AutoCaddie

An AI-Driven Smart Coach

Louis Deal (EE), Rylan Simpson (CpE), Jarett Artman (CpE) , Joshua Glaspey (CpE)
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING - GROUP 22

1 Executive Summary

Technology is reshaping every facet of our lives, the domain of sports and physical training remains ripe for innovation. Addressing this gap, a project led by Group 22, stands as a transformative solution aimed at revolutionizing the way athletes and fitness enthusiasts train, improve, and excel.

AutoCaddie leverages cutting-edge Artificial Intelligence algorithms to serve as a smart coach, offering real-time, individualized feedback for enhancing performance. Unlike traditional coaching methodologies that may be subject to human biases and limitations, AutoCaddie employs data analytics and sensor technologies to provide precise and actionable insights.

The project encapsulates a comprehensive development framework that spans across hardware integration, software engineering, and user interface design. Cutting-edge sensors collect a wealth of performance metrics, which are then processed through machine learning models to generate tailored coaching feedback. This represents a leap in technological capability, converging multiple disciplines to deliver a product that not only stands on its own merit but sets new standards for smart coaching solutions.

This document serves as an exhaustive guide detailing every aspect of the AutoCaddie project—from initial research and design constraints to testing protocols and final implementation. It incorporates an in-depth examination of relevant existing technologies, meticulous part selection based on empirical research, and a discussion of design standards to ensure compliance with industry norms.

The AutoCaddie project is not merely a technological advancement; it is a paradigm shift. It promises to elevate the quality of training and performance optimization to unprecedented levels, offering tangible benefits to athletes, coaches, and sports organizations alike. Therefore, it is poised to make a significant impact in bridging the technology gap in today's sports training landscape.

For those who wish to delve deeper into the technical and administrative facets of the project, the document includes extensive appendices and references, providing a wealth of information for further study and research.

2 Project Description

2.1 Project Background and Motivation

2.1.1 Introduction

One of the most challenging aspects of sports training and physical fitness is the lack of personalized, real-time feedback. Whether you're a professional athlete or a casual fitness enthusiast, the absence of immediate and accurate coaching can hinder performance and lead to a plateau in improvement. To address this problem, we introduce AutoCaddie, a cutting-edge AI-driven smart coaching system that aims to redefine the way individuals train and improve in their respective sports or fitness regimes.

AutoCaddie employs a comprehensive suite of sensors and data analytics tools to capture a wide range of performance metrics. These metrics are processed through sophisticated machine learning algorithms, which then provide real-time, personalized coaching feedback directly to the user. This system is designed to be integrated into existing training and can also function independently, offering a versatile solution adaptable to various sports and physical activities.

In addition to real-time feedback, AutoCaddie features a interface that provides a more in-depth analysis of the collected data. This includes the information of the current swing performed by the player and other data we collected from our camera footage. This feedback will be immensely valuable to the player.

Moreover, all the data captured by AutoCaddie can be securely stored and accessed remotely, serving multiple purposes. It can be used for further analysis, for more nuanced feedback, or even utilized for research and development in sports science.

By delivering a highly adaptable, precise, and user-friendly smart coaching solution, AutoCaddie aims to bridge the existing gap in personalized sports training. The system promises not only to enhance individual performance but also to contribute to the broader field of sports technology and analytics.

2.1.2 Motivation

As engineers with a passion for golf, we understand firsthand the challenges and limitations of traditional training methods in the sport. The existing systems often lack real-time, personalized feedback essential for meaningful improvement. AutoCaddie emerged from our desire to integrate our engineering skills with our hobbies, aiming to address these gaps. By leveraging cutting-edge AI and sensor technology, we aspire to enhance not only our own golfing experience but also to make advanced, data-driven coaching accessible to golfers at all levels. Through AutoCaddie, we aim to elevate the standards of training, making the sport more engaging and rewarding for everyone involved.

2.1.3 Function of Project

The function of the AutoCaddie project is to capture and analyze the biomechanics of a golfer's swing using a combination of cameras and sensors. The system processes this data in real-time through machine learning algorithms to identify areas for improvement. The analyzed information is then immediately displayed on a computer screen in a user-friendly format, providing golfers with actionable tips to enhance their swing. This enables golfers to make instant corrections, fostering more effective practice sessions and accelerating skill development.

2.2 Project Goals and Objectives

2.2.1 General Goals

Develop an artificial intelligence program utilizing machine learning to provide multiple modes of feedback for a user's golf swing (using primarily a driver).

Utilize integrated hardware components to provide multiple data streams which can be utilized to improve feedback streams.

Design a PCB board capable of performing wireless transmission of simple numeric data to a transceiver connected physically to a laptop.

Complete firmware code which enables wireless transmission via UART, reception of IMU data via I2C, and power-saving via the endemic low-power modes contained in the MSP430 MCU.

2.2.2 Software:

Main Goal:

Develop an ML-driven AI to provide feedback to a user's swing that integrates both IMU data and video camera feeds. Feedback should come in the form of visual and audio feedback queues.

Objectives:

-Find a database to train the AI

GolfDB is a repository of golf videos from a set number of camera perspectives and is freely available to the public. These videos will be utilized to train the machine learning program to evaluate a swing from two different camera angles, one from the side and one from behind.

-Integrate video feed data to AI program

Two cameras (listed under budget) will be utilized to send video data to the laptop for AI analysis via a wired connection.

-Effectively train the AI to recognize swings and make comparisons

The AI program will receive video data via a cable and will compare the active video feed to the videos gathered from GolfDB, returning audio and visual feedback based on that comparison.

-Program the AI to effectively provide feedback based off of comparison

The machine learning program will be programmed such that it returns audio feedback (via a buzzer either on the player or the laptop to indicate proper arm straightness) and visual feedback (via a GUI displayed on the laptop screen).

-Integrate IMU sensor data with AI

Absolute position and calculated arm straightness (3 IMU position vectors will be included in a vector calculation to determine the overall angle from shoulder to wrist across the arc of the swing and will be compared to a threshold of acceptable arm angles) will be fed to the AI program to assist it in providing feedback.

-Design and integrate visual and audio feedback queues

An on-screen GUI designed by the machine learning team will be utilized to provide visual feedback based on the IMU data and analyzed active video feeds, while a buzzer will be used to indicate proper position of the arm.

