “Fiber Optic Lasers: Understanding Lasers in Optical Transceivers,” Pro Optix. Accessed: Oct. 24, 2024. [Online]. Available: <https://www.prooptix.com/news/fiber-optic-lasers/>
- [181] “101 Series: LEDs vs. VCSELs,” Fluke Networks. Accessed: Oct. 24, 2024. [Online]. Available: <https://www.flukenetworks.com/blog/cabling-chronicles/101-series-leds-vs-vcseles>
- [182] “Deep Analysis on Optical Transceiver Module | FS Community,” Knowledge. Accessed: Oct. 24, 2024. [Online]. Available: <https://community.fs.com/article/deep-analysis-on-optical-transceiver-module.html>
- [183] “Optical Communication Band - FiberLabs Inc.,” FiberLabs Inc. - Specialty Fiber Amplifiers and Fluoride Fibers. Accessed: Oct. 24, 2024. [Online]. Available: <https://www.fiberlabs.com/glossary/optical-communication-band/>
- [184] “Coherent Modulation vs. PAM4 in 800G Optical Transmission | FS Community,” Knowledge. Accessed: Oct. 24, 2024. [Online]. Available: <https://community.fs.com/article/coherent-modulation-vs-pam4-in-800g-optical-transmission.html>
- [185] “1.1. NRZ Fundamentals,” Intel. Accessed: Oct. 24, 2024. [Online]. Available: <https://www.intel.com/content/www/us/en/docs/programmable/683852/current/nrz-fundamentals.html>
- [186] “NRZ Vs. PAM4 Modulation Techniques | FS Community,” Knowledge. Accessed: Oct. 24, 2024. [Online]. Available: <https://community.fs.com/article/nrz-vs-pam4-modulation-techniques.html>

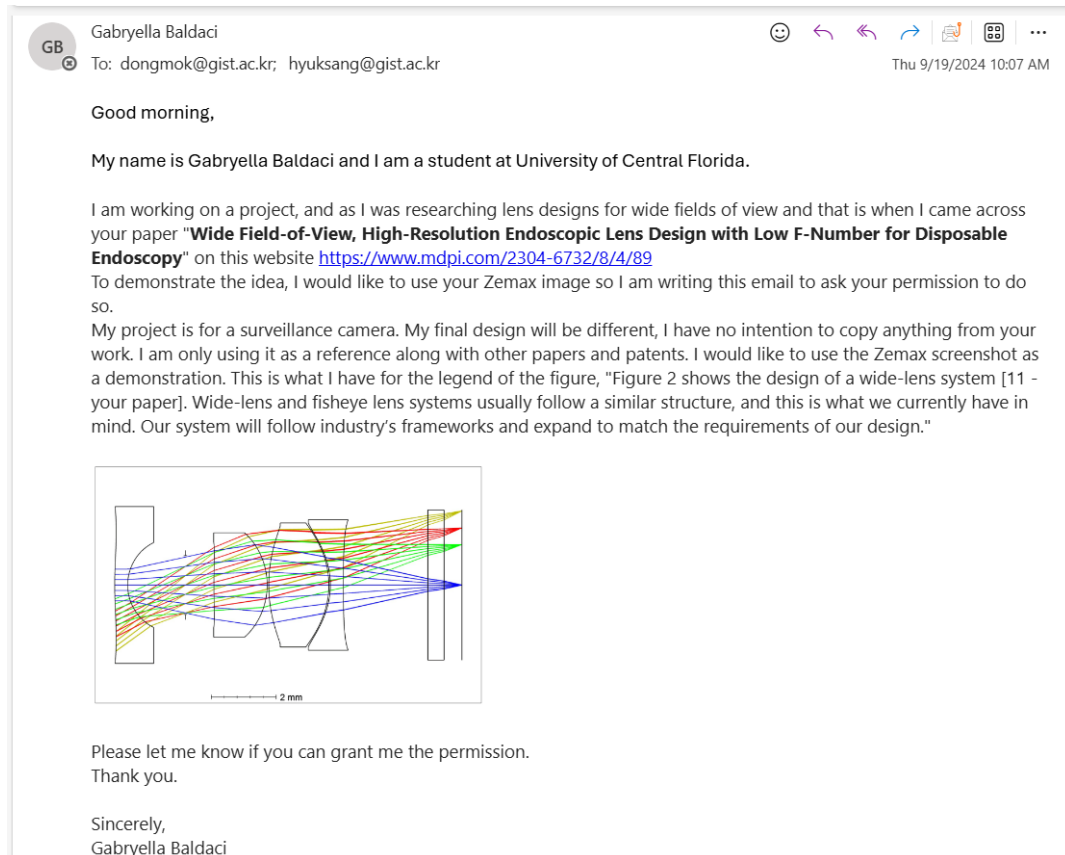
- [187] Z. Xia, “What Is QAM? How Does QAM Work?,” Huawei. Accessed: Oct. 24, 2024. [Online]. Available: <https://info.support.huawei.com/info-finder/encyclopedia/en/QAM.html>
- [188] “Coherent Modulation vs. PAM4 in 800G Optical Transmission | FS Community,” Knowledge. Accessed: Oct. 24, 2024. [Online]. Available: <https://community.fs.com/article/coherent-modulation-vs-pam4-in-800g-optical-transmission.html>
- [189] “Cisco Transceiver Modules - Use of Mode Conditioning Patch Cables in Gigabit Ethernet and 10 Gigabit Ethernet Laser-Based Transmissions,” Cisco. Accessed: Oct. 24, 2024. [Online]. Available: https://www.cisco.com/c/en/us/products/collateral/interfaces-modules/transceiver-modules/product_bulletin_c25-530836.html
- [190] “Laser Communications Relay Demonstration (LCRD) Overview - NASA.” Accessed: Nov. 18, 2024. [Online]. Available: <https://www.nasa.gov/directorates/stmd/tech-demo-missions-program/laser-communications-relay-demonstration-lcrd-overview/>
- [191] “Starlink | Technology,” Starlink. Accessed: Nov. 18, 2024. [Online]. Available: <https://www.starlink.com/us/technology>
- [192] Taylor Scully, “How to Choose a Mean Well Power Supply,” How to Choose a Mean Well Power Supply: LED Driver and Power Supply Guide. [Online]. Available: <https://www.ledsupply.com/blog/meanwell-power-supply-guide/>
- [193] Mouser Electronics, “LRS-50-5.” [Online]. Available: <https://www.mouser.com/ProductDetail/MEAN-WELL/LRS-50-5?qs=vDxCgdWo2h9hVrbBDwCVQA%3D%3D>
- [194] Mean Well, “LRS-50-5 Datasheet.” [Online]. Available: https://www.farnell.com/datasheets/2547982.pdf?_gl=1*1qk8ied*_gcl_au*MjM2NzY4MDI3LjE3Mjk1NjkwMjA
- [195] Mean Well, “RS-15-5.” [Online]. Available: <https://us.rs-online.com/product/mean-well/rs-15-5/70069559/>
- [196] RS Pro, “RS Pro 5V DC Power Supply.” [Online]. Available: <https://us.rs-online.com/product/rs-pro/1618272/71291688/>
- [197] CUI Inc., “CUI SWI5-N Datasheet.” Sep. 27, 2023. [Online]. Available: <https://www.cui.com/product/resource/swi5-n.pdf>
- [198] Alfred Covino, “What is Differential Impedance?” [Online]. Available: <https://blog.picamfg.com/what-is-differential-impedance>
- [199] Zachariah Peterson, “How to Design to a Differential Impedance Specification.” [Online]. Available: <https://resources.altium.com/p/how-design-differential-impedance-specification>
- [200] Lynsy Woodahl, “Ethernet PHY PCB Design Layout Checklist,” Application Report, Jun. 2021. [Online]. Available: Ethernet PHY PCB Design Layout Checklist
- [201] “HDMI Layout Guidline.” [Online]. Available: <https://ez.analog.com/video/w/documents/687/hdmi-layout-guideline>
- [202] Poulomi Ghosh, “Applying IPC-2221 Standards in Circuit Board Design.” [Online]. Available: <https://www.protoexpress.com/blog/ipc-2221-circuit-board-design/>

- [203] “EN IEC 62368 Testing,” D.L.S. Electronic Systems, Inc. Accessed: Oct. 24, 2024. [Online]. Available: <https://www.dlsemc.com/en-62368-testing/>
- [204] CUI Inc., “Efficiency Standards for External Power Supplies,” Standards. [Online]. Available: <https://www.cui.com/efficiency-standards>
- [205] Alex Deng, “Technical Compliance Statement CE EMC Test Report,” Mean Well, Test Report C1M2106134. [Online]. Available: <https://www.meanwell.com/Upload/PDF/UHP-1500/UHP-1500-EMC-cer.pdf>
- [206] “Quality | Edmund Optics.” Accessed: Oct. 19, 2024. [Online]. Available: <https://www.edmundoptics.com/capabilities/quality/>
- [207] “ISO 9001:2015,” ISO. Accessed: Oct. 19, 2024. [Online]. Available: <https://www.iso.org/standard/62085.html>
- [208] J. Lane, “Tutorial on ISO 10110 Optical Drawing Standard OPTI 521 – Intro to Opto-Mechanical Engineering.” The University of Arizona, Dec. 13, 2009. [Online]. Available: <https://wp.optics.arizona.edu/optomech/wp-content/uploads/sites/53/2016/10/Tutorial-on-ISO-10110-Optical-Drawing-Standards.pdf>
- [209] “Custom Optics Manufacturing and OEM Ordering.” Accessed: Oct. 19, 2024. [Online]. Available: <https://www.thorlabs.com>
- [210] “IEEE Standards Association,” IEEE Standards Association. Accessed: Nov. 19, 2024. [Online]. Available: <https://standards.ieee.org/ieee/802.3/7071/>
- [211] “Introduction to Optical Transceiver MSA Standards | FS Community,” Knowledge. Accessed: Oct. 24, 2024. [Online]. Available: <https://community.fs.com/article/how-much-do-you-know-about-msa-standard.html>
- [212] “Security & Surveillance,” Assent Risk Management. Accessed: Oct. 20, 2024. [Online]. Available: <https://www.assentriskmanagement.co.uk/industries/iso-certification-for-security-surveillance-organisations/>
- [213] “ISO 27001 Annex A Controls Archives,” High Table. Accessed: Oct. 20, 2024. [Online]. Available: <https://hightable.io/category/iso27001-annex-a-controls/>
- [214] S. Barker, “The complete guide to ISO/IEC 27002:2022,” High Table. Accessed: Oct. 20, 2024. [Online]. Available: <https://hightable.io/the-ultimate-guide-to-iso-27002-changes-2022/>
- [215] “ISO 22301:2019,” ISO. Accessed: Oct. 20, 2024. [Online]. Available: <https://www.iso.org/standard/75106.html>
- [216] “IEC 62676-4 Ed. 1.0 b:2014 - Video surveillance systems for use in security applications - Part 4: Application guidelines.” Accessed: Oct. 20, 2024. [Online]. Available: <https://webstore.ansi.org/standards/iec/iec62676ed2014>
- [217] “Is It Legal for Security Cameras to Record Audio?,” Brinks HomeTM. Accessed: Oct. 20, 2024. [Online]. Available: <https://brinkshome.com/smartcenter/are-security-cameras-allowed-to-record-audio>
- [218] R. Edwards, “Security Camera Laws, Rights, and Rules,” SafeWise. Accessed: Oct. 20, 2024. [Online]. Available: <https://www.safewise.com/security-camera-laws/>
- [219] R. E. Fischer, B. Tadic-Galeb, and P. R. Yoder, *Optical System Design*, 2nd ed. McGraw Hill, 2008.
- [220] I. I. Kim, B. McArthur, and E. J. Korevaar, “Comparison of laser beam propagation at 785 nm and 1550 nm in fog and haze for optical wireless

- communications,” presented at the Information Technologies 2000, E. J. Korevaar, Ed., Boston, MA, Feb. 2001, pp. 26–37. doi: 10.1117/12.417512.
- [221] “Triplet Fiber Optic Collimators/Couplers.” Accessed: Nov. 23, 2024. [Online]. Available: <https://www.thorlabs.com>
- [222] “How to compress video using FFmpeg,” Shotstack. Accessed: Nov. 24, 2024. [Online]. Available: <https://shotstack.io/learn/compress-video-ffmpeg/>

Appendix B – Copyright Permission

Most of the figures and tables in this document were made by the authors themselves. Permission was requested to use Figure 2.3



Appendix C – Data Sheet

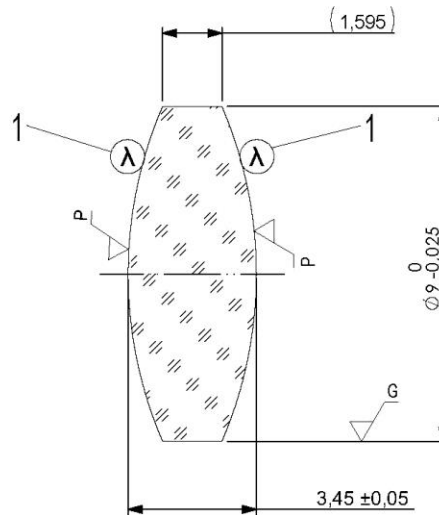
1. Camera Module: Raspberry Pi Camera Module 3 Specification

Sensor:	Sony IMX708
Resolution:	11.9 megapixels
Sensor size:	7.4mm sensor diagonal
Pixel size:	1.4µm × 1.4µm
Horizontal/vertical:	4608 × 2592 pixels
Common video modes:	1080p50, 720p100, 480p120
Output:	RAW10
IR cut filter:	Integrated in standard variants; not present in NoIR variants
Autofocus system:	Phase Detection Autofocus
Dimensions:	25 × 24 × 11.5mm (12.4mm height for Wide variants)
Ribbon cable length:	200mm
Cable connector:	15 × 1mm FPC
Operating temperature:	0°C to 50°C
Compliance:	FCC 47 CFR Part 15, Subpart B, Class B Digital Device Electromagnetic Compatibility Directive (EMC) 2014/30/EU Restriction of Hazardous Substances (RoHS) Directive 2011/65/EU
Production lifetime:	Raspberry Pi Camera Module 3 will remain in production until at least January 2030

Variants

	Camera Module 3	Camera Module 3 NoIR	Camera Module 3 Wide	Camera Module 3 Wide NoIR
Focus range	10cm-∞	10cm-∞	5cm-∞	5cm-∞
Focal length	4.74mm	4.74mm	2.75mm	2.75mm
Diagonal field of view	75 degrees	75 degrees	120 degrees	120 degrees
Horizontal field of view	66 degrees	66 degrees	102 degrees	102 degrees
Vertical field of view	41 degrees	41 degrees	67 degrees	67 degrees
Focal ratio (F-stop)	F1.8	F1.8	F2.2	F2.2
Infrared-sensitive	No	Yes	No	Yes