2.2.3 Hardware

Main Goal:

Design a system which uses Inertial Measurement Units (IMUs) to measure the kinematics of the user's arm (absolute position, 3d position vector), and wirelessly relays this numeric data via a transceiver to another transceiver tethered to a computer.

Objectives:

-Select and acquire hardware components

As of now, the parts to be purchased are 3 Inertial Measurement Units (IMUs), 1 MCU (MSP430), a battery (HQRP 2200mAh), 2 cameras, 2 wireless transceivers (NRF24L01).

-Integrate I2C communication of IMUs with MCU chip

I2C, a hardware serial communication technology, will be used to extract the measurements provided by the IMUs and send them to the MCU.

-Integrate wireless communication between MCU and main processor computer.

UART, another serial communication mode, will be utilized to send the calculations from the MCU chip to the wireless transceiver on the PCB, and will be used to extract the received data on the laptop to the AI analysis program.

-Design mounting system for IMU sensors on user.

The IMUs will be integrated into a flexible sport compression sleeve, set into the surface of it at the wrist, elbow, and shoulder. The wires connecting the IMUs to the MCU will be fed through flexible tubing which will prevent the wires from inhibiting the ergonomics of the wearable system.

-Design IMU housing components for mounting system

The IMUs will be mounted into a compression sleeve using an adhesion method (likely Velcro) and a compression method to prevent their motion (likely elastic bands which can be tightened from a central knot).

-Design data collector/data transmission PCB

A system model has already been developed, but a lower-level model of the PCB will not be completed for some time:

-Design PCB housing for mounting system

The PCB will be housed within a 3d-printed, filament-based plastic housing designed to the specific shape of the PCB board. The battery will be contained within and will be activated with a switch to ensure that power can be conserved. The design will be such that it does not impede the motion of the player.

-Integrate video camera feed into main processor computer

2.3 Engineering Requirements and Constraints

2.3.1 Engineering Requirements Overview

The AutoCaddie project is an interdisciplinary initiative that converges the fields of mechanical engineering, computer science, and data analytics to construct a sophisticated, real-time golf swing analyzer. This senior design project aims to solve a specific problem in the realm of sports technology: providing immediate, actionable feedback to golfers to improve their swing.

The hardware setup involves a strategically positioned high-resolution frontal camera along with additional sensors. These devices are meticulously engineered to capture crucial metrics of a golfer's swing, such as speed, angle, and body posture. One unique feature of this system is the incorporation of a buzzer mechanism. This buzzer sounds at the appropriate moment, signaling the golfer to start their swing and thereby ensuring a standardized initiation point for all swing data collected.

The raw sensor and IMU data will be processed by a dedicated computing unit located near the golfer. This unit is equipped with machine learning algorithms designed to analyze the biomechanical aspects of a golf swing in real-time. The algorithms will offer immediate coaching advice based on the analyzed data, displayed on a nearby computer screen.

The screen's user interface is designed for simplicity and ease of use, ensuring that golfers can effortlessly understand and apply the real-time feedback provided. The feedback is tailored to highlight areas requiring immediate attention, allowing for quick adjustments and improved swings in subsequent attempts.

An essential hardware requirement for the project is the creation of a 3D-printed housing unit that attaches to the golfer's arm. This housing will serve as a secure mount for the circuitry and sensors. Additionally, a sleeve will be designed to house the sensors in a manner that allows for accurate data capture while maintaining comfort and freedom of movement for the golfer.

The AutoCaddie system is built to be adaptable, capable of functioning both as an integrated part of existing training setups and as a standalone unit. While the system does not allow for adjustments based on the training environment, it is engineered to be reliable and effective in a variety of standard golf training settings.

In essence, the AutoCaddie project embodies a harmonious blend of hardware and software engineering components. The project serves not only as a practical, real-world application for immediate, data-driven golf swing improvement but also as an invaluable educational experience for the team. It equips team members with the hands-on experience necessary for multidisciplinary engineering roles in their future professional endeavors.

2.3.2 Engineering Requirements Table

Requirement	Description	Reasoning
Cost	Total costs less than $500	System must be affordable for market entry and accessible for all players
Responsiveness	eedback within 10 seconds	Prevents long waiting time for feedback, including connection feedback systems for IMUs
Connection range	Stable connection from at least 20 meters of signal source	Requirements align with wireless data communication restrictions to send valid data for accurate feedback.
Accuracy	Feedback is accurate at least 90% of the time	Requirement sets standard for AI generated feedback, requiring reliable and desirable feedback.
Battery Life	5 hours battery life minimum of wireless transceiver	Requirements establish usability standards, enabling users to practice with the system for a meaningful duration.
Ergonomic	Wearable system is nonrestrictive and mountable in less than 5 minutes	Requirement establishes a usability standard for user-friendly system mounting.
Mounted Dimensions	Wireless transmission PCB designed in less than a 10cm-by-10cm frame	Requirement facilitates ergonomic design, following non-obstructing sizing for placement on user.
Weight	Wearable system weighs at most 5 pounds	Requirement facilitates ergonomic design, avoiding user fatigue of a weightier system.

2.3.3 Engineering Constraints

The design of AutoCaddie seeks to involve both direct-conditional and user-resultant AI-driven feedback. To achieve this design endpoint, the system is required to integrate both a senor system to measure and analyze user kinematics, and a video feed of the user for a machine learning algorithm to analyze the user's motions. The primary driving force behind the AutoCaddie system is the machine-learning based AI. This machine-learning based AI system must have some form of existing data to extrapolate off of, and due to time constraints, a pre-existing database. For these purposes, a database such as GolfDB shall be utilized. For accurate and consistent comparison of live data to the database, a video feed of the user must support at least a 30 frame-per-second video stream of at least a 420-pixel resolution. Furthermore, due to the fact that the user should practice their gold swings in a proper outdoor environment, the software environment must be constrained to a portable computer system, such as a laptop, which must be plugged in or have some form of steady charge available, as to not overly limit a user's use time.

Regarding the hardware, there are several constraints that should be put in place. First of all, to promote potential accessibility for users in-market, the total costs of the device should remain under 500 dollars. Equally as important, the design of the hardware sensor systems must be designed as to not impede the movement of the user while being as accurate as possible. The IMU sensors must be secured on the user and must be immobile with respect to the user's arm. Additionally, direct cable connections to the computer for data processing can interfere with user swings, so the data communications must be constrained to wireless communications that can be received by the laptop or other data processor's technology. These communications must also be constrained to a given data transmission speed floor (chosen to be ≥100kbps), so that the AI can utilize a sufficient amount of data within the time frame of a user's swing action. Finally, in parallel to how the software's computer environment must be portable, the cameras for a video stream of a user must also be portable, as to allow a user to take the system to a desired location for training. The full list of constraints is as follows.