Appendix D – ISO 10110 Drawing



FOCAL LENGTH (EFL): $9 \pm 0,09$
 BACK FOCAL LENGTH (BFL): 7,9

1 AR COATING:

$\frac{1}{4}$ WAVE MgF₂ @ 550nm
 R(AVG) < 1,75% FROM 400-700nm (N-BK7)

LEFT SURFACE		MATERIAL	RIGHT SURFACE	
R	11,38±0,011 CX	GLASS N-SF5 673/322	R	11,38±0,011 CX
Øe	8,1		Øe	8,1
3/	3 (0,5) λ=632,8 nm		3/	3 (0,5) λ=632,8 nm
4/	-		4/	4,46'
5/	40 - 20 (MIL-PRF-13830B)		5/	40 - 20 (MIL-PRF-13830B)
6/	-		6/	-
PROTECTIVE CHAMFERS AS NEEDED		FIRST ANGLE PROJECTION	 Edmund Optics®	
ROHS: COMPLIANT	SCALE: NONE λ = 587,6nm			
SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE. DIMENSIONS ARE FOR REFERENCE ONLY			TITLE	
FOR INFORMATION ONLY: DO NOT MANUFACTURE PARTS TO THIS DRAWING			9mm Dia, x 9mm FL, MgF2 Coated, Double-Convex Lens	
INDICATIONS IN ACCORDANCE WITH ISO 10110		ALL DIMS IN: mm	DWG NO	32019
			SHEET SIZE A4	SHEET 1 OF 1

Appendix E – Software Code

Haar Cascade Code

```
1 import cv2 as cv
2
3 path = cv.data.harcascades + 'haarcascade_fullbody.xml'
4
5 person_rec = cv.CascadeClassifier(path)
6
7 frame = cv.imread('C:\\Users\\Death\\OneDrive\\Documents\\image.jpg')
8
9
10
11 def person():
12     gray = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)
13
14     gray = cv.equalizeHist(gray)
15
16     gray = cv.GaussianBlur(gray, (5,5), 0)
17
18     bodies = person_rec.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=1, minSize=(20,20), flags=cv.CASCADE_SCALE_IMAGE)
19
20     for (x,y,w,h) in bodies:
21
22         cv.rectangle(frame, (x,y), (x+w,y+h), (0,255,0), 2)
23         font=cv.FONT_HERSHEY_COMPLEX
24         cv.putText(frame, 'Person', (x+6,y-6), font, .5, (0,255,0), 1)
25
26 person()
27
28 cv.imshow('Body Rec', frame)
29 cv.waitKey(0)
```

HOG Code

```
1 import cv2 as cv
2
3 hog = cv.HOGDescriptor()
4 hog.setSVMDetector(cv.HOGDescriptor_getDefaultPeopleDetector())
5
6 path = 'C:\\Users\\Death\\OneDrive\\Documents\\image.jpg'
7 frame = cv.imread(path)
8
9 frame = cv.resize(frame, (650, 500))
10
11 frame = cv.GaussianBlur(frame, (5,5), 0)
12
13 bodies, weights = hog.detectMultiScale(frame, winStride=(5,5), padding=(8,8), scale=1.02)
14
15 for (x,y,w,h) in bodies:
16     cv.rectangle(frame, (x,y), (x+w,y+h), (0,255,0), 2)
17     font = cv.FONT_HERSHEY_COMPLEX
18     cv.putText(frame, 'Person', (x-6,y-6), font, .5, (0,255,0), 1)
19
20 cv.imshow('Person', frame)
21 cv.waitKey()
22
```

YOLOv11 Code

```

1  import cv2 as cv
2  from ultralytics import YOLO
3
4  model = YOLO('yolo11n.pt')
5
6  path = 'C:\\Users\\Death\\OneDrive\\Documents\\image.jpg'
7  frame = cv.imread(path)
8
9  result = model(frame)
10
11 for results in result:
12     for box in results.boxes:
13         x1,y1,x2,y2 = box.xyxy[0].tolist()
14         conf = box.conf[0].item()
15         class_id = int(box.cls[0].item())
16
17         if conf > 0.5:
18             label = f'{model.names[class_id]}: {conf:.2f}'
19             cv.rectangle(frame, (int(x1), int(y1)), (int(x2), int(y2)), (0, 255, 0),)
20             cv.putText(frame, label, (int(x1) + 10, int(y1) - 5), cv.FONT_HERSHEY_COMPLEX, 0.5, (0, 255, 0), 1)
21
22 cv.imshow('Person', frame)
23 cv.waitKey()

```

YOLOv10 Code

```

25 import cv2 as cv
26 from ultralytics import YOLO
27
28 model = YOLO('yolov10n.pt')
29
30 path = 'C:\\Users\\Death\\OneDrive\\Documents\\image.jpg'
31 frame = cv.imread(path)
32
33 result = model(frame)
34
35 for results in result:
36     for box in results.boxes:
37         x1,y1,x2,y2 = box.xyxy[0].tolist()
38         conf = box.conf[0].item()
39         class_id = int(box.cls[0].item())
40
41         if conf > 0.5:
42             label = f'{model.names[class_id]}: {conf:.2f}'
43             cv.rectangle(frame, (int(x1), int(y1)), (int(x2), int(y2)), (0, 255, 0),)
44             cv.putText(frame, label, (int(x1) + 10, int(y1) - 5), cv.FONT_HERSHEY_COMPLEX, 0.5, (0, 255, 0), 1)
45
46 cv.imshow('Person', frame)
47 cv.waitKey()

```

Computer Vision Demo Code

```

1  import cv2 as cv
2  import numpy as npy
3  #import cvzone
4  from picamera2 import Picamera2
5  from ultralytics import YOLO
6
7  model = YOLO('yolo11n.pt')
8  names = model.names
9  model.classes = [0] #Detects people only
10
11  picam2 = Picamera2
12  picam2.preview_configuration.main.size = (1920, 1080)
13  picam2.preview_configuration.main.format = "RGB888"
14  picam2.preview_configuration.align()
15  picam2.configure("preview")
16  picam2.start()
17
18  #vid = cv.VideoCapture(0)
19
20  #Frame counter
21  # cnt = 0
22
23  while True: #if True then loop will run
24      #detected, frame = vid.read()
25      # cnt += 1 #Increment frame count
26
27
28
29
30
31
32  frame = picam2.capture_array()
33
34
35  gray = cv.cvtColor(frame, cv.COLOR_RGB2GRAY)
36  gray = cv.cvtColor(gray, cv.COLOR_GRAY2BGR)
37
38  #Set result to run YOLO11 tracking method to track objects in frames, persisting tracking among frames
39  # iou = intersection over union (for intersection of detection boxes)
40  result = model.track(frame, persist=True, iou = .3)
41
42  #Person counter
43  person_cnt = 0
44
45  #if no detection boxes and no tracking ids for the detection boxes is found
46  if result[0].boxes is not None and result[0].boxes.id is not None:
47      #Gets boxes, person ID dependent on class and confidence level
48      boxed = result[0].boxes.xyxy.int().cpu().tolist()
49      person_id = result[0].boxes.cls.int().cpu().tolist()
50      confidence = result[0].boxes.conf.cpu().tolist()
51
52  #Plotting tracks
53  for box, person, conf in zip(boxed, person_id, confidence):
54      person_name = names[person] #set person names to the class name (person)
55      if person_name == 'person': #if person name is the same as the class name
56          person_cnt += 1 #Increment person counter
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76

```

```

57      x,y,x1,y1 = box
58      cv.rectangle(gray, (x,y), (x1,y1), (0, 0, 255), 2) #Bounding box (detection box)
59      cv.putText(gray, f'Person,Conf: {conf:.2f}', (x,y - 10), 1, 1, (0, 0, 255), 1) #Label
60
61  #Label for total count of people detected on frames
62  cv.putText(gray, f'Total Count: {person_cnt}', (10, 50), 1, 1, (0, 0, 255), 1)
63
64  #implement grayscaling
65
66  #Displays result
67  cv.imshow('Person', gray)
68
69  #Break when 'Esc' key pressed
70  if cv.waitKey(5) & 0xFF == 27:
71      break
72
73  #vid.release() #Releases video
74  picam2.stop()
75  cv.destroyAllWindows() #Closes windows
76

```

Computer Vision 1st Iteration

```

1  import cv2 as cv
2  import numpy as npy
3  #import cvzone
4  #from picamera2 import Picamera2
5  from ultralytics import YOLO
6
7  model = YOLO('yolo11n.pt')
8  names = model.names
9  model.classes = [0] #Detects people only
10
11  vid = cv.VideoCapture(0)
12
13  #Frame counter
14  # cnt = 0
15
16  while True: #if True then loop will run
17      detected, frame = vid.read()
18      # cnt += 1 #Increment frame count
19      # if cnt % 2 != 0: #Skip every other frame
20      #     continue

```

```

30
31  #Checks if video is detected if not breaks out of loop
32  if not detected:
33      break
34
35  #frame = picam2.capture_array()
36
37  #gray = cv.cvtColor(frame,cv.COLOR_RGB2GRAY)
38  #gray = cv.cvtColor(gray,cv.COLOR_GRAY2BGR)
39
40  #Set result to run YOLO11 tracking method to track objects in frames, persisting tracking among frames
41  # iou = intersection over union (for intersection of detection boxes)
42  result = model.track(frame, persist=True, iou = .3)
43
44  #Person counter
45  person_cnt = 0
46
47  #if no detection boxes and no tracking ids for the detection boxes is found
48  if result[0].boxes is not None and result[0].boxes.id is not None:
49      #Gets boxes, person ID dependent on class and confidence level
50      boxed = result[0].boxes.xyxy.int().cpu().tolist()
51      person_id = result[0].boxes.cls.int().cpu().tolist()
52      confidence = result[0].boxes.conf.cpu().tolist()
53
54  #Plotting tracks

```

```

55     for box, person, conf in zip(boxed, person_id, confidence):
56         person_name = names[person] #set person names to the class name (person)
57         if person_name == 'person': #if person name is the same as the class name
58             person_cnt += 1 #Increment person counter
59             x,y,x1,y1 = box
60             cv.rectangle(frame, (x,y), (x1,y1), (0, 0, 255), 2) #Bounding box (detection box)
61             cv.putText(frame, f'Person,Conf: {conf:.2f}', (x,y - 10), 1, 1, (0, 0, 255), 1) #Label
62
63     #Label for total count of people detected on frames
64     cv.putText(frame, f'Total Count: {person_cnt}', (10, 50), 1, 1, (0, 0, 255), 1)
65
66     #implement grayscale
67
68     #Displays result
69     cv.imshow('Person', frame)
70
71     #Break when 'Esc' key pressed
72     if cv.waitKey(5) & 0xFF == 27:
73         break
74
75     vid.release() #Releases video
76     #picam2.stop()
77     cv.destroyAllWindows() #Closes windows

```

Computer Vision 2nd Iteration

```

1  import cv2 as cv
2  import numpy as npy
3  #import cvzone
4  #from picamera2 import Picamera2
5  from ultralytics import YOLO
6
7  model = YOLO('yolo11n.pt')
8  names = model.names
9  model.classes = [0] #Detects people only
10
11  vid = cv.VideoCapture(0)
12
13  #Frame counter
14  # cnt = 0
15
16  while True: #if True then loop will run
17      detected, frame = vid.read()
18      # cnt += 1 #Increment frame count
19      # if cnt % 2 != 0: #Skip every other frame
20      #     continue
21
22      #Checks if video is detected if not breaks out of loop
23      if not detected:
24          break
25

```

```

26  #frame = picam2.capture_array()
27
28  gray = cv.cvtColor(frame,cv.COLOR_RGB2GRAY)
29  gray = cv.cvtColor(gray,cv.COLOR_GRAY2BGR)
30
31  #Set result to run YOLO11 tracking method to track objects in frames, persisting tracking among frames
32  # iou = intersection over union (for intersection of detection boxes)
33  result = model.track(frame, persist=True, iou = .3)
34
35  #Person counter
36  person_cnt = 0
37
38  #if no detection boxes and no tracking ids for the detection boxes is found
39  if result[0].boxes is not None and result[0].boxes.id is not None:
40      #Gets boxes, person ID dependent on class and confidence level
41      boxed = result[0].boxes.xyxy.int().cpu().tolist()
42      person_id = result[0].boxes.cls.int().cpu().tolist()
43      confidence = result[0].boxes.conf.cpu().tolist()
44
45  #Plotting tracks
46  for box, person, conf in zip(boxed, person_id, confidence):
47      person_name = names[person] #set person names to the class name (person)
48      if person_name == 'person': #if person name is the same as the class name
49          person_cnt += 1 #Increment person counter
50      x,y,x1,y1 = box

```

```

51         cv.rectangle(gray, (x,y), (x1,y1), (0, 0, 255), 2) #Bounding box (detection box)
52         cv.putText(gray, f'Person,Conf: {conf:.2f}', (x,y - 10), 1, 1, (0, 0, 255), 1) #Label
53
54         #Label for total count of people detected on frames
55         cv.putText(gray, f'Total Count: {person_cnt}', (10, 50), 1, 1, (0, 0, 255), 1)
56
57         #implement grayscaling
58
59         #Displays result
60         cv.imshow('Person', frame)
61
62         #Break when 'Esc' key pressed
63         if cv.waitKey(5) & 0xFF == 27:
64             break
65
66     vid.release() #Releases video
67     #picam2.stop()
68     cv.destroyAllWindows() #Closes windows

```

Computer Vision Raspberry Pi 5 Code

```

1  import cv2 as cv
2  import numpy as npy
3  #import cvzone
4  from picamera2 import Picamera2
5  from ultralytics import YOLO
6
7  model = YOLO(['yolo11n.pt'])
8  names = model.names
9  model.classes = [0] #Detects people only
10
11  picam2 = Picamera2
12  picam2.preview_configuration.main.size = (1920, 1080)
13  picam2.preview_configuration.main.format = "RGB888"
14  picam2.preview_configuration.align()
15  picam2.configure("preview")
16  picam2.start()
17
18  #vid = cv.VideoCapture(0)
19
20  #Frame counter
21  # cnt = 0
22
23  while True: #if True then loop will run
24      #detected, frame = vid.read()
25      # cnt += 1 #Increment frame count

```

```

32      frame = picam2.capture_array()
33
34      gray = cv.cvtColor(frame, cv.COLOR_RGB2GRAY)
35      gray = cv.cvtColor(gray, cv.COLOR_GRAY2BGR)
36
37      #Set result to run YOLO11 tracking method to track objects in frames, persisting tracking among frames
38      # iou = intersection over union (for intersection of detection boxes)
39      result = model.track(frame, persist=True, iou = .3)
40
41      #Person counter
42      person_cnt = 0
43
44      #if no detection boxes and no tracking ids for the detection boxes is found
45      if result[0].boxes is not None and result[0].boxes.id is not None:
46          #Gets boxes, person ID dependent on class and confidence level
47          boxed = result[0].boxes.xyxy.int().cpu().tolist()
48          person_id = result[0].boxes.cls.int().cpu().tolist()
49          confidence = result[0].boxes.conf.cpu().tolist()
50
51      #Plotting tracks
52      for box, person, conf in zip(boxed, person_id, confidence):
53          person_name = names[person] #set person names to the class name (person)
54          if person_name == 'person': #if person name is the same as the class name
55              person_cnt += 1 #Increment person counter