Software:

- Existing database as training reference
- Video feed speed of ≥30fps
- Video resolution of ≥ 420p
- AI must be containable in a portable computer unit
- Computer must have a constant energy source

Hardware:

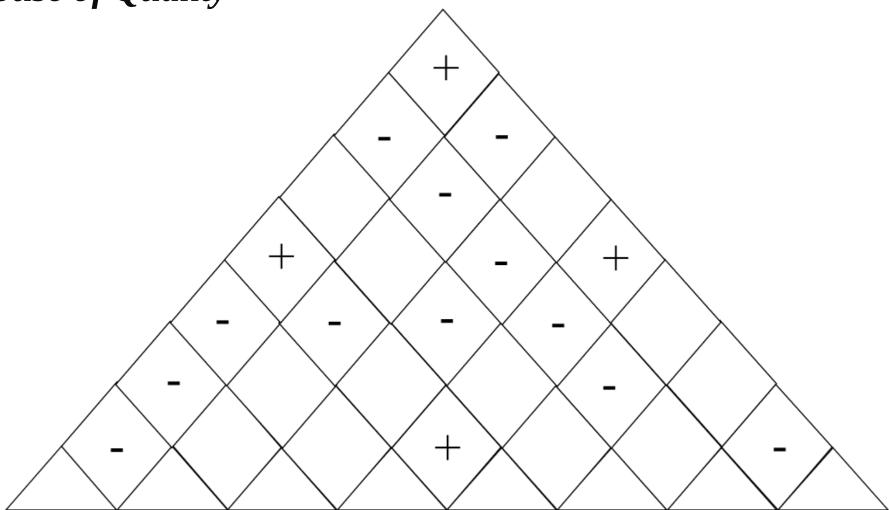
- Total costs must be < $500
- IMUs must be secured on user and immobile with respect to user arm
- User-mounted sensor system cannot restrict user swings
- Wireless communication standards must used and are receivable by computer
- Video cameras must be portable and can directly connect to a computer

2.4 House of Quality Analysis

2.4.1 User / Marketing Specifications

Affordability	< $500 price point
Battery Life	5 Hour Lifespan
Feedback speed	< 5 response generation time
Weight	< 5 pounds (On user)
Accuracy	> 90% feedback accuracy

2.4.1 House of Quality



		Costs	Battery Life	Responsiveness	Ergonomics	Weight	Accuracy	Connection Range	Mounted Dimensions	
		-	+	+	+	-	+	+	-	
Affordability	+	++	-		-	+		-	+	Minimal costs of system (<$500)
Battery Life	+	-	++		-	-		-	-	> 5 Hour battery lifespan
Feedback Speed	+		-	++			-			< 10 second response time
Weight	-	+	-		+	++		-	++	< 5 pounds total weight
Accuracy	+			-			++			> 90% correctness on feedback
		< $500	> 5 Hours	< 10 second Response time	< 5 minute user mounting	< 5 pounds	> 90% correctness on feedback	Stable up to 20 meters	< 10x10 cm	

As seen by the house of quality diagram, both the user/market specifications share some conditions, including that of battery life, weight, and accuracy. Regarding individual conditions of both user/market specifications and engineering requirements, both battery life and weight are the most influential. Increasing battery life requires either less powerful systems that draw less power, or requires the integration of a larger battery, thus increasing conditions such as weight and costs. Furthermore, reducing the weight of the system restricts the selection of both parts and overall design, possibly requiring the integration of more costly condensed application parts. Overall, most of the requirements or specifications are influenced by the hardware design.

Chapter 3: Research Related to Project Definition

AutoCaddie is a project that demands knowledge of data acquisition's technical and physical aspects. Now that the goals, constraints, and high-level analysis are determined, it is possible to move forward with research toward the technology to be utilized by this product.

Before diving into specific research, it paints a stronger basis to reference previous examples of similar projects. By observing the methods and implementations of previous endeavors, we can shed light on some of the problems that may arise throughout the development process. Not to mention, it can help give a bit of creativity by introducing new perspectives to a similar topic.

As an overview, there are a few areas of interest that are worth exploring. AutoCaddie's primary selling point is automating the coaching process of golf and allowing the process to be more convenient. For knowledge, it is worth investigating existing AI golf coach programs to determine different methods of implementation. For example, it's worth noting if other products rely exclusively on software or introduce other physical means of recording data. Additionally, it is worth investigating technology that utilizes IMU data via physical body tracking to determine linearity or other gyroscopic data. Other useful examples of projects that can help gain primarily fall into the reign of neural networks, which will be explored later.

3.1 Existing Similar Projects and Topic

This project is not necessarily the first of its kind. There are other existing AI golf coaches that provide user feedback to help improve technique. That being said, many of these products are sold at high prices, which takes away from the convenience of the system's design.

To explore this topic in general, it is important to reference previous examples that have investigated similar topics previously.

3.1.1 Smart Coaching

Coaching, overall, is the method of providing feedback to a user for improvement in some task. This can be done with verbal instruction, replication, or, effectively, statistical analysis. The takeaway from AutoCaddie is having a virtual AI coach provide useful feedback to the user in the form of processed data analysis on a live video feed.

Generally speaking, a program, controlled by some governing method, is trained on data that is directly related to an area of improvement that a user is requesting. For golf, this can relate to driving down a field. For tennis, this can relate to an initial service motion. There are infinitely many examples of where it is applicable, but some of these products in action are as follows.

One example, coach2vec [1], uses AI to understand the style of play for professional coaches in soccer. Its design gathers data from ball possessions, or periods where the ball is being controlled by the observed team, and gathers a list of statistics collectively from the players to map similarities to specific coaches. For example, observing the duration of possessions, on average, might signify a specific context of coaching style. The impressive aspect of this design is the usage of "k-Means" similarity to map soccer attributes to a matrix table and query specific details from coaches. In AutoCaddie, we need to establish a direct mapping between a specific training database to a live video feed. However, this process will not directly call for matrix decomposition due to the nature of the parameters. We can observe the behavior in a linear fashion and do 1:1 comparisons rather than query for related areas of interest.

Another example, Singlearity-PA [2], utilizes a large database of baseball games to enhance predictive modeling. The use of neural networks, trained on extensive pitch-by-pitch Statcast data from several years, allows for a more accurate and versatile approach to predicting plate appearance outcomes. The purpose of this program is to counter the limitations of previous calculation-based models by accounting for large amounts of history data that can be overlooked by mere equations. As a result, it ends up outperforming previous result-guessing techniques. The purpose of AutoCaddie does not stray far from this concept. It utilizes a neural network to produce more accurate data than a linear algorithm could produce by a frame-by-frame analysis. Due to the sheer amount of training data, it has access to a broader range of references to classify input data, as well as has stronger predictive capabilities towards the input actions of a swing.

Taking topics from both of these projects, it is important to utilize a neural network to achieve fast and accurate analysis of a video feed. Where algorithms can be limited by an absence of edge-case factors in a specific equation, neural networks shine by referencing large quantities of training data to predict results. Having the GolfDB database of videos allows for accurate coaching in golf that would be limited by normal analysis alone.

3.1.2 AI Golf Coaching

There are many existing smart golf coaching products, mostly composed of software-based assessments. Some examples of well-established products are ALFA Swing [3], 18birdies [4], and Sportsbox [5], which are all mobile app golf coaches. Additionally, there are many examples of research papers pertaining to the topics of computer vision, data analysis, and neural networks which are used throughout these programs. Of the three listed, notably all of them utilize solely video capture to produce feedback. Many of these products rely on the convenience of recording and advising through the same app. Although mobile app development is a consideration, for the sake of the neural network in question, AutoCaddie is planned to connect to a computer program with stronger cameras positioned around the user for precise data acquisition. Something that is lacking in these mobile programs is the reliability of the data, as different cameras present different framerates and image quality. ALFA Swing requires an iPhone X [6], or better to get accurate information, whereas AutoCaddie does not require anything outside of itself.

However, AI Golf [7], is unique in its research on self-training. As mentioned earlier, coaching can be executed in a variety of ways, replication being one of the primary methods. AI Golf establishes this connection and expands by directly comparing the user’s swing to a professional’s swing. The main difference between it and AutoCaddie is that it establishes its training data by combining a professional input motion sequence with output human poses to map to the user for specific feedback. AutoCaddie will preprocess incoming data into a specific form to match the training data for a direct comparison and provide predictive output based on the comparisons made to the database. However, an important topic worth referencing is discrepancy detection, or the ability of a program to identify differences between two separate feeds. In AutoCaddie, the preprocessing step needs to be capable of not producing discrepancies, but if so, it needs to be capable of recognizing that the video is of a different style rather than observed values being incorrect. Otherwise, false negatives can be generated.

In terms of hardware-based golf coaching, not as many products exist. SMARTGOLF AI Coach [8], is an AI golf coaching system that creates feedback by observation of data through sensors attached to the head of a golf club. It simulates a swing through the use of an interactive video game, and provides personalized feedback in real time in terms of ball trajectory, club face, and advice for specific swings such as slices and hooks. Although this technology is useful for coaching with spin and club face, we believe that this is accomplishable with a rear-facing camera and computer vision analysis. Therefore, this type of hardware is unnecessary to the finished product.

3.1.3 Real-Time AI Analysis

On another note, it is necessary to utilize the correct neural network model in order to provide proper real-time analysis. There are different ways to train a system such that it is hyperspecific to recognizing select parameters, which is necessary to create a smart golf coach AI.

One example of this is the usage of “Reservoir computing” [9] with image pattern recognition, but it is useful in its property of short-term memory. Projects have been instigated under this concept, such as experimentation with lowvoltage dynamic memristors, which naturally responds to low-voltage electrical pulses to exhibit short-term memory effects. However, due to the nature of AutoCaddie, exhibiting a short term memory system is not something that is necessary, since it will be observing all aspects of the data acquisition process and comparing it to a database. Nevertheless, understanding appropriate models and examples as to why specific use cases may yield unwanted results is important for the aspect of this project.

An application of real-time neural network processing is the Guitar Amplifier Emulation with Deep Learning [10]. This product explores the development and evaluation of deep neural network models for simulating audio distortion within guitar amplifiers. Notably, it explores two different models of interest, one being the recurrent neural network that was introduced in the last paragraph. The benefit of using this model is its boost in processing speed. However, a version of “WaveNet” was also utilized, which is a deep generative neural network used specifically for waveforms - which was developed by DeepMind [11]. It is designed to optimize hyperparamaters such as layer depth and activation functions. This model worked well when a large layer version was implemented. AutoCaddie aims to follow a similar model structure as WaveNet in that it can focus on specific hyperparamaters of a golf drive and provide convincing feedback. However, generative AI is a stretch of a topic since it generates new data given input feeds, whereas the goal of this project is to evaluate the given data.

Closer related, another project introduces a deep convulutional neural network for profile analysis of big powder diffraction data - known as the Paramater Quantification Network (PQ-Net) [12]. It can analyze live data from powder X-ray diffraction patterns from multi-phase systems, and create predictions for relevant information such as scale factors, lattice paramaters, among others. Most notably, the system’s performance was evaluated across different test datasets, and it was concluded that the PQ-Net provides faster and more accurate analyses than previous algorithmic methods. The network itself is designed as a deep convulutional neural network, which is designed well for image-related tasks. AutoCaddie mirrors this concept very closely with analyzing live image data and providing insightful feedback, so this architecture is one that can be referenced during the design of the AI. Specifically, PQ-Net is trained via patterns with known parameters, and learns to predict the expected behavior. Then, when given live data, the same process is used to create conclusions about the data. Many other low-level concepts from this project, such as minimizing the mean absolute error (MAE) loss function when training, incorporating dropout layers to prevent overfitting, and implementing deep ensembles are all important characteristics of improving performance in an algorithm as large as AutoCaddie's aims to be.

Overall, understanding differences between neural network models and their applications can help narrow down similar concepts to be used within generating the design. There is no correct method of creating a deep neural network, so long as it is trained correctly. That is where models such as PQ-Net shine, in that they are designed to optimize hyperparameters, and the mathematical methods can be reused within the AutoCaddie project.

3.1.4 Putting It All Together

Overall, there are many existent technologies that give insight to some of the goals AutoCaddie aims to accomplish. Whether it be in terms of general coaching with AI assistance, custom-tailored golf feedback, or even optimizations via deep neural networks, there are many applicable examples. However, AutoCaddie will combine the best aspects of all of these areas in order to deliver the highest performance result.