```

```

57         x,y,x1,y1 = box
58         cv.rectangle(gray, (x,y), (x1,y1), (0, 0, 255), 2) #Bounding box (detection box)
59         cv.putText(gray, f'Person,Conf: {conf:.2f}', (x,y - 10), 1, 1, (0, 0, 255), 1) #Label
60
61     #Label for total count of people detected on frames
62     cv.putText(gray, f'Total Count: {person_cnt}', (10, 50), 1, 1, (0, 0, 255), 1)
63
64     #implement grayscaling
65
66     #Displays result
67     cv.imshow('Person', gray)
68
69     #Break when 'Esc' key pressed
70     if cv.waitKey(5) & 0xFF == 27:
71         break
72
73 #vid.release() #Releases video
74 picam2.stop()
75 cv.destroyAllWindows() #Closes windows
76

```

Video Compression and Decompression Demo Code

```

Testpy > ...
1 import subprocess
2 import os
3 |
4 # variable to store name of file
5 input_file = "Video_Demo.mov"
6 # variable to store name of output file
7 output_file = "output.mov"
8 # variable to store name of decompressed output file
9 decompressed_file = "decompressed.mov"
10 # command for ffmpeg
11 # ffmpeg -i input_file -c:v libx264 -pix_fmt yuv420p -crf 23 -vf scale=1280:720 -r 30 output_file
12
13 # command used to run ffmpeg for compression
14 command = [
15     "ffmpeg",
16     "-i", input_file,           # takes input file
17     "-c:v", "libx264",          # uses H.264 codec
18     "-pix_fmt", "yuv420p",      # command to convert iPhone prores to H.264 format
19     "-crf", "23",               # lower value generates high quality but lower compression
20                                # higher value generates lower quality video but higher compression
21     "-preset", "ultrafast",     # speed of compression, faster compression equals larger file
22     "-vf", f"scale=1280:720",   # converts file to different resolution
23     "-r", "30",                 # changes frame rate to 30 fps
24     output_file                 # creates new video file after compression
25 ]
26
27 command2 = [
28     "ffmpeg",
29     "-i", output_file,          # takes in output file
30     "-c:v", "libx264",          # uses H.264 codec
31     "-pix_fmt", "yuv420p",      # command to convert iPhone prores to H.264 format
32     "-crf", "0",               # lower value generates high quality but lower compression
33                                # higher value generates lower quality video but higher compression
34     "-preset", "ultrafast",     # speed of compression, faster compression equals larger file
35     "-vf", f"scale=1980:1080",  # converts file to different resolution
36     "-r", "60",                 # changes frame rate to 60 fps
37     decompressed_file           # creates new video file after compression
38 ]
39
40
41 # if statement checks to see if file is present
42 if os.path.isfile(input_file):
43     try:
44         # runs ffmpeg command
45         subprocess.run(command, check=True)
46         print(f"Compression successful. Output file saved as {output_file}")
47     except subprocess.CalledProcessError as e:

```

```

Test.py > -
27 Command2 = [
28     "ffmpeg",
29     "-i", output_file,          # takes in output file
30     "-c:v", "libx264",         # uses H.264 codec
31     "-pix_fmt", "yuv420p",     # command to convert iPhone prores to H.264 format
32     "-crf", "0",               # lower value generates high quality but lower compression
33                               # higher value generates lower quality video but higher compression
34                               # setting this to 0 means lossless compression
35     "-preset", "ultrafast",    # speed of compression, faster compression equals larger file
36     "-vf", f"scale=1980:1080", # converts file to different resolution
37     "-r", "60",               # changes frame rate to 60 fps
38     decompressed_file         # creates new video file after compression
39 ]
40
41 # if statement checks to see if file is present
42 if os.path.isfile(input_file):
43     try:
44         # runs ffmpeg command
45         subprocess.run(command, check=True)
46         print(f"Compression successful. Output file saved as {output_file}")
47     except subprocess.CalledProcessError as e:
48         # exception if compression unsuccessful
49         print(f"Error occurred during compression: {e}")
50 else:
51     # prints if file not found
52     print(f"Error: The file '{input_file}' does not exist.")
53
54 # if statement checks to see if file is present
55 if os.path.isfile(output_file):
56     try:
57         # runs ffmpeg command to decompress
58         subprocess.run(command2, check=True)
59         print(f"Decompression successful. Output file saved as {decompressed_file}")
60     except subprocess.CalledProcessError as e:
61         # exception if decompression unsuccessful
62         print(f"Error occurred during decompression: {e}")
63 else:
64     # prints if file not found
65     print(f"Error: The file '{output_file}' does not exist.")

```

C Code flashed to PCB

```
1  #include <SPI.h>
2  #include <Ethernet.h>
3  #include <EthernetUdp.h>
4
5  #define ETH_MISO 14
6  #define ETH_MOSI 1
7  #define ETH_SCLK 21
8  #define ETH_CS 39
9
10 byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
11
12 // Use static IP for link-local communication
13 IPAddress ip(169, 254, 74, 47);
14 unsigned int localPort = 8090;
15
16 EthernetUDP Udp;
17
18 void setup() {
19     Serial.begin(921600); // USB serial output to PC
20     while (!Serial);
21
22     // Serial.println("Initializing Ethernet...");
23
24     SPI.begin(ETH_SCLK, ETH_MISO, ETH_MOSI, ETH_CS);
25     Ethernet.init(ETH_CS);
26     Ethernet.begin(mac, ip);
27
28     if (Udp.begin(localPort)) {
29         // Serial.print("Listening for UDP on port ");
30         // Serial.println(localPort);
31     } else {
32         // Serial.println("Failed to start UDP listener.");
33     }
34 }
35
36 void loop() {
37     int packetSize = Udp.parsePacket();
38     if (packetSize) {
39         uint8_t packetBuffer[packetSize];
40         int len = Udp.read(packetBuffer, sizeof(packetBuffer));
41         if (len > 0) {
42             Serial.write(packetBuffer, len); // Send raw MPEG-TS to PC via USB
43         }
44     }
45 }
46
```

C Code used to pipeline into FFmpeg

```
#include <windows.h>
#include <stdio.h>

int main() {
    HANDLE hSerial, hPipeWrite;
    DCB dcbSerialParams = {0};
    COMMTIMEOUTS timeouts = {0};
    DWORD bytesRead;
    char buffer[4096];

    SECURITY_ATTRIBUTES saAttr = { sizeof(SECURITY_ATTRIBUTES), NULL, TRUE };
    HANDLE hPipeRead;

    // Create a pipe for ffmpeg's stdin
    if (!CreatePipe(&hPipeRead, &hPipeWrite, &saAttr, 0)) {
        fprintf(stderr, "Failed to create pipe\n");
        return 1;
    }

    // Prepare ffmpeg process
    STARTUPINFO si = { sizeof(STARTUPINFO) };
    PROCESS_INFORMATION pi;
    si.dwFlags |= STARTF_USESTDHANDLES;
    si.hStdInput = hPipeRead;
    si.hStdOutput = GetStdHandle(STD_OUTPUT_HANDLE);
    si.hStdError = GetStdHandle(STD_ERROR_HANDLE);

    // ffmpeg command
    char cmd[] = "ffmpeg -fflags nobuffer -flags low_delay -f mpegts -";

    if (!CreateProcessA(NULL, cmd, NULL, NULL, TRUE, 0, NULL, NULL, &si, &pi)) {
        fprintf(stderr, "Failed to start ffmpeg\n");
        return 1;
    }

    // Close the read end of the pipe in our process
    CloseHandle(hPipeRead);

    // Open COM3
    hSerial = CreateFile("\\\\.\\COM3", GENERIC_READ, 0, 0,
        OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);
    if (hSerial == INVALID_HANDLE_VALUE) {
        fprintf(stderr, "Error opening COM3\n");
        return 1;
    }
}
```

```

dcbSerialParams.DCBlength = sizeof(dcbSerialParams);
GetCommState(hSerial, &dcbSerialParams);

// Set the baud rate to 921600
dcbSerialParams.BaudRate = 921600;
dcbSerialParams.ByteSize = 8; // 8 data bits
dcbSerialParams.StopBits = ONESTOPBIT; // 1 stop bit
dcbSerialParams.Parity = NOPARITY; // No parity
SetCommState(hSerial, &dcbSerialParams);

// Set timeouts for serial communication
timeouts.ReadIntervalTimeout = 1;
timeouts.ReadTotalTimeoutConstant = 1;
timeouts.ReadTotalTimeoutMultiplier = 1;
SetCommTimeouts(hSerial, &timeouts);

// Read from serial and write to ffmpeg's stdin
while (1) {
    if (ReadFile(hSerial, buffer, sizeof(buffer), &bytesRead, NULL)) {
        if (bytesRead > 0) {
            DWORD bytesWritten;
            WriteFile(hPipeWrite, buffer, bytesRead, &bytesWritten, NULL);
        }
    }
}

CloseHandle(hPipeWrite);
CloseHandle(hSerial);
return 0;
}

```

Final Code

```

1 import cv2 as cv
2 from picamera2 import Picamera2
3 import numpy as npy
4 from ultralytics import YOLO
5 #import datetime
6 import subprocess
7
8 # Load YOLO model
9 model = YOLO('yolo11n.pt')
10 names = model.names
11 model.classes = [0] # Detects people only
12
13 # Initialize the Picamera2
14 picam2 = Picamera2()
15 picam2.preview_configuration.main.size = (1280, 720) # Resolution used
16 picam2.preview_configuration.main.format = "RGB888"
17 picam2.preview_configuration.align()
18 picam2.configure("preview")
19 picam2.start()
20
21 # Start FFmpeg process to stream frames via UDP
22 ffmpeg_cmd = [
23     'ffmpeg',
24     '-f', 'rawvideo',
25     '-pix_fmt', 'bgr24',
26     '-s', '1280x720', # Must match picamera resolution
27     '-r', '30',
28     '-i', '-',
29     '-vcodec', 'libx264',

```

```

30     '-preset', 'ultrafast',
31     '-tune', 'zerolatency',
32     '-f', 'mpegs',
33     'udp://169.254.241.65:8090'
34 ]
35
36 ffmpeg_proc = subprocess.Popen(ffmpeg_cmd, stdin=subprocess.PIPE)
37
38 # Gets current date and time
39 #current_time = datetime.datetime.now()
40
41 while True:
42     # Capture real-time video frame
43     frame = picam2.capture_array()
44
45     # Convert to grayscale
46     gray = cv.cvtColor(frame, cv.COLOR_RGB2GRAY)
47
48     # Convert grayscale to BGR (overlay)
49     gray = cv.cvtColor(gray, cv.COLOR_GRAY2BGR)
50
51     # Run YOLO tracking
52     result = model.track(frame, persist=True, iou=.3)
53
54     # Person counter
55     person_cnt = 0
56
57     # Empty arrays for storing boxes, person_id and confidence level
58     boxed = []
59     person_id = []
60     confidence = []
61
62     # Extract detection results
63     if result[0].boxes is not None and result[0].boxes.id is not None:
64         boxed = result[0].boxes.xyxy.int().cpu().tolist()
65         person_id = result[0].boxes.cls.int().cpu().tolist()
66         confidence = result[0].boxes.conf.cpu().tolist()
67
68     # Draw bounding boxes and labels
69     for box, person, conf in zip(boxed, person_id, confidence):
70         person_name = names[person]
71         if person_name == 'person':
72             person_cnt += 1
73             x, y, x1, y1 = box
74             cv.rectangle(gray, (x, y), (x1, y1), (0, 0, 255), 2)
75             cv.putText(gray, f'Person,Conf: {conf:.2f}', (x, y - 10), 1, 1, (0, 0, 255), 2)
76
77     # Total count label
78     cv.putText(gray, f'Total Count: {person_cnt}', (5, 30), 1, 2, (0, 0, 255), 2)
79
80     # Timestamp label
81     #cv.putText(gray, f'{current_time}', (720, 50), 1, 2, (0, 0, 255), 2)
82
83     # Show locally (optional)
84     cv.imshow('Person', gray)

```

```

85
86     # Stream frame to Ffmpeg
87     try:
88         ffmpeg_proc.stdin.write(gray.tobytes())
89     except BrokenPipeError:
90         print("Ffmpeg pipe closed. Exiting.")
91         break
92
93     # Exit on 'Esc' key
94     if cv.waitKey(5) & 0xFF == 27:
95         break
96
97     # Cleanup
98     picam2.stop()
99     ffmpeg_proc.stdin.close()
100    ffmpeg_proc.wait()
101    cv.destroyAllWindows()
102

```

Appendix F – ChatGPT prompts and outcomes

Case Study 1:

Prompt 1

I am designing a surveillance camera with a wide field of view. This camera will be placed inside a well-lit place or used for daytime surveillance. I will design a lens system/assembly for it. However, before doing so I need to find a camera sensor that I will eventually mount the lens on it. What are some aspects I should take into account when choosing a sensor for a camera that will transmit live video to my surveillance system?

ChatGPT Outcome 1

When choosing a camera sensor for a surveillance camera that will transmit live video with a wide field of view in a well-lit or daytime environment, here are some key aspects to consider:

1. Sensor Size:

- The sensor size will directly influence the field of view and the depth of field. A larger sensor will capture more of the lens's image circle, providing a wider field of view for a given focal length.
- Common sensor sizes to consider:
 - 1/2.7-inch or larger (e.g., 1/1.8-inch) for high-quality imaging.

- Full-frame sensors (if budget and space allow) can provide excellent image quality and flexibility but are often more expensive and larger in physical size.

- APS-C sensors can offer a balance between size and cost, with a moderate crop factor (~1.5x).

2. Resolution:

- Choose a resolution that meets your surveillance needs while ensuring smooth video transmission.

- HD (1080p) or 4K resolution is common for modern surveillance cameras.

- 4K resolution offers more detail and is ideal for wide-area monitoring, but it requires more bandwidth for live transmission.

- 1080p may be sufficient if high detail isn't critical, and it requires less bandwidth.

3. Frame Rate:

- For smooth live video, aim for a frame rate of 30 fps or higher.

- If you're dealing with fast-moving objects, 60 fps or more can provide smoother motion capture.

- A higher frame rate increases the bandwidth requirements but results in better video quality for fast-moving subjects.

4. Dynamic Range:

- A sensor with a good dynamic range is important, especially if the camera will face varying lighting conditions (e.g., shadows, bright spots in the scene). Wide dynamic range (WDR) capabilities can handle such lighting variations, ensuring details in both bright and dark areas are captured well.

- In a well-lit environment, WDR can still be useful to prevent overexposure or underexposure in certain parts of the image.

5. Low-Light Performance:

- While your design is intended for daytime or well-lit environments, you should still consider a sensor with reasonable low-light performance in case lighting conditions vary or for occasional use in dimmer settings.

- Low-light sensors with large pixels (i.e., lower resolution for a given sensor size) capture more light, resulting in better image quality in low-light situations.

6. Shutter Type:

- Consider using a global shutter if your camera will be capturing fast-moving objects, as it eliminates the rolling-shutter effect (which can cause motion distortion).

- If motion artifacts are not a major concern, a rolling shutter may suffice, and sensors with rolling shutters tend to be more affordable.

7. Data Interface:

- Ensure the sensor supports a data interface that is compatible with your transmission system (e.g., MIPI CSI, USB, or Ethernet).

- Since this is for live video transmission, consider the bandwidth requirements. Higher resolutions and frame rates demand faster data interfaces.

8. Power Consumption:

- Low power consumption is critical if the camera is running continuously or if you have specific power constraints. Sensors with lower power consumption will generate less heat and are ideal for a system that operates 24/7.

9. Lens Compatibility:

- The sensor must be compatible with the lens system you're designing, particularly in terms of mount type (e.g., C-mount, M12), image circle, and field of view.

- Sensor size impacts how the lens will project the image, so consider both the focal length of your lens and the sensor size to achieve the required FOV.

10. Latency:

- For live video, low latency is crucial to ensure real-time video streaming. The sensor, along with the camera's processing pipeline, should support low-latency video output.

- Hardware video encoding (e.g., H.264 or H.265) directly on the sensor or module can reduce the delay in live transmission.

11. Compression and Bandwidth:

- Consider a sensor that supports onboard compression (such as H.264, H.265) to reduce the data bandwidth needed for live transmission. This helps with smoother video streams without sacrificing too much image quality.

12. Environmental Durability:

- Ensure the sensor is built to handle environmental factors, especially if the camera may be exposed to temperature fluctuations, dust, or vibrations in an indoor environment.

- A sensor that can function in a range of temperatures may be beneficial, especially for areas that can experience heat from lighting or equipment.

13. Cost:

- Sensor costs can vary significantly depending on their size, resolution, and additional features like WDR or low-light capabilities.

- Balancing cost and performance is key—ensure the sensor meets your performance requirements for field of view, resolution, and frame rate while staying within budget.

By considering these factors: size, resolution, frame rate, dynamic range, data interface, and power consumption—you can choose a sensor that not only fits your design needs but also ensures optimal performance for live video transmission in your surveillance system.

Copilot Outcome 1

When choosing a sensor for your surveillance camera, especially for a well-lit, daytime environment, there are several key aspects to consider:

Sensor Size and Resolution: Larger sensors generally capture more light, resulting in better image quality.

Field of View (FoV): Since you need a wide field of view, ensure the sensor can support lenses that offer a wide-angle perspective. [This is crucial for covering large areas with fewer cameras².](#)

Frame Rate: For live video transmission, a higher frame rate (e.g., 30 fps or higher) ensures smooth video playback, which is important for real-time monitoring¹.

Dynamic Range: A sensor with a good dynamic range can handle varying lighting conditions within the scene, ensuring that both bright and shadowed areas are captured clearly¹.

Lens Compatibility: Make sure the sensor is compatible with the lens system you plan to design. [The lens and sensor should be well-matched to maximize image quality¹.](#)

Connectivity and Compression: Consider how the sensor handles video compression and transmission. [Efficient compression \(e.g., H.264 or H.265\) reduces bandwidth usage without significantly compromising image quality³.](#)

Power Consumption: For continuous operation, choose a sensor with low power consumption to ensure reliability and reduce operational costs¹.

Cost and Availability: Finally, balance your budget with the sensor's features and availability. [Sometimes, the latest sensors might offer advanced features but at a higher cost¹.](#)

By considering these factors, you can select a sensor that meets your surveillance needs and integrates well with your lens design. If you have any specific requirements or need further assistance, feel free to ask!

Case Study 2:

Prompt 2

I want to build a lens system that will be mounted on a CMOS sensor. I want my camera to have at least 80 degree field of view, I want to minimize aberrations. Can you give me ideas for the design? How many lenses should I use? Which type of lenses should I use?

ChatGPT Outcome 2

To design a lens system with at least 80-degree field of view (FOV) for a camera mounted on a CMOS sensor, while minimizing aberrations, you need a multi-lens design. Here's a detailed guide on how to approach it:

Key Design Considerations:

1. **Wide Field of View (80 degrees):** Achieving a wide FOV requires short focal lengths relative to the sensor size, and the lens design must manage distortion and field curvature.
2. **Minimizing Aberrations:** To reduce spherical, chromatic, and coma aberrations, a multi-lens approach is necessary. You'll likely need at least 3-5 lens elements.
3. **CMOS Sensor Dimensions:** The sensor size determines the required focal length for the desired FOV. If your sensor size is 6.45mm x 3.43mm (diagonal 7.4mm), the focal length needs to be short, around 3-4mm for a 80-degree FOV.

Recommended Design Outline:

1. **Use a Retrofocus Design (Reverse Telephoto)**
 - For wide-angle lenses, a retrofocus design works well. This is a design where the back focal length is longer than the effective focal length, which is essential for maintaining a wide FOV while allowing space for the sensor and camera body.
 - This approach helps control aberrations, especially off-axis aberrations like coma and astigmatism.
2. **Number of Lenses: 3 to 5 Elements**

- Three-element system: This would be a minimal design where you can use two positive lenses and one negative lens to correct for spherical aberrations and chromatic aberrations. However, further optimization might be required for edge-to-edge sharpness.

- Four to five-element system: This will give you more control over aberrations, especially field curvature, distortion, and chromatic aberrations.

3. Types of Lenses:

- Achromatic Doublets: Use at least one or two achromatic doublets to correct chromatic aberrations. These are pairs of lenses made from different glass types to reduce the focus shift between different wavelengths of light (reduce color fringing).

- Meniscus Lenses: Positive or negative meniscus lenses can help in shaping the field curvature while controlling distortion. For a wide-angle system, positive meniscus lenses are often placed at the front.

- Plano-Convex/Concave Lenses: These lenses can be used to manage spherical aberrations, especially in the central region of the lens.

- Symmetry: Symmetric or near-symmetric lens designs (using two positive lenses with a negative lens in between) help minimize distortion and reduce coma.

4. Lens Layout Suggestion:

A 5-element design could look like this:

- First element: A positive meniscus lens near the object side to control distortion.
- Second element: A negative lens (concave) to help manage field curvature and reduce the system's effective focal length.
- Third and fourth elements: A pair of achromatic doublets to handle chromatic aberration and further reduce spherical aberrations.
- Fifth element: Another positive element to refine focus and manage edge sharpness.

5. Coatings and Glass Types:

- Use anti-reflective coatings to reduce ghosting and flare, especially since wide-angle systems are prone to internal reflections.
- Choose glass materials that balance cost and performance, such as N-BK7 for positive lenses and N-F2 or SF5 for the negative lens.

6. Lens Diameter:

- Given the field of view and focal length, your lens elements need to have diameters large enough to capture the full field. Depending on the final design, the front elements could be between 7mm and 9mm in diameter.

Optimizing the Design:

- Minimize Distortion: Since wide-angle lenses are prone to barrel distortion, incorporate lens elements that correct this, like the meniscus lenses in the early part of the design.
- Minimize Chromatic Aberration: Use achromatic doublets, particularly near the middle of the system, to ensure that the light of different wavelengths focuses at the same point.
- Minimize Field Curvature: Use lenses with appropriately designed curvatures to correct field curvature, ensuring the image remains sharp from the center to the edge.

Example Configuration (for Guidance):

1. First Element: Positive meniscus lens (convex toward the object).
2. Second Element: Negative plano-concave lens (controls focal length and field curvature).
3. Third Element: Achromatic doublet (for chromatic aberration correction).
4. Fourth Element: Another achromatic doublet (to refine aberration corrections and image quality).
5. Fifth Element: Plano-convex lens (final focusing element).

Final Notes:

- Start with ray tracing in Zemax to adjust the radii of curvatures, thicknesses, and spacings between lenses to achieve the desired FOV and minimize aberrations.
- Optimize your design iteratively, using the Merit Function Editor in Zemax, focusing on minimizing chromatic aberration, spherical aberration, and distortion.

This design approach should give you a solid starting point for a high-performance lens system for your CMOS sensor with a > 80-degree field of view.

Case Study 3:

Prompt 3

Based on this project, what are some recommended power supplies that are currently in the market?

ChatGPT Outcome:

For a project like the iSPY optical security system, which combines advanced imaging, processing, and free-space optical (FSO) communication technologies, selecting the right power supplies is crucial to ensure system reliability, efficiency, and safety. Here are some recommended types of power supplies that would be appropriate:

1. Switch Mode Power Supply (SMPS)

- Description: SMPS units convert AC to DC power efficiently, providing stable DC output while handling varying load conditions and power factors. They are preferred for their efficiency and compact size.
- Application: Useful for powering the MCU, camera modules, and other digital components on PCB #1 and #2 where efficiency and space are critical.

2. Battery Packs

- Description: Rechargeable battery packs, such as lithium-ion or lithium-polymer, offer portability and can provide a high-energy density, which is vital for systems that may need to operate independently of a fixed power source.
- Application: Batteries can power the entire system during mobile operations or when installed in locations without direct access to mains electricity. They're particularly crucial for backup power to ensure the system remains operational during power outages.

3. Linear Power Supply

- Description: Known for providing extremely low noise and ripple compared to SMPS, which is beneficial for sensitive electronic components.
- Application: Ideal for powering noise-sensitive components such as the analog circuits in transceivers and sensors, especially where signal integrity is paramount.

4. Power over Ethernet (PoE)

- Description: PoE can transmit electrical power, along with data, to remote devices over standard twisted-pair cable in an Ethernet network.

- Application: Suitable for networked components within the system, such as cameras or any part of the system that communicates data over Ethernet. This simplifies wiring and reduces installation costs and complexity.

5. Uninterruptible Power Supply (UPS)

- Description: UPS systems provide power backup using batteries and an inverter to supply AC or DC power when main power is unavailable, and also condition incoming power to protect against surges, spikes, and brownouts.

- Application: Ensures continuous operation of the entire security system during power failures, protecting against data loss and hardware damage during critical operations.

6. Solar Power Solutions

- Description: For outdoor installations, incorporating solar panels can offer a renewable energy solution, reducing reliance on external power sources and enhancing the system's sustainability.

- Application: Powers components during daylight with battery storage to provide energy during nighttime or cloudy conditions, ideal for remote or environmentally sensitive locations.

7. Regulated DC Power Supplies

- Description: These provide a constant voltage or current despite variations in load current or input voltage, which is crucial for maintaining the stability and performance of the electronic components.

- Application: Feeds stable DC power to sensitive circuits and components within the iSPY system, ensuring reliable operation and preventing damage due to voltage fluctuations.

Each of these power supply types brings specific benefits that can enhance the performance and reliability of your iSPY system, making it crucial to choose based on the component requirements, operational environment, and intended deployment scenario.

Copilot Outcome:

For your iSPY project, which requires a 5V DC regulated output and low power consumption, here are some recommended power supplies currently available on the market:

1. **Mean Well LRS-50-5:**
 - a. **Output:** 5V DC, 10A (50W)
 - b. **Efficiency:** Up to 88%
 - c. **Features:** Compact size, high reliability, low cost, and built-in protections (short circuit, overload, over voltage).
 - d. **Price:** [Around \\$15-\\$20¹](#).
2. **Adafruit 5V 4A Switching Power Supply:**
 - a. **Output:** 5V DC, 4A (20W)
 - b. **Features:** High efficiency, compact design, and includes a 2.1mm barrel jack.
 - c. **Price:** [Around \\$15²](#).
3. **CUI Inc. SWI5-5-N-P5:**
 - a. **Output:** 5V DC, 1A (5W)
 - b. **Features:** Wall plug-in power supply, high efficiency, and includes multiple plug options for international use.
 - c. **Price:** [Around \\$10³](#).
4. **RS Pro 5V DC Power Supply:**
 - a. **Output:** 5V DC, 3A (15W)
 - b. **Features:** Compact, reliable, and includes over voltage, over current, and short circuit protection.
 - c. **Price:** Around \$20.
5. **Mean Well RS-15-5:**
 - a. **Output:** 5V DC, 3A (15W)
 - b. **Efficiency:** Up to 78%
 - c. **Features:** Compact size, high reliability, and built-in protections (short circuit, overload, over voltage).
 - d. **Price:** Around \$10-\$15.

These power supplies are well-suited for low-power applications like your iSPY project, providing reliable and efficient power with necessary protections. If you need more specific recommendations or have any other questions, feel free to ask!

Google AI Outcome:

When looking for a power supply for a designed project, some well-regarded options include Meanwell power supplies, Cincon products, RECOM power supplies, and depending on your specific needs, the ZKETECH EBD-A20H variable DC power supply could be a good choice as well; all offering a variety of options for different voltage and current requirements, with good reliability and industry standard certifications.

Key points to consider when choosing a power supply:

- **Voltage and Current Requirements:**

Match the output voltage and current of the power supply to your project's needs, ensuring sufficient capacity to handle peak loads.

- **Form Factor:**

Consider the physical size and mounting options needed for your project.

- **Efficiency:**

Look for power supplies with high efficiency ratings to minimize heat generation.

- **Safety Certifications:**

Ensure the power supply meets relevant safety standards like UL, CE, and TUV.

Some recommended brands and series:

- **Meanwell:** Considered a top choice for a wide range of power supplies, known for reliability and quality.
- **Cincon:** Another reliable option with a focus on efficiency and long lifespan
- **RECOM:** Offers a broad selection of AC-DC and DC-DC converters, often suitable for compact applications
- **WAGO ECO 2:** Compact industrial power supply with good space-saving design

Carlo Gavazzi SPMA Series: Compact DIN-mount power supply suitable for industrial and building automation

Case Study 4:

Prompt 4

Can you recommend me some camera modules for a surveillance camera I am trying to build? I will work on the lens design later and choose the best lens or lenses but I want to have a camera module in mind. I am looking for something that can give me a wide field of view but not too expensive.

ChatGPT Outcome 4

Here are some camera modules that may suit your surveillance camera design, offering a wide field of view while remaining cost-effective:

1. **Arducam IMX219 Camera Module**

- **Sensor:** Sony IMX219
- **Resolution:** 8 MP (3280x2464)
- **Field of View:** Options available with lenses that provide up to 160° wide-angle.
- **Interface:** MIPI CSI-2
- **Price:** Around \$25-40 USD

- **Features**: This is a popular camera module for Raspberry Pi, offering decent image quality for surveillance and other general-purpose applications. You can replace the lens to achieve your desired field of view.

2. **Raspberry Pi Camera Module 3**

- **Sensor**: Sony IMX708
- **Resolution**: 12 MP (4608x2592)
- **Field of View**: 120° (Wide-angle version)
- **Interface**: MIPI CSI-2
- **Price**: Around \$30-40 USD
- **Features**: It offers a wide field of view (120°) in the wide-angle version, which fits your requirement for wide surveillance. It supports HDR (High Dynamic Range), which is useful for well-lit environments.