Having the correct feedback given by an AI is necessary for the system to function properly. Creating the illusion that a system is personalized to the same level of feedback as a professional coach will help draw out the best results. A similar concept to this is the Turing Test [13], which is a physiological study proposed by Turing in 1950 in which a machine passes if it can convince a person that it is not a program, but rather a real person. If AutoCaddie can establish accurate feedback on major components of the golf stroke, then the effectivity of the product will rise.

This accuracy will be determinant on the effectiveness of the data analysis system. To boost results, AutoCaddie will implement a deep neural network that is trained on a collection of proper golf swing technique videos. By training a model on correct data, and developing consistent evaluation metrics, we can feed live data into the model and have it smartly calculate statistics and improvements.

The strongest means of implementation can be via both hardware and software. Mainly, cameras can be used to capture action sequences of the user during a swing, in a similar fashion to existing phone coaching applications. However, having sensors also positioned on the user can help with synchronizing the data, as well as capture temporal data that may be less accurate taken from a 30 fps video stream.

Therefore, AutoCaddie is targetted towards providing accurate data using the correct means of implementation.

3.2 Relevant Technologies

To effectively address the challenges and objectives of this senior design project, it is imperative to explore the relevant technologies that will play a role in its implementation. Specifically, the success hinges upon harnessing some of these cutting-edge advancements and integrating them into our design.

Throughout this section, we will explore some of these new technologies that will be relevant throughout this project by providing an overview of their capabilities, benefits, and alignment with our goals. By understanding these topics, we can make insightful decisions about major components within AutoCaddie.

3.2.1 Network Architecture

As previously established, GolfDB is a large database of golf swing videos which can be used as training data for a deep neural network model. However, knowing that its goal is to capture the full swing sequence, there is also research conducted to create a deep convoluted neural network (CNN) model to master the analysis of the process. By referencing the relevant topics throughout "GolfDB: A Video Database for Golf Swing Sequencing" [14], we can gain an insight on SwingNet, the general swing sequencing network.

Importantly, the structure of SwingNet rides on the employment of MobileNetV2 [15], which is a CNN based on an inverted residual structure and is well suited for mobile applications because of its "width multiplier" system that scales the number of channels per network layer. To capture the temporal information crucial for golf swing sequencing, SwingNet uses a sequence of feature vectors obtained by applying global average pooling to the final feature map in MobileNetV2, which is then fed into a Long Short-Term Memory (LSTM) network. By using regressional analysis on each layer, probabilities of specific actions taken throughout the sequence are estimated.

In other words, for the task of analyzing golf swing sequencing, SwingNet takes a sequence of RGB images, depicted by frames of a video feed, and maps them to a corresponding sequence of event probabilities. Then the corresponding feature vector is input to the LSTM, which utilizes a softmax algorithm to obtain the probabilities.

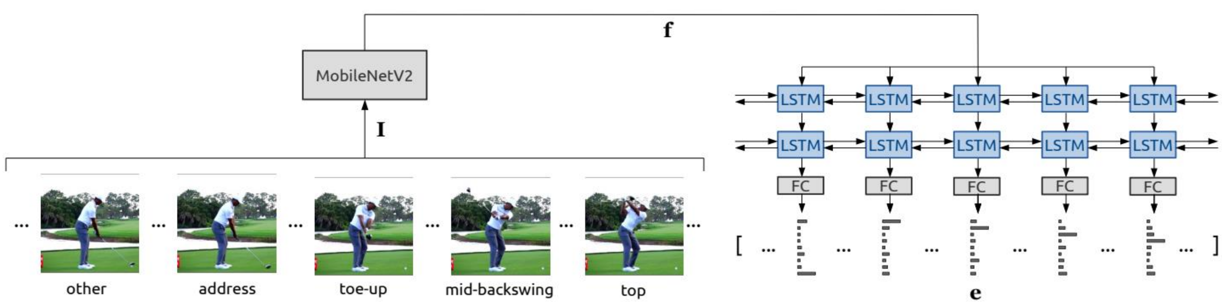


Figure [FIGURE NUMBER]: The Network Architecture of SwingNet. Variable **I** represents the sequence of RGB images which are mapped to a corresponding of event probabilities **e**. The feature vectors **f** are generated by MobileNetV2. Accessed via [16].

In terms of implementation, there are many different data preprocessing techniques discussed in the next section used to standardize the data. In general, the network itself it implemented using PyTorch [17], with batch normalization applied after each convolutional layer in MobileNetV2.

To improve the robustness of the model, SwingNet benefits from data augmentation techniques. This includes random image flipping, random selection of starting frames, and affine transformations, which help account for variations in camera angles and golfer handedness due to the nature of GolfDB extracting input from YouTube. The training of the model is performed using batch selection of frames, and the Adam optimizer method [18], using an initial learning rate of 0.001.

From this point forward, the model is set in place. Each swing input is divided into actions based on maximum likelihood estimations, and then analyzed through the network.

At a core level, AutoCaddie demands the same architecture. That being said, the focus of the feedback is set for digital output as a computer program rather than a mobile application, which demerits the need for MobileNetV2. Therefore, Python implementations of the training model will be explored in a future section. In terms of data processing, this methodology proves to be efficient. Referring to Figure [SWINGNET FIGURE], we will preprocess the video input into a collection of frame streams optimized to match the database. From there, we can use mapping to classify every action using event probabilities, then allow the implemented network to analyze it.

Therefore, for AutoCaddie, it is important to understand the hyperparameters in question, which is explained in [FUTURE SECTION HERE]. As demonstrated by SwingNet, classifying actions using event probabilities and a feature matrix will allow for stronger accuracy.

3.2.2 Live Video Analysis

Additionally, it is smart to reference work related to SwingNet in terms of live video analysis. Due to the nature of calling from the same video database for AutoCaddie, it is imperative to understand the method of standardizing video data to make the analysis transparent. To produce accurate results, our project must be able to replicate similar methods.

Mainly, computer vision needs to be applied to video data. As researched by Soonchan Park et al. [19], it is possible to develop key points throughout a recording by referencing pose analysis using depth tracking cameras. The algorithm being observed uses a dataset collected via a depth camera and motion capsure system to evaluate performance. Similar to neural networks, mean error (mE) are used to assess accuracy. Importantly, SwingNet complements this technology by referencing poses to attribute actions in the network analysis. So, for both AutoCaddie and SwingNet, there does not need to be depth tracking, as being able to capture poses within a swing sequence can help the neural network correctly attribute the motion to its evaluation reference.