3. **E-Con Systems See3CAM_CU55**

- **Sensor**: Sony IMX547
- **Resolution**: 5 MP (2592x1944)
- **Field of View**: 100° with the standard lens; other lenses available for wider angles.
- **Interface**: USB 3.0
- **Price**: Around \$60-80 USD
- **Features**: This module provides high-quality imaging and supports a range of lens options. Its USB interface simplifies integration, and it's well-suited for real-time applications.

4. **OV5647 Camera Module**

- **Sensor**: OmniVision OV5647
- **Resolution**: 5 MP (2592x1944)
- **Field of View**: Available with lenses that can offer up to 120°.
- **Interface**: MIPI CSI-2
- **Price**: Around \$15-25 USD
- **Features**: A budget-friendly option, this module is used in many low-cost surveillance and robotics systems. It supports interchangeable lenses for a customizable field of view.

5. **AI-Thinker ESP32-CAM**

- **Sensor**: OV2640
- **Resolution**: 2 MP (1600x1200)

- **Field of View**: Around 100° with the stock lens.
- **Interface**: Wi-Fi/Bluetooth (ESP32-based)
- **Price**: Around \$10-20 USD
- **Features**: This module is extremely affordable and includes an ESP32 microcontroller for wireless communication, making it suitable for low-cost wireless surveillance solutions. While the resolution is lower, it could work for certain low-budget surveillance applications.

Considerations for Choosing a Camera Module:

- **Field of View**: Since you're focusing on a wide field of view, you can choose modules that either come with wide-angle lenses or support lens swapping, allowing you to design a custom lens later.
- **Interface**: Depending on your system setup, decide between MIPI CSI-2 (common in embedded systems) or USB (easier for plug-and-play integration).
- **Resolution**: Higher resolutions are better for capturing more detail, especially if you need to zoom in on portions of the image.
- **Price**: All the listed options are relatively affordable and range from \$10 to \$80 depending on the features and resolution.

These camera modules should give you flexibility for your lens design while being budget-friendly for your surveillance system.

Prompt 4.1

Can I replace the lenses of the 2 option? 0.3 MegaPixels USB Camera for Raspberry Pi / NVIDIA Jetson Nano / UNIHAIKER what is the shutter type?

ChatGPT Outcome 4.1

Yes, you can replace the lenses on the **Raspberry Pi Camera Module 3**, but it requires some modifications. By default, the camera module comes with a fixed lens, but third-party accessories, like lens adapters, can allow you to swap lenses to customize the field of view or optical performance.

Steps to Replace the Lens:

1. **Remove the Existing Lens**: You would need to carefully remove the fixed lens, which usually involves unscrewing or detaching the plastic housing.
2. **Use a Lens Adapter**: Third-party vendors offer adapters that can be attached to the camera module, allowing you to use different lenses (like M12 or CS-mount lenses).

3. **Choose Compatible Lenses**: Make sure to choose a lens that matches the sensor size (1/2.43" for the IMX708 sensor in this case) to ensure proper image coverage and minimize vignetting.

Since you are planning to design a custom lens system later on, replacing the lens on this module could give you more flexibility in achieving your desired field of view. Just keep in mind that modification requires precision, and any changes to the lens could affect calibration and focus.

For a **0.3 MegaPixel USB camera** designed for platforms like the Raspberry Pi, NVIDIA Jetson Nano, or UNIHAKER, the shutter type is most likely **rolling shutter**. This is because:

- **Rolling shutters** are more common in lower-cost and lower-resolution camera modules due to their simpler design and lower power requirements.
- **Global shutters** are typically found in more expensive, high-performance cameras that require accurate image capture of fast-moving objects, which isn't typical for a 0.3 MP camera.

To confirm the exact shutter type, it's best to check the datasheet or specifications provided by the manufacturer, as some models may vary. Would you like help finding those details for a specific model?

GPTA

Prompt 1

Can you describe entropy coding (e.g. huffman and arithmetic) in terms of live video feed compression?

Outcome 1

Entropy coding, such as Huffman and arithmetic coding, plays a key role in the compression of live video feeds by efficiently encoding data based on its statistical properties. Here's a breakdown of how these methods work in the context of video compression:

1. Huffman Coding in Video Compression:

Huffman coding is a lossless entropy coding technique used to compress data by encoding frequently occurring symbols with shorter bit lengths and less frequent symbols with longer bit lengths. In a live video feed, frames consist of a large amount of data that can be

processed symbolically (e.g., pixel values, differences between pixel values in adjacent frames, or quantized coefficients from a transform like DCT used in formats like H.264).

- **Process:**

- **Frame Analysis:** The video is broken down into frames, and each frame is analyzed to determine the frequency of certain patterns (e.g., pixel intensity values or motion vectors).
- **Symbol Mapping:** Huffman coding then maps more frequently occurring patterns to shorter codes and less frequent ones to longer codes.
- **Transmission:** The compressed frame data is transmitted in the form of a bitstream, where shorter codes dominate, leading to reduced overall data size.

Example: Suppose a video has many repeated pixel values for background color in a stationary scene. These frequent pixel values can be assigned shorter codes, reducing the number of bits needed for the static portion of the video.

- **Advantages for Live Video:**

- Huffman coding can be used to encode residual data in video compression algorithms like H.264, which first applies motion compensation to remove temporal redundancy between frames.
- The compression speed of Huffman coding makes it suitable for real-time video feed compression.

- **Limitations:**

- Huffman coding is relatively simple and may not achieve the highest compression rates for highly complex video data, where more adaptive techniques like arithmetic coding might outperform it.

2. Arithmetic Coding in Video Compression:

Arithmetic coding is a more sophisticated entropy coding technique that often outperforms Huffman coding by encoding entire sequences of symbols as a single number (usually a fraction). This provides greater flexibility and efficiency, especially for symbols with probabilities that aren't powers of two.

- **Process:**

- **Frame Analysis:** Like Huffman coding, arithmetic coding starts by analyzing video frames and predicting the probability of different symbols (e.g., pixel values or residuals from motion compensation).
- **Range Assignment:** Instead of assigning fixed-length codes, arithmetic coding represents the entire sequence of symbols as a fractional number within a range based on their probabilities.

- **Bitstream Generation:** The resulting bitstream is more compact, as arithmetic coding can efficiently represent fractional probabilities with high precision.

Example: Suppose a region of the video has a predictable color gradient or motion pattern. Arithmetic coding can represent this predictability in a compact form, minimizing the data required for transmission.

- **Advantages for Live Video:**

- Arithmetic coding is more adaptive than Huffman coding, meaning it can handle video data with complex or varying statistical properties (e.g., scenes with sudden movement or high texture).
- It achieves near-optimal compression rates, which can be critical for bandwidth-limited live video transmission.

- **Limitations:**

- Arithmetic coding is more computationally intensive, which could make it challenging for very low-latency live video compression systems unless hardware-accelerated or highly optimized implementations are used.

Integration in Live Video Compression Standards (e.g., H.264/HEVC):

Both Huffman and arithmetic coding are used in popular video compression standards. For instance:

- **H.264/AVC:** Uses a combination of Huffman coding (for variable-length coding) and Context-Adaptive Binary Arithmetic Coding (CABAC) for more efficient encoding. CABAC plays a crucial role in compressing the residual data after motion estimation.
- **HEVC (H.265):** Leverages arithmetic coding (specifically CABAC) more heavily to reduce the bit rate even further, making it particularly suitable for higher resolution live streams like 4K or 8K.

Summary:

- **Huffman Coding:** Quick, simple, and effective for coding frequent symbols in predictable scenes, but less optimal for complex or highly dynamic video data.
- **Arithmetic Coding:** Offers better compression efficiency for complex video content but at the cost of higher computational complexity, which may impact real-time encoding unless optimized.

In a live video feed compression system, both Huffman and arithmetic coding work to minimize the bitstream size by efficiently encoding video data based on its statistical properties, which is essential for transmitting high-quality video over bandwidth-constrained channels.

GPTB

Prompt 1

Could you explain bandwidth and latency in terms of my project?

Outcome 1

In the context of your video compression and transmission system, **bandwidth** and **latency** are two critical factors to consider for efficient data transmission over free space using a laser.

Bandwidth:

Bandwidth refers to the maximum amount of data that can be transmitted over the laser communication channel in a given period of time. It is usually measured in bits per second (bps). For your project, the **bandwidth** defines how much compressed video data (in bitstream form) can be sent per second through the laser.

- **Higher bandwidth** means you can transmit more video data in less time, which is especially important if you're dealing with high-resolution or high-frame-rate video.
- **Lower bandwidth** can result in slower transmission, potentially causing delays or the need to use more aggressive video compression, which can reduce quality.

Since you are working with video compression, it's essential to optimize the amount of data sent per second. Efficient compression algorithms will help minimize the bitstream size, allowing it to fit within the available bandwidth.

Latency:

Latency is the time delay between when the video data is sent from the source (via laser) and when it is received and decoded at the destination. In your case, latency is affected by several factors:

- **Propagation delay:** the time it takes for the laser signal to travel through free space from the transmitter to the receiver.
- **Processing delay:** the time spent compressing the video, encoding it into a bitstream, and decoding it on the receiving end.
- **Buffering delay:** the time spent waiting for enough data to be buffered to avoid interruptions in the video stream.

Low latency is crucial for real-time applications, such as live video transmission, where any noticeable delay can disrupt the user experience.

In summary:

- **Bandwidth** determines how much video data can be sent per second, affecting video quality and frame rate.
- **Latency** affects the delay between video capture, transmission, and playback, critical for real-time streaming.

Balancing compression efficiency (to minimize bandwidth use) and ensuring low latency will be key to the success of your laser-based video transmission system.

Prompt 2

Would compression also play a factor in latency?

Outcome 2

Yes, **compression** plays a significant role in latency as well, especially in your video compression and transmission system.

Here's how compression affects latency:

1. Compression Time (Encoding Latency):

Compression involves encoding the video data into a smaller bitstream, which can introduce latency due to the time it takes to process the video frames. The more complex the compression algorithm (e.g., higher compression ratios or advanced encoding techniques), the longer it may take to compress the video. This contributes to **encoding latency**.

- **Higher compression** (more processing): can reduce bandwidth needs but takes more time to encode, thus increasing the latency.
- **Lower compression** (less processing): reduces the encoding time, thus lowering latency but may require more bandwidth for transmission.

2. Decompression Time (Decoding Latency):

On the receiving end, the compressed video bitstream needs to be **decompressed** (decoded) before it can be displayed. Like compression, decompression also requires processing time, which introduces additional latency, especially if the compression method is complex.

- **Complex codecs**: like H.264 or H.265 (often used in video compression) might take more time to decode, increasing latency.
- **Simpler codecs**: may have faster decompression times, reducing latency but might not compress as effectively.

3. Trade-off Between Compression Ratio and Latency:

There's often a trade-off between how much compression you apply and the latency you introduce:

- **Aggressive compression**: reduces the amount of data to be transmitted (saving bandwidth), but adds time for encoding and decoding, which increases latency.
- **Light compression**: transmits larger amounts of data (requiring more bandwidth), but reduces the time spent compressing and decompressing, thus lowering latency.

4. Keyframe Interval and Buffering:

Most video compression schemes use keyframes (complete frames) and intermediate frames (which store only the differences between frames). The **keyframe interval** (how often a full frame is encoded) can impact latency. Longer intervals between keyframes might save bandwidth but introduce latency as the receiver needs to wait for the next keyframe to start decoding a portion of the stream.

- **Short keyframe intervals** reduce latency but increase bandwidth consumption.

5. Compression Mode:

Some modern video codecs, like H.265, support **low-latency modes**, where the compression is optimized to minimize latency at the expense of a slightly larger bitstream.

In summary, **compression does affect latency** due to the time required for both encoding and decoding. For your project, you'll need to balance the level of compression with the system's latency requirements. If real-time transmission is a priority, you might opt for faster compression algorithms (low-latency codecs) even if they don't achieve the highest compression ratios, to ensure minimal delay.

GPTC

Prompt 1

Can you give me libraries for codecs in c++, java, and python? Please provide an explanation of each library as well.

Outcome 1

Here are some libraries for working with codecs in C++, Java, and Python:

C++ Codec Libraries

1. **FFmpeg** (libavcodec)
Description: FFmpeg is a complete, cross-platform solution to record, convert and stream audio and video. Its codec library, libavcodec, is widely used for encoding and decoding audio and video. It supports a vast range of codecs including H.264, VP8, VP9, AAC, MP3, and many others.
 - **Use case:** Video and audio encoding, decoding, and transcoding.
 - **Pros:** Supports many formats, well-documented, highly optimized.
 - **Cons:** Steeper learning curve; LGPL license may impose restrictions.
 - **Website:** [FFmpeg](https://ffmpeg.org/)
2. **GStreamer**
Description: GStreamer is a multimedia framework that allows building media-handling components like codecs and transcoders. It supports a variety of codecs via plugins, which can be combined to create pipelines for media processing.
 - **Use case:** Real-time streaming, media playback, and processing.

- **Pros:** Highly modular, great for building complex media pipelines.
- **Cons:** More complex than FFmpeg, especially for beginners.
- **Website:** GStreamer

3. OpenH264

Description: OpenH264 is a codec library that provides real-time H.264 encoding and decoding. It's developed by Cisco and is widely used for video streaming and teleconferencing solutions.

- **Use case:** Real-time video compression (H.264).
- **Pros:** Lightweight, focused on H.264.
- **Cons:** Limited to H.264; doesn't support other formats.
- **Website:** [OpenH264](#)

Java Codec Libraries

1. JCodec

Description: JCodec is a pure Java library for video and audio codec processing. It supports basic operations like encoding and decoding H.264 and VP8 formats, as well as media container operations such as MP4.

- **Use case:** Basic encoding and decoding of common formats in Java.
- **Pros:** Written in pure Java, no native dependencies.
- **Cons:** Limited codec support, lacks real-time performance optimizations.
- **Website:** [JCodec](#)

2. Xuggle

(Xuggler)

Description: Xuggle is a Java wrapper for FFmpeg, providing an easy way to work with audio and video codecs in Java. It supports a wide variety of formats and offers access to FFmpeg's vast functionality.

- **Use case:** Handling multimedia streams and encoding/decoding them in Java.
- **Pros:** Supports most codecs, built on FFmpeg.
- **Cons:** Project maintenance has been inconsistent in recent years.
- **Website:** [Xuggle](#)

3. JMF (Java Media Framework)

Description: JMF was an older framework for capturing, playing, streaming, and transcoding media. It includes support for some codecs but is no longer actively developed or maintained.

- **Use case:** Legacy systems and simple multimedia tasks.
 - **Pros:** Provides basic multimedia support.
 - **Cons:** Outdated, limited codec support.
 - **Website:** [JMF](#) (deprecated)
-

Python Codec Libraries

1. PyAV

Description: PyAV is a Pythonic binding for FFmpeg, allowing easy access to FFmpeg's audio and video codec capabilities. It can be used to encode, decode, and manipulate multimedia files using FFmpeg's libavcodec.

- **Use case:** Working with audio and video files for tasks like encoding/decoding, file manipulation, and transcoding in Python.
- **Pros:** Simple FFmpeg binding; easy-to-use API.
- **Cons:** Limited to FFmpeg's capabilities; must install FFmpeg separately.
- **Website:** [PyAV](#)

2. MoviePy

Description: MoviePy is a high-level library for editing video files in Python. It supports reading and writing video files, as well as applying transformations such as cutting, concatenation, effects, and basic audio/video codec handling.

- **Use case:** Simple video editing and codec manipulation.
- **Pros:** User-friendly, high-level API; integrates with FFmpeg.
- **Cons:** Not suitable for complex, low-level codec operations.
- **Website:** [MoviePy](#)

3. FFmpeg-Python

Description: FFmpeg-Python is a convenient Python wrapper for FFmpeg, enabling users to build and execute FFmpeg commands programmatically. It supports all FFmpeg codecs, containers, and filters.

- **Use case:** Complex multimedia processing pipelines using FFmpeg.
 - **Pros:** Provides access to all of FFmpeg's functionality; flexible.
 - **Cons:** Steeper learning curve for FFmpeg command syntax.
 - **Website:** [FFmpeg-Python](#)
-

These libraries span a wide range of codec support and are often built on FFmpeg or similar core technologies, providing different levels of abstraction depending on your needs.

Prompt 2

Does ffmpeg support H.265 as well?

Outcome 2

Yes, **FFmpeg** supports **H.265 (HEVC)** encoding and decoding. The H.265 codec offers better compression efficiency than H.264, which is especially useful for high-definition and 4K video content.

Key Details About H.265 Support in FFmpeg:

1. Encoding:

- FFmpeg can encode videos to H.265 using the **libx265** encoder.
- libx265 is an open-source encoder for H.265/HEVC, offering high-efficiency video compression.
- You can specify libx265 during encoding with the `-c:v libx265` option.

2. Decoding:

- FFmpeg can also decode H.265 videos using the **libx265** decoder.
- You can process and view H.265-encoded content by simply specifying the input file.

Example FFmpeg Command to Encode Video Using H.265:

bash

Copy code

```
ffmpeg -i input.mp4 -c:v libx265 -crf 28 -preset slow output.mp4
```

- `-c:v libx265`: Specifies the H.265 encoder.
- `-crf 28`: Controls the quality (Constant Rate Factor). Lower values give better quality but larger files.
- `-preset slow`: Optimizes the encoding process (slow gives better compression at the cost of speed).

Additional Notes:

- **Licensing**: H.265 (HEVC) is patented, and although libx265 is open-source, you may need to consider licensing fees if you plan to use it in a commercial product.
- **Hardware Acceleration**: FFmpeg also supports hardware-accelerated encoding and decoding for H.265 if your GPU supports it (e.g., using `nvenc` for NVIDIA GPUs).

FFmpeg provides comprehensive support for H.265, making it an excellent choice for projects requiring high-efficiency video compression.

GPTD

Prompt 1

What are three potential MCUs that could work well with my project of having computer vision and video compression?

Outcome 1

For a project involving computer vision and video compression, you need a microcontroller (MCU) with sufficient processing power, memory, and support for hardware acceleration for video tasks. Here are three potential MCUs that could work well:

1. ESP32-S3

- **Key Features:**
 - **Dual-core 32-bit Xtensa LX7 CPU**, up to 240 MHz.
 - Integrated **AI accelerator** and **vector instructions** for efficient handling of image and signal processing tasks.
 - Supports **8/16-bit parallel cameras**, making it suitable for basic computer vision tasks.
 - **Wi-Fi and Bluetooth** integrated for wireless video transmission.
- **Advantages:**
 - Low cost, relatively low power consumption, and built-in wireless support.
 - Suitable for lightweight video compression tasks like MJPEG.
- **Limitations:**
 - Limited for heavy video compression tasks like H.264 without external hardware.

2. STM32H7 Series

- **Key Features:**
 - **Cortex-M7** core running at up to 480 MHz with **2MB flash** and **1MB RAM**.
 - **Hardware JPEG encoding/decoding** support, which can significantly help with video compression.
 - **FMC and DCMI interfaces** for connecting cameras and high-speed data transfers.
- **Advantages:**
 - High processing power, making it suitable for more advanced video compression algorithms.

- The hardware JPEG codec offloads the compression workload from the CPU.
- **Limitations:**
 - More complex to program and higher power consumption than ESP32.

3. Raspberry Pi Pico W (RP2040 with Wireless)

- **Key Features:**
 - Dual-core **Cortex-M0+** running at 133 MHz, with **264KB of RAM**.
 - **Pico W** version includes **Wi-Fi** for wireless video transmission.
 - Support for **Pio** (Programmable I/O) that allows handling of parallel interfaces like camera input.
- **Advantages:**
 - Affordable, with a large developer community and libraries for interfacing with cameras.
 - While it doesn't have built-in video compression, it can interface with external compression modules (e.g., using an external H.264 encoder).
- **Limitations:**
 - Limited onboard memory and processing power compared to the STM32H7 and ESP32-S3, so external hardware would be needed for high-end compression.

The STM32H7 or ESP32-S3 would be ideal for handling lightweight computer vision tasks and basic video compression natively, while the Raspberry Pi Pico W could be suitable if external components are added for more complex tasks.

Prompt 2

What about the msp 430fr6989

Outcome 2

The **MSP430FR6989** is a low-power microcontroller from Texas Instruments, part of the MSP430 series, known for ultra-low power consumption and high energy efficiency. However, it may not be the best choice for a project involving computer vision and video compression due to the following reasons:

Limitations of the MSP430FR6989:

1. Processing Power:

- The MSP430FR6989 is based on a **16-bit RISC architecture** with a maximum clock speed of **16 MHz**, which is much lower compared to the MCUs typically used in computer vision or video compression tasks.

- Video compression and computer vision tasks are computationally intensive, requiring much higher processing power, such as 32-bit architectures with higher clock speeds and additional computational capabilities (like hardware accelerators).

2. **Memory:**

- It comes with **128 KB FRAM** and **2 KB SRAM**, which is quite limited for handling video data or even image buffers.
- Computer vision tasks, especially video processing, require a significant amount of memory for storing frames, applying transformations, and buffering data streams, which would quickly exceed this capacity.

3. **No Camera or Graphics Interfaces:**

- Unlike some other MCUs, the MSP430FR6989 does not have dedicated camera or display interfaces (like DCMI or parallel camera interfaces), which are crucial for real-time image acquisition and video processing.

4. **No Dedicated Video/Compression Hardware:**

- The MSP430 lacks any hardware acceleration for video encoding or decoding (like JPEG or H.264 hardware codecs) that are common in other microcontrollers like the STM32H7 series.
- Without hardware support, video compression would need to be done entirely in software, which would be far too slow on this chip.

When the MSP430FR6989 Could Be Useful:

- If your project required ultra-low power consumption and only basic sensing tasks (like reading from low-power sensors or basic IoT applications), the MSP430FR6989 would be an excellent choice.
- It could be used in a peripheral role, such as controlling sensors or handling communication tasks in a more complex system where another device does the heavy lifting for video and image processing.

Conclusion:

The MSP430FR6989 is not a good fit for video compression or computer vision applications due to its limited processing power, memory, and lack of video-related hardware. You would be better off with a more powerful MCU like the **ESP32-S3**, **STM32H7**, or even a small single-board computer like the **Raspberry Pi** if your project demands more computational power and memory.

Prompt 3

What are three potential raspberry pi boards that I could use for this project?

Outcome 3

For a project involving computer vision and video compression, Raspberry Pi boards offer a good balance of computational power, flexibility, and available software libraries. Here are three Raspberry Pi boards that could be a great fit:

1. Raspberry Pi 4 Model B

- **Key Features:**

- **Quad-core Cortex-A72 (ARM v8)** 64-bit SoC @ 1.5GHz.
- Available with **2GB, 4GB, or 8GB RAM**, which is more than enough for video processing and compression tasks.
- **Dual-display support** with 4K output, which is useful if you need to monitor your video feed or test video quality.
- Includes **VideoCore VI GPU** with hardware acceleration for **H.264** and **H.265** video encoding/decoding.

- **Advantages:**

- Great for computer vision tasks using frameworks like OpenCV and TensorFlow Lite.
- High processing power and memory make it capable of handling real-time video compression and transmission.
- A large community and a wide range of libraries for interfacing with cameras, such as the **Raspberry Pi Camera Module**.

- **Ideal For:**

- More complex video compression tasks, live streaming, and image processing applications.

2. Raspberry Pi Zero 2 W

- **Key Features:**

- **Quad-core Cortex-A53** CPU @ 1GHz.
- **512MB RAM**, which is lower than the Pi 4 but can still handle lightweight video compression and basic computer vision tasks.
- Built-in **Wi-Fi and Bluetooth**, making it suitable for wireless video transmission projects.
- Compatible with the **Raspberry Pi Camera Module**.

- **Advantages:**

- Extremely compact and power-efficient, making it a great option for portable or battery-powered projects.

- Cheaper than the Raspberry Pi 4, which makes it appealing for simpler projects or cost-sensitive applications.
- Supports hardware-accelerated H.264 encoding/decoding for efficient video compression.
- **Ideal For:**
 - Lightweight video processing tasks, basic computer vision, and low-cost implementations of video streaming.

3. Raspberry Pi 3 Model B+

- **Key Features:**
 - **Quad-core Cortex-A53 CPU @ 1.4GHz.**
 - **1GB RAM**, which is enough for mid-range video compression tasks and basic computer vision.
 - **Dual-band 2.4GHz and 5GHz Wi-Fi** with Gigabit Ethernet, allowing faster video transmission over networks.
 - Hardware support for **H.264 encoding/decoding**.
- **Advantages:**
 - More affordable than the Pi 4 while still providing decent processing power for video compression.
 - Supports most of the same peripherals as the Pi 4, including the camera modules, making it a versatile choice for video projects.
- **Ideal For:**
 - Medium complexity video compression, remote video monitoring, and projects where high-end performance isn't a strict requirement.

Comparison Overview:

- **Raspberry Pi 4 Model B:** Best choice for handling complex video compression (H.264/H.265), real-time image processing, and more resource-demanding computer vision tasks.
- **Raspberry Pi Zero 2 W:** Great for compact, portable solutions with lighter video processing tasks. Its small size and low power draw make it suitable for projects where space or energy efficiency is key.
- **Raspberry Pi 3 Model B+:** Balanced option for mid-range video tasks, where cost is a factor but performance is still important for moderate video processing.

These Raspberry Pi boards provide a variety of options depending on the complexity of your video compression and computer vision needs. The **Pi 4 Model B** would be ideal for

more advanced tasks, while the **Zero 2 W** could be a solid choice for a simpler or more space-constrained solution.

GPTE

Prompt 1

what should I write about when writing research on codecs and how to compress a live video stream to transfer across a laser using bitstream.

Outcome 1

When writing research on codecs and how to compress a live video stream for transmission via a laser using a bitstream, you should cover both the theoretical and practical aspects of video compression and transmission. Below is a structured approach to what you can include in your research:

1. Introduction to Codecs and Video Compression

- **Definition of Codecs:** Explain what a codec is—combination of *encoder* and *decoder*—and its role in video compression.
- **Types of Codecs:** Discuss the major categories:
 - **Lossless Codecs** (e.g., Huffman coding, Run-Length Encoding)
 - **Lossy Codecs** (e.g., H.264, H.265, VP9)
- **Importance of Compression:** Emphasize the need for compression in reducing bandwidth and storage requirements, especially for real-time applications like live streaming.

2. Understanding Video Compression Techniques

- **Spatial Compression:** Compression within individual frames (Intra-frame compression) by removing redundant data within a single image (e.g., JPEG-based).
- **Temporal Compression:** Compression across frames (Inter-frame compression) by encoding the differences between consecutive frames, such as using motion estimation and compensation (e.g., P-frames, B-frames).
- **Entropy Coding:** Methods to further reduce the size of encoded frames by efficiently coding the data (e.g., Huffman coding, Arithmetic coding).
- **Challenges in Live Video Compression:** Discuss the complexity of achieving low-latency and high-efficiency compression without significant quality loss.

3. Real-Time Streaming Requirements

- **Latency and Bandwidth Considerations:** Detail how latency is critical for live streaming, as is minimizing the bit rate for efficient transmission over limited bandwidth (such as a laser).

- **Quality vs. Compression Trade-Off:** Explain how compression reduces quality but allows for higher transmission efficiency. Highlight tools like bitrate control, adaptive compression (CRF in H.264), and variable vs constant bit rates.
- **Codec Comparison:** Compare commonly used codecs for live video streaming (H.264/AVC, H.265/HEVC, VP8/VP9, AV1), discussing factors like compression efficiency, computational complexity, and compatibility.

4. Bitstream Transmission Using Laser

- **Overview of Bitstream Transmission:** Explain how video streams are converted into a bitstream, including formatting into packets for transmission over different media, in this case, laser communication.
- **Laser Communication Overview:**
 - **Free-Space Optical Communication (FSO):** How lasers are used to transmit data wirelessly, including an overview of modulation schemes (e.g., amplitude modulation, pulse-width modulation) to encode the bitstream.
 - **Advantages:** High bandwidth, low interference, and long-range communication potential.
 - **Challenges:** Susceptibility to environmental factors like weather (fog, rain) and line-of-sight requirements.

5. Modulation Techniques for Laser Transmission

- **Pulse Code Modulation (PCM):** Explain how data from a compressed video bitstream can be encoded using PCM.
- **Amplitude Modulation:** Technique where the strength (amplitude) of the laser beam is modulated to represent binary data.
- **Frequency/Phase Modulation:** Alternative methods to improve the signal quality or robustness.
- **Error Correction and Recovery:** Discuss error detection/correction techniques (like Reed-Solomon codes, Hamming codes) to ensure data integrity, especially for long-distance transmission or over noisy environments.

6. Compression Techniques in Relation to Transmission Medium

- **Bitrate Control:** How to manage bitrate and compression settings to match the available bandwidth of the laser transmission.
- **Adaptive Streaming:** Investigate how adaptive bitrate streaming (ABR) techniques could be employed to adjust the compression dynamically based on transmission conditions.
- **Fragmentation of Video Stream:** Methods for dividing the compressed video into smaller packets suitable for transmission over a bitstream and how to synchronize the frames during reception.

7. Implementation and Performance Evaluation

- **Experimentation:** Explain how you might implement such a system, including software (like FFmpeg) and hardware components (laser transmitter/receiver, camera, FPGA, microcontroller).
- **Simulations:** Use simulation tools to model the performance of the transmission under different conditions, such as varying distances, environmental factors, and data rates.
- **Metrics for Evaluation:** Quality of experience (QoE) metrics such as peak signal-to-noise ratio (PSNR), Structural Similarity Index (SSIM), bitrate, frame rate, and latency.

8. Security Considerations

- **Encryption for Transmission:** Discuss techniques for securing the bitstream during transmission, such as AES encryption applied to the video stream before modulation.
- **Authentication and Tamper Detection:** Ways to ensure that the transmitted data has not been altered during transmission.