Similarly, action recognition is a high-demand task, and involves predicting a single action for a video. In fact, this topic is being researched actively and has commonly seen the deployment of CNNs as a solution. Early work by Karpathy et al. [20] explored various methods for fusing temporal information from image sequences. This was extended by Simonyan and Zisserman [21] by incorporating an additional stream of optical flow information into the CNN-based model. From this point forward, expansions have been seen, such as 3-dimensional rendering of input data, but from a basic level, utilizing CNN is a successful method for action detection.

Furthermore, this topic can be divided into temporal action detection, which represents mid-level tasks, wherein the objective is to predict the start and end frames of actions within untrimmed videos. This is especially useful when trying to highlight specific hyperparameters within a live video feed, such as AutoCaddie will present. An effective strategy for solving this problem, proposed by Yeung et al. [22], employs reinforcement learning to develop a policy for identifying action boundaries. This makes the process dynamic and can allow for detection when only a fraction of the video frames is observable.

On a different note, in the context of video analysis, event spotting is the focus on identifying specific time instances when well-defined events occur within videos rather than predicting the temporal boundaries. This is a solution proposed by Giancola et al. [23] in response that the temporal extents of human actions within video datasets containing inheriting ambiguity, and thus the overall accuracy raising conflict.

3.2.3 Python Libraries

AutoCaddie will be trained on a neural network to provide accurate analyses of incoming data. In terms of architecture implementation, there are several libraries to select from. SwingNet referenced PyTorch, which is convenient due to it being readily deployable upon installation, having a strong mobile compatibility, and allowing for computational shortcuts that can still yield high reliability. In the following section, popular libraries will be contrasted to determine which is best suited for the mission of direct analysis of video and sensor data.

At a basic level, NumPy [24] is a python library that is strong at numerical operations. Specifically, it shines with matrix operations. It is often used for implementing the fundamental building blocks of neural networks, such as matrix operations and initialization of weights and biases. Therefore, it excels at low-level operations, since the majority of deep neural networks consists of matrix computations. However, it is limited in the fact that it does not have automatic differentiation capabilities, which is difficult to work with gradients. Not to mention, it requires manual construction of the network itself.

SciPy [25] is a library built on top of NumPy that provides additional computational functions, including optimization, integration, interpolation, and more. Although it's a popular library, it does not have anything more than NumPy in terms of neural network development. Especially because it does not modify the NumPy *linalg* package, it is not particularly suited for training models.

Pandas [26] is a powerful library for data wrangling, feature engineering, and data exploration. As discussed with SwingNet's implementation, much of the data undergoes massive preprocessing to normalize the input and standardize the process, such as establish bounding boxes and image scaling. Pandas provides support for many of these demands, which makes it a powerful tool for Pandas. However, its limitations are that it will not work as a neural network, since it does not have provided support for these operations. It excels at data manipulation, but not training.

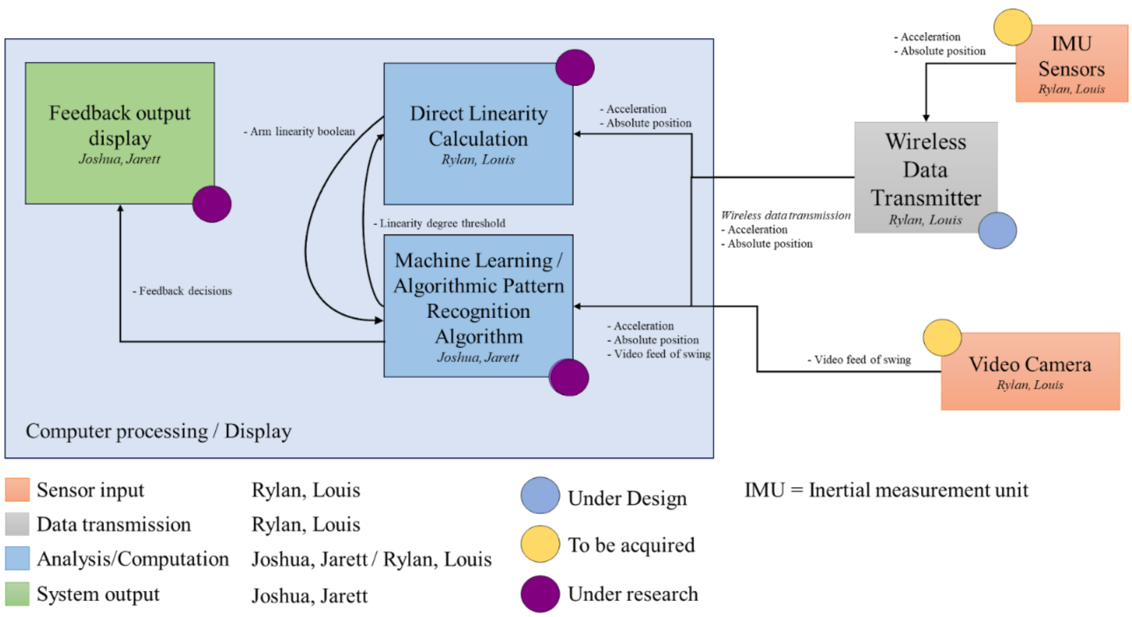
TensorFlow [27] is a deep learning framework that is known for its static computation graph and production-ready deployment capabilities. It offers high-level APIs and low-level control over model design, making it efficient at building, training, and deploying models. However, it is limited in that it is considered not user-friendly.

PyTorch [28] is another deep learning framework that is widely used for building, training, and deploying neural networks. It offers automatic differentiation, GPU support, and a rich ecosystem of pre-trained models and libraries. This makes it easier to develop with, and has renounced community support.

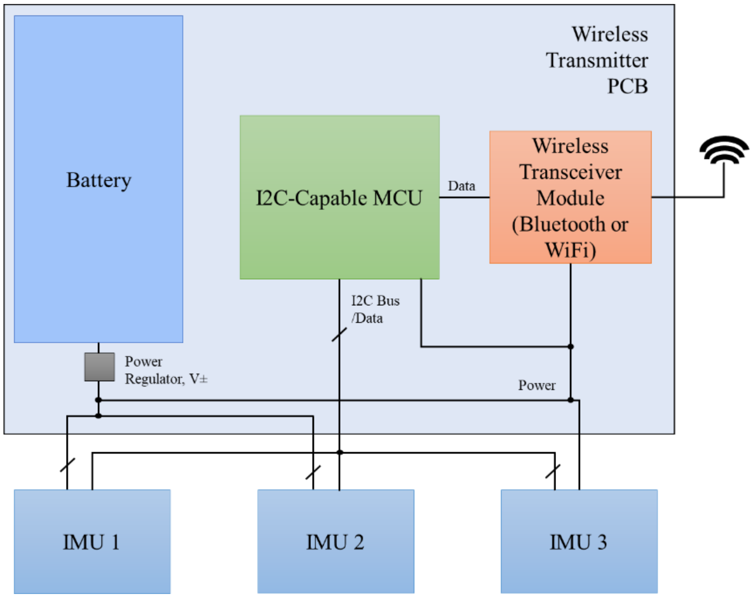
Putting all of these factors together, Pandas can provide many data preprocessing methods efficiently that can be used for preparing the video and sensor data to be mapped into the feature matrix discussed above. Therefore, it can be implemented as our approach to solve many of the tasks outlined in **3.2.2**. However, this is not to say that NumPy and SciPy will also not be referenced, as combining the libraries can allow for stronger optimizations – specifically matrix operations.

Additionally, due to the overwhelming community support, built-in GPU support, and proven effectiveness with SwingNet, PyTorch is the strongest option to implement the neural network model for AutoCaddie. To make sure we adhere to the power requirements in demand from **Chapter 2**, having GPU support is crucial as an optimization.

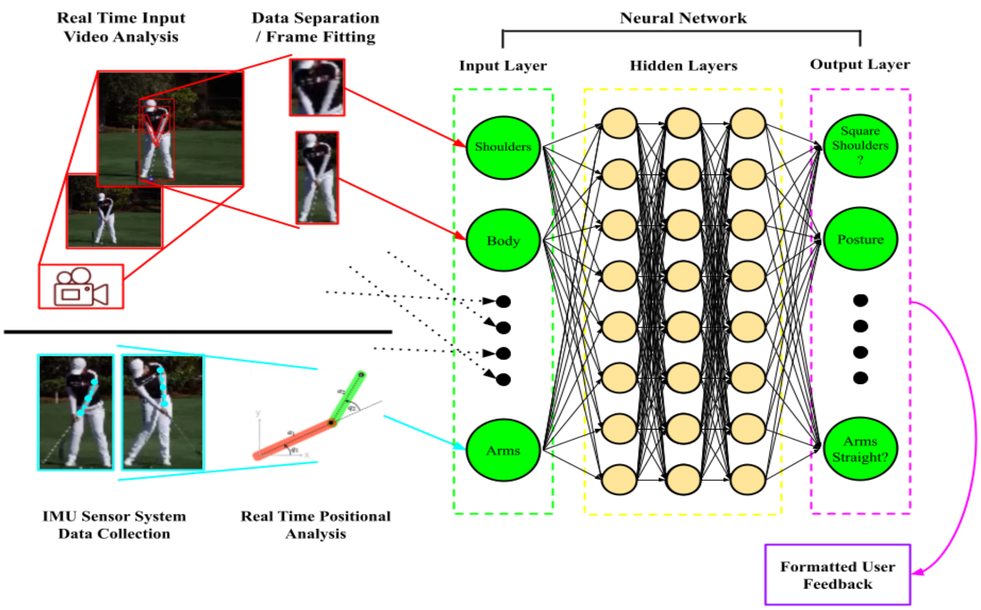
Chapter [A]: Diagrams and Functional Design



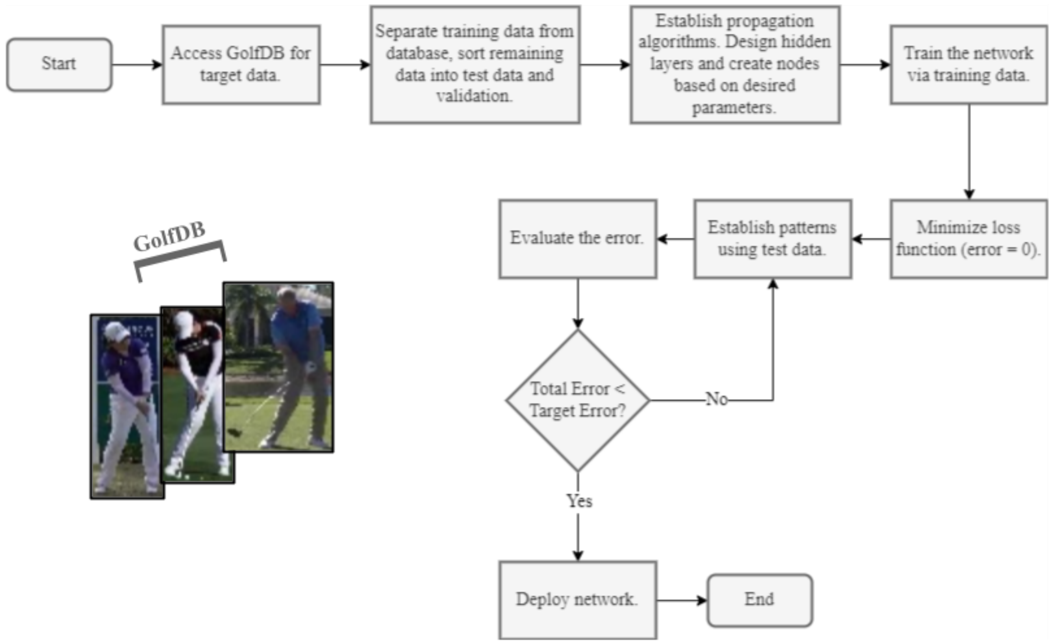
Flow chart / Block diagram of high-level, complete system processes