9. Conclusion and Future Directions

- **Current Limitations:** Summarize the current challenges and limitations in using laser communication for real-time video transmission, including environmental susceptibility and codec computational complexity.
- **Future Research:** Discuss potential areas for further research, such as improving error correction techniques, enhancing codec efficiency, or developing new modulation schemes for laser-based communication.

By covering these areas, your research will provide a comprehensive overview of the topic, integrating both the theoretical underpinnings of video compression and practical applications related to laser-based transmission.

GPTF

Prompt 1

I need a microprocessor chip that can run python, opencv, h.264, and vscode. It should also be compatible with certain raspberry pi camera modules

Outcome 1

For your requirements, a good option would be the **Raspberry Pi 4 Model B** or the **Raspberry Pi 400**, as they offer enough processing power to run Python, OpenCV, H.264 encoding, and even Visual Studio Code.

Here's a breakdown of why these models fit your needs:

1. Operating System and Software Support

- The Raspberry Pi 4 and 400 models support Raspberry Pi OS, which is compatible with Python, OpenCV, and H.264 encoding.
- You can install Visual Studio Code, as it has an official version for Raspberry Pi OS, allowing you to work directly on the device.

2. Performance

- The **Raspberry Pi 4** (available in 2GB, 4GB, and 8GB RAM variants) is powerful enough to handle lightweight machine learning tasks with OpenCV and perform video encoding using H.264.
- The **Raspberry Pi 400** (4GB RAM) is slightly faster due to its form factor and cooling but otherwise provides similar performance.

3. Compatibility with Camera Modules

- Raspberry Pi models support the official **Raspberry Pi Camera Modules** (e.g., Camera Module 2, HQ Camera), and with recent updates, they support other third-party cameras through the **libcamera** framework.
- The hardware acceleration on Raspberry Pi OS makes these models ideal for real-time video processing with OpenCV, which is critical for applications that involve computer vision.

4. Additional Hardware Options

- If your application requires intensive video processing, you may consider adding a **Coral USB Accelerator** for dedicated machine learning acceleration or **Raspberry Pi Compute Module 4** for a custom setup.

Would you like recommendations on specific camera modules or accessories to enhance performance?

Prompt 2

I want to buy the chip so I can create my own pcb

Outcome 2

If you're designing a custom PCB and need a microprocessor that supports Python, OpenCV, H.264 encoding, and Raspberry Pi camera modules, the **Raspberry Pi Compute Module 4 (CM4)** is an excellent choice. The CM4 offers similar capabilities to the Raspberry Pi 4 but is designed for embedded applications, making it ideal for custom PCB design.

Why the Compute Module 4 (CM4) Suits Your Needs:

1. **Modularity and Customization:** The CM4 comes without a PCB, so you can integrate it directly onto your own board. It includes the Broadcom BCM2711 SoC (same as the Raspberry Pi 4), making it capable of running Python, OpenCV, and H.264.

2. **Camera Compatibility:** The CM4 includes two MIPI CSI-2 camera connectors, which are compatible with Raspberry Pi camera modules, allowing direct use of high-quality camera modules for computer vision tasks.
3. **Variants:** You can choose CM4 versions with different RAM (1GB, 2GB, 4GB, or 8GB) and eMMC storage options or with microSD card support. This flexibility lets you optimize cost and performance based on your application requirements.
4. **Software:** The CM4 supports Raspberry Pi OS, which means Python, OpenCV, and Visual Studio Code can run on it just as they would on a standard Raspberry Pi. The hardware video encoder/decoder also supports H.264, ideal for video applications.
5. **PCB Design Resources:** The Raspberry Pi Foundation provides design files, schematics, and reference designs for the CM4, making it easier to integrate into a custom PCB.

Would you like suggestions on any specific resources for PCB design with the CM4?

Prompt 3

Is it possible to just get the chip?

Outcome 3

The exact Broadcom BCM2711 chip (the core chip used in the Raspberry Pi 4 and Compute Module 4) isn't available separately for purchase to the general public. Broadcom generally sells this chip directly to large manufacturers in bulk rather than individual buyers.

However, here are a few alternatives if you're looking for a similar processor setup:

1. Raspberry Pi Compute Module 4 (CM4)

- This is the closest option to getting the chip itself, as it's a small board that can be integrated onto a custom PCB.
- CM4 essentially is the BCM2711 chip mounted with essential components like RAM, storage options, and connectors, giving you a barebones setup.

2. Alternatives with Similar Capabilities

- **NVIDIA Jetson Nano Module:** A more powerful alternative for computer vision tasks with support for Python, OpenCV, and H.264. It's designed to be embedded in custom PCBs.
- **BeagleBone AI-64:** This is a single-board computer with capabilities similar to the Raspberry Pi but is more open for customization, including for embedded applications.

3. Other ARM-Based Microcontrollers with OpenCV Support

- Some microcontrollers from STMicroelectronics or ESP32 family can run lightweight OpenCV and Python-like environments (MicroPython), though they may not meet all your requirements for performance.

The **Compute Module 4** is likely your best bet for creating a custom PCB while maintaining compatibility with the Raspberry Pi ecosystem. Would you like guidance on CM4 design resources for PCB integration?

Prompt 4

How can one create a pcb with the compute module 4 integrated

Outcome 4

Creating a PCB with the Raspberry Pi Compute Module 4 (CM4) integrated involves several stages, from planning the design to testing the final board. Here's a step-by-step guide to help you get started:

1. Define the Requirements

- **Functionality:** List out all the functionality you need, such as power supply, USB, HDMI, networking, GPIO, camera, and display interfaces.
- **Ports and Interfaces:** Decide which interfaces you want to expose, such as USB, HDMI, Ethernet, or GPIO pins.
- **Power Requirements:** Determine your power requirements and decide on the power source (e.g., USB-C, barrel jack).

2. Get the Design Files and Schematics from Raspberry Pi

- The Raspberry Pi Foundation provides **CM4 schematics**, **mechanical drawings**, and **reference design files** (available in KiCAD, a free PCB design software). These files will help you understand the CM4 pinout and connections.
- You can download these resources from the Raspberry Pi documentation website.

3. Choose a PCB Design Tool

- **KiCAD:** Free and open-source, with the CM4 reference designs already available.
- **Altium Designer:** A more advanced tool, often used in professional settings, but requires setting up the CM4 components.
- **Eagle PCB:** Has a large user base and supports many components, although it may require importing the CM4 design.

4. Plan the PCB Layout

- **Position the CM4 Socket:** Place the **100-pin and 50-pin connectors** (J6 and J7) that the CM4 uses on your board. These connectors provide access to power, GPIO, I2C, SPI, UART, HDMI, and other interfaces.

- **Place Critical Components:** Start with components that connect directly to the CM4, such as HDMI connectors, USB ports, and Ethernet.
- **Separate Power and Signal Layers:** Use separate PCB layers for power and signal paths to avoid interference.

5. Power and Grounding

- **Power Supply Design:** The CM4 requires a **5V power supply** and can draw substantial current, so ensure that your power supply can handle the load (up to 2A or more).
- **Grounding and Noise Management:** Ensure a proper ground plane, especially around high-speed interfaces like HDMI or USB, to minimize noise.

6. PCB Routing

- **High-Speed Signals:** Route high-speed signals (like HDMI, USB, and camera interfaces) with care. Follow differential pair and impedance matching guidelines, especially for HDMI and USB signals.
- **GPIO and Peripherals:** Route GPIO and other peripheral signals as required, keeping paths short and minimizing cross-talk for reliable signal integrity.
- **Thermal Management:** If the design has power-intensive components, consider adding heat dissipation features (e.g., copper pours, thermal vias).

7. Testing and Prototyping

- **Build a Prototype:** Many PCB manufacturers offer low-cost prototypes for initial testing.
- **Verify Power and Signals:** Test that the power circuits deliver stable voltage, and verify signal integrity for high-speed interfaces.
- **Run Software Tests:** Load Raspberry Pi OS and test key functionality like USB, HDMI, GPIO, and connected peripherals.

8. Manufacturing

- Once the design is finalized and tested, you can manufacture the PCB in bulk. Many PCB manufacturers offer assembly services, allowing you to place the CM4 connectors directly on the board.

Tools and Resources

- **Compute Module 4 IO Board:** This development board serves as a reference design and can be used to prototype or test configurations.
- **Documentation and Community:** The Raspberry Pi forum and community are valuable resources for advice and troubleshooting.

Would you like details on specific tools or techniques for routing and signal integrity?

GPTG

Prompt 1

Do you remember my project about free space surveillance?

Outcome 1

Yes, I remember! Your project, iSPY, is a secure, computer-vision-capable object detection surveillance system that uses free-space optics (FSO) to relay real-time video. It leverages computer vision for object detection, and you're working with the Raspberry Pi Compute Module 4 as the MCU in a custom PCB design, running OpenCV for video compression and decompression.

Is there something specific you'd like to discuss or need help with?

Prompt 2

I'd like to find an MCU that can handle the decompression of the transmission from bitstream and then once decompressed, display the live feed onto a display. The process of the entire project will start with a raspberry pi model 4 b that is running the openCV computer vision (with a camera module connected) and the video compression using H.264. OpenCV and H.264 are utilized in python. Once the data is compressed, it will transmit the data in bitstream via freespace (like a laser or light). The second PCB is a custom PCB with a yet to be determined MCU. The second PCB should receive the transmission via a receiver. Then H.264 will decompress the data and display the live feed to a display. I need to find a suitable MCU that can be used in our custom PCB to decompress the transmitted data and display it to a monitor. I know of good brands like ESP but don't know where to start with research.

Outcome 2

To find an MCU that can handle H.264 decompression and display the live feed to a monitor, it's essential to prioritize certain features, as H.264 decompression can be resource-intensive for microcontrollers. Here's a structured approach to start your research:

Key Considerations for the MCU

1. **H.264 Hardware Support:** Look for MCUs with built-in video processing or hardware acceleration for H.264, which will improve decompression performance without overwhelming the processor.
2. **Display Interface:** Ensure the MCU has native support for video output, ideally with interfaces like HDMI, MIPI DSI, or RGB that can easily drive displays.
3. **RAM and Flash Storage:** Decompression of H.264 video and video output require substantial RAM. Look for MCUs with at least 512MB to 1GB of RAM.
4. **Processing Power:** A more powerful processor (dual-core or higher, preferably ARM Cortex-A or Cortex-M series) will help with real-time video decompression and display.

Suggested MCU Families and Options

1. ESP32-S3:

- The ESP32 series is very popular, and the **ESP32-S3** has AI-optimized features and enough processing power for simpler video handling tasks. However, it lacks built-in hardware acceleration for H.264, so it may be challenging for real-time decompression and display tasks without significant optimization.

2. NXP i.MX Series:

- The **i.MX6ULL** or **i.MX8M Mini** are part of NXP's ARM-based series, designed for multimedia applications. They offer H.264 decoding capabilities and support for high-resolution displays via MIPI-DSI or HDMI interfaces, making them strong candidates for video processing.

3. STM32 MP1 (STM32MP157):

- This MCU from STMicroelectronics has a Cortex-A7 core that can handle complex tasks like video decompression. It supports external RAM, which you can configure based on your memory needs, and has an LCD controller to drive displays, making it suitable for handling the video display portion.

4. Raspberry Pi Compute Module 4 (alternative):

- If size isn't the primary constraint, the **Raspberry Pi CM4** can serve as a compact, powerful alternative that natively supports H.264 decoding and display output. It would simplify development as it already supports H.264 natively and integrates easily with HDMI displays.

Next Steps

1. **Evaluate Processing Benchmarks:** Compare benchmarks for H.264 decompression on each MCU candidate to see if they can handle real-time playback.
2. **Prototype with Evaluation Boards:** Before committing to custom PCB design, testing with an evaluation board for your selected MCU will help confirm performance and compatibility with your display.

The **NXP i.MX** series or **STM32 MP1** family are likely top options, given their multimedia support and display interface options. Let me know if you'd like more detailed specs or assistance on a specific MCU!

Prompt 3

I need to make sure it is a chip that can be integrated into a custom PCB. Are these chips?

Outcome 3

Yes, all the options I mentioned are available as chips that can be integrated into a custom PCB, as opposed to just development boards. Here's a breakdown of their suitability for integration into your custom PCB:

1. ESP32-S3 (and ESP32 Series in General)

- **Chip Form:** Available as a standalone chip (e.g., ESP32-S3-WROOM or ESP32-S3-MINI), which can be directly integrated into a custom PCB.
- **Considerations:** While the ESP32-S3 offers significant processing power, it might struggle with real-time H.264 decompression. However, if you decide to explore software-based decompression on a less resource-intensive stream, it could still be a viable option.

2. NXP i.MX Series

- **Chip Form:** The **i.MX6ULL** and **i.MX8M Mini** are offered in various chip package types suitable for custom PCB integration.
- **Considerations:** NXP's i.MX series is popular in embedded systems where multimedia processing is needed, making it well-suited to handling H.264 decompression and output to displays. These chips may require more complex PCB design due to their high pin count and need for power management.

3. STM32 MP1 Series (STM32MP157)

- **Chip Form:** Available in chip form (e.g., STM32MP157AAC), specifically designed for integration into custom PCBs.
- **Considerations:** STMicroelectronics' MP1 series offers both ARM Cortex-A and Cortex-M cores in a single chip, providing ample processing power for tasks like H.264 decompression. It also includes an LCD controller for display integration, which simplifies video output.

4. Raspberry Pi Compute Module 4 (CM4)

- **Module Form:** The CM4 itself is a module rather than a single chip. While it is not exactly a traditional "chip," it is designed specifically for custom integration and can be treated as an embedded module in your custom PCB.
- **Considerations:** Using the CM4 would require a custom carrier board that includes connectors for display output, power regulation, and peripherals. This module has built-in H.264 support, making it very convenient for video applications, though it occupies more board space than traditional chips.

Final Thoughts

For a single-chip solution, the **NXP i.MX6ULL/i.MX8M Mini** or **STM32 MP1 (STM32MP157)** are likely your best options, as they provide the H.264 support and display interfacing capabilities in a compact chip form factor.

Would you like specific details on pin configurations or required support components for any of these chips?

Prompt 4

Can you provide links to these chips so I can research the product?

Outcome 4

To support H.264 decompression and video output for your surveillance system's custom PCB, the NXP i.MX 6ULL processor is a robust choice. This processor features a single Arm Cortex-A7 core clocked up to 900 MHz, optimized for efficiency, which aligns well with your project's power and performance needs. It includes integrated memory and multimedia interfaces compatible with displays and peripheral connections. Specifically designed for video and media applications, it offers support for multiple interfaces, such as camera sensors and display modules, making it highly suitable for real-time video decoding and display tasks required in your setup.

For further details, you can review the technical specifications and options for the i.MX 6ULL directly on NXP's product page

[NXP® Semiconductors Official Site](https://www.nxp.com/semiconductors/processors/arm-cortex-a7/imx6ull)

[ps://www.nxp.com/assets/block-diagram/en/i.MX6ULL.pdf](https://www.nxp.com/assets/block-diagram/en/i.MX6ULL.pdf)). Additionally, the NXP i.MX6ULL Evaluation Kit provides development support that may assist with prototyping and testing for your custom board.

Prompt 5

Is STM32MP157AAC3 one of the chips you were discussing?

Outcome 5

Yes, the **STM32MP157AAC3** is indeed one of the chips in the STM32MP1 series, specifically designed for multimedia applications and capable of handling complex processing tasks like H.264 decompression. This series from STMicroelectronics includes an Arm Cortex-A7 core paired with a Cortex-M4 core, making it well-suited for applications requiring both high processing power and real-time control, which is ideal for your project involving video decompression and display.

The STM32MP157AAC3 also supports various peripherals, including display interfaces (like LCD controllers), memory interfaces, and general-purpose I/O, which simplifies connecting external displays and sensors directly on a custom PCB. The chip is available in multiple package options to fit different PCB layouts and designs.

For further technical details, you can check STMicroelectronics' official STM32MP157AAC3 product page or refer to the STM32MP1 series documentation for a broader overview.

Prompt 6

If you had to pick between the stm and the nxp, which one would be most suitable>

Outcome 6

Choosing between the **STM32MP157AAC3** from STMicroelectronics and the **i.MX6ULL** from NXP depends on a few key aspects of your project requirements, such as performance needs, ease of development, and specific multimedia requirements.

Comparison of Key Features

1. Video Processing and Display Support

- **STM32MP157AAC3:** Includes an ARM Cortex-A7 core with multimedia support and a display controller capable of driving an LCD directly. It supports an external memory interface, which is useful for handling video data, and offers flexibility for applications needing real-time processing and control.
- **i.MX6ULL:** The i.MX6ULL is optimized for general multimedia applications and is a more cost-effective option within NXP's lineup. While it can support video processing, it does not natively include a high-performance GPU or display controller like some other i.MX chips (e.g., i.MX8), which could limit complex graphical or high-framerate applications. However, for H.264 decompression and moderate-resolution display tasks, it should still be capable.

2. Performance and Power Consumption

- **STM32MP157AAC3:** Offers dual-core processing (Cortex-A7 + Cortex-M4) which gives it an advantage in scenarios where real-time control is crucial alongside media processing. It also has low power consumption, making it efficient for embedded applications that require moderate power efficiency with high performance.
- **i.MX6ULL:** The i.MX6ULL is a single-core Cortex-A7, designed to be highly power-efficient and cost-effective but may not handle high-performance tasks as efficiently as the STM32MP157.

3. Ecosystem and Support

- **STM32MP157AAC3:** STMicroelectronics has a strong ecosystem with the STM32 family, making development support, libraries, and community resources more readily available. It has good Linux support, which may simplify development with OpenCV and H.264 decoding if you're working within the Linux ecosystem.
- **i.MX6ULL:** NXP's i.MX series is also well-supported, especially in industrial and multimedia applications. The i.MX6ULL has robust Linux support and a range of application notes and reference designs, though it may have slightly fewer ready-made resources than STM32.

4. Cost and Complexity

- **i.MX6ULL:** Generally, it is more cost-effective and available in simpler configurations, making it ideal for projects focused on cost efficiency and straightforward multimedia tasks.

- **STM32MP157AAC3:** This is a more advanced and somewhat more costly option but offers flexibility and additional processing power for complex or multitasking applications.

Recommendation

For your project:

- If **power efficiency and budget** are top priorities, and if you're confident that single-core processing will meet your needs for H.264 decompression and display, the **i.MX6ULL** is a solid choice.
- If you need **higher processing capabilities** and real-time control in addition to multimedia support, the **STM32MP157AAC3** would likely be a better fit.

In summary, **STM32MP157AAC3** may offer more flexibility and power for handling both the decompression and display requirements if your project demands high performance and low latency. For simpler applications, the **i.MX6ULL** is a cost-effective, capable choice.

Appendix G – Zemax Images

Image 1: Design from book “Designing Optics Using Zemax”

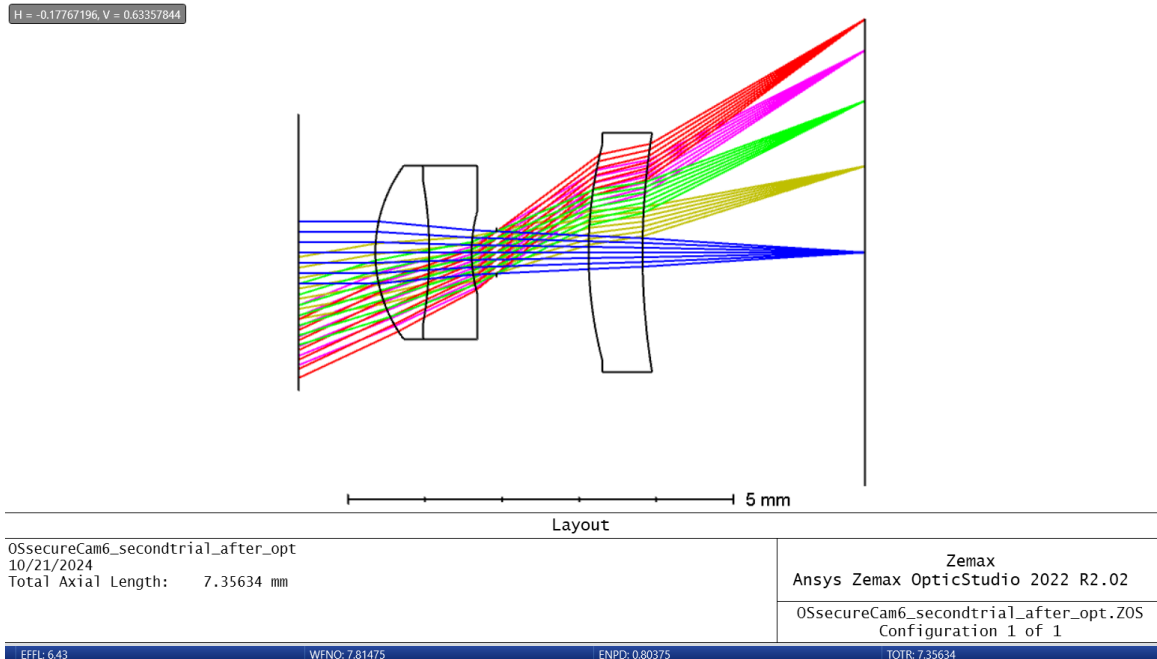
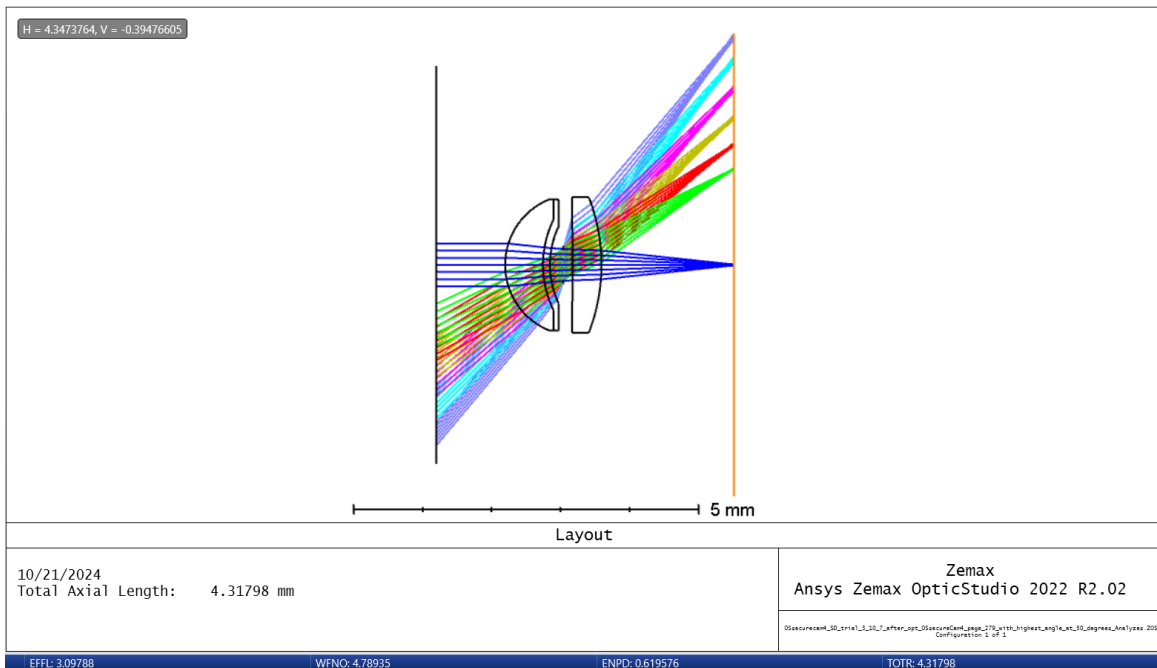


Image 2: Author’s own design following framework from “Designing Optics Using Zemax”



Appendix H – Computer Vision

Image 1: PGI Architecture from “YOLOv9: Advancing the YOLO Legacy.”

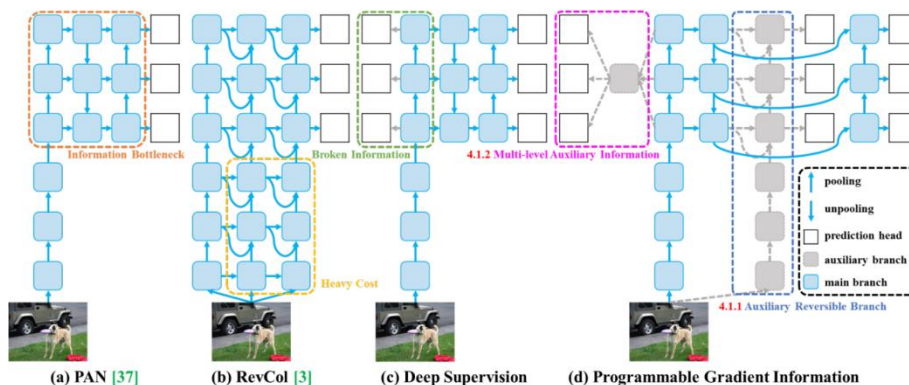


Image 2: GELAN Architecture from “YOLOv9: Advancing the YOLO Legacy.”

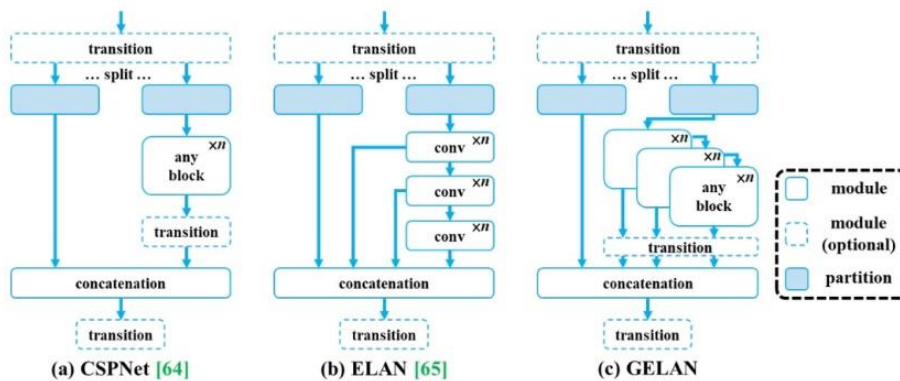


Image 3: Comparison of State-of-the-Art Real-Time Object Detectors from “YOLOv9”

Detection (COCO)		Segmentation (COCO)			
Model	size (pixels)	mAP ^{val} ₅₀₋₉₅	mAP ^{val} ₅₀	params (M)	FLOPs (B)
YOLOv9t	640	38.3	53.1	2.0	7.7
YOLOv9s	640	46.8	63.4	7.2	26.7
YOLOv9m	640	51.4	68.1	20.1	76.8
YOLOv9c	640	53.0	70.2	25.5	102.8
YOLOv9e	640	55.6	72.8	58.1	192.5

Image 4: YOLOv10 Model Variations from “YOLOv10: Paper Explanation and Inference Results.”

Model	Input Size	AP ^{Val}	FLOPS (G)	Latency (ms)
YOLOv10-N	640	38.5	6.7	1.84
YOLOv10-S	640	46.3	21.6	2.49
YOLOv10-M	640	51.1	59.1	4.74
YOLOv10-B	640	52.5	92.0	5.74
YOLOv10-L	640	53.2	120.3	7.28
YOLOv10-X	640	54.4	160.4	10.70

Image 5: YOLOv11 Model Variations from “YOLO11 NEW.”

Model	size (pixels)	mAP ^{val} ₅₀₋₉₅	Speed CPU ONNX (ms)	Speed T4 TensorRT10 (ms)	params (M)	FLOPs (B)
YOLO11n	640	39.5	56.1 ± 0.8	1.5 ± 0.0	2.6	6.5
YOLO11s	640	47.0	90.0 ± 1.2	2.5 ± 0.0	9.4	21.5
YOLO11m	640	51.5	183.2 ± 2.0	4.7 ± 0.1	20.1	68.0
YOLO11l	640	53.4	238.6 ± 1.4	6.2 ± 0.1	25.3	86.9
YOLO11x	640	54.7	462.8 ± 6.7	11.3 ± 0.2	56.9	194.9

Image 6: YOLO-NAS Model Variations from “Deci’s YOLO-NAS: Next-Generation for Object Detection”

Model	mAP	Latency (ms)
YOLO-NAS S	47.5	3.21
YOLO-NAS M	51.55	5.85
YOLO-NAS L	52.22	7.87
YOLO-NAS S INT-8	47.03	2.36
YOLO-NAS M INT-8	51.0	3.78
YOLO-NAS L INT-8	52.1	4.78

Appendix I – Video Compression Images

Image 1: Image Depicting Spatial Compression (Intraframe Compression)



Intraframe Compression

Every frame is encoded individually

Image 2: Image Depicting Temporal Compression (Interframe Compression)



Interframe Compression

Only the differences between frames are encoded

Image 3: Assigned Bit Lengths of Letters in Sequence

$S = 000$ (3 bits)

$A = 001$ (3 bits)

$C = 01$ (2 bits)

$K = 1$ (1 bit)

Image 4: Demonstration of PCM Output

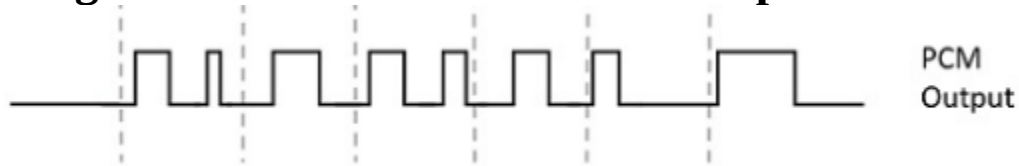


Image 5: Image of Raspberry Pi Compute Module 4



Image 1: Power Delivery System