High-level circuit diagram of IMU data transmitter PCB circuit



Machine Learning Top-Down Design Concept



Machine Learning Training Concept Flowchart

[A].1: Real-time Video Analysis

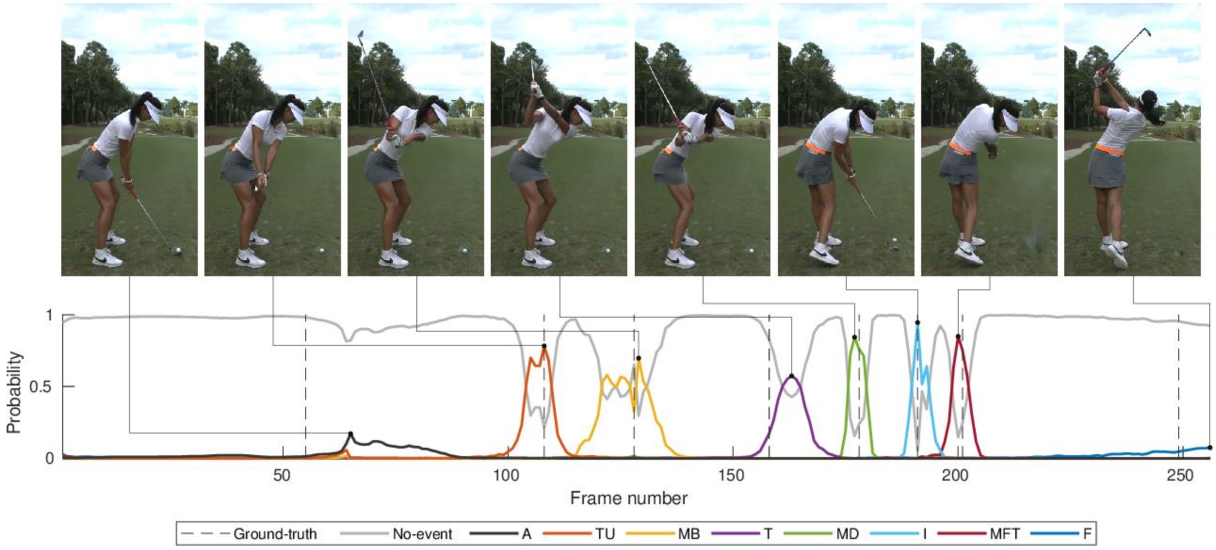
The AutoCaddie project introduces a camera-based approach to enhance golf swing analysis. This system comprises a dedicated high-speed camera setup strategically positioned to capture golf swings from a single angle. The camera's high frame rate ensures the precise recording of every phase of the swing, capturing intricate details that are crucial for accurate analysis.

The captured high-speed video footage undergoes advanced image processing algorithms. These algorithms are designed to meticulously extract key parameters critical to understanding the mechanics of the golfer's swing. Parameters of interest include the angle of attack, club face angle, and body posture. By tracking the trajectory and orientation of the golf club and the golfer's body, these algorithms provide insights into the alignment, timing, and coordination of the swing.

The strength of our approach lies in its ability to correlate the parameters extracted from the video frames with data obtained from wearable sensors. This fusion of visual and sensor data allows for a comprehensive analysis that transcends the limitations of individual data sources. By combining video-derived parameters with sensor readings, the system provides a holistic view of the golfer's swing mechanics, offering valuable insights into areas of improvement.

The camera-based system's capabilities enable real-time analysis. By processing video footage and parameter extraction in near real-time, golfers receive almost instant feedback on their swing mechanics. This immediate insight empowers them to make timely adjustments during practice sessions, fostering skill refinement.

[A].2: Artificial Intelligence Design



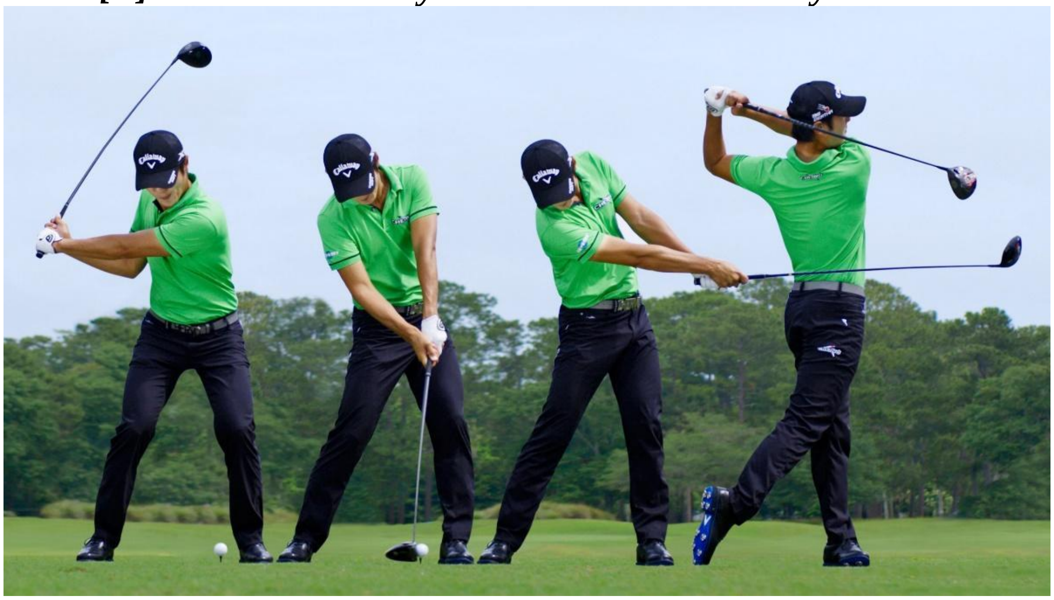
Probabilistic estimation of current action by frame. Base image courtesy of GolfDB: A Video Database for Golf Swing Sequencing[3]

In order to properly assign feedback to the user, a supervised neural-network model will be employed. First, the data is captured by an active camera, and fed through a live stream to the system. Next, the data is formatted using advanced image processing algorithms, which packages it to a readable format for every variable in the system's scope. Then the data is fed into a neural network, which has been trained specifically for the task of analyzing the swing.

As referenced before, GolfDB is a large collection of golf swings given in high resolution. AutoCaddie's neural network system will be trained from this database, which will give it access to many examples of proper technique with our variables in question, such as the angle of attack, club face angle, and body posture. For the hardware setup of this design, the camera will be positioned such that the reference angle for the database collection and our current physical setup are the same, so the only analysis that needs to be processed is the technical swing.

This system utilizes computer vision algorithms. For the data to be processed in real time, the data stream needs to match what is provided by its training data. There are many factors to consider, such as displacement from the stream location, frame count, and origin of the action. However, each of these problems is solvable through our approach. Action-started data tracking can be implemented via the information provided in the following research paper: "Temporal Action Localization in Untrimmed Videos via Multi-stage CNNs[4]". Also, frame resizing can be implemented by using interpolation algorithms to match the training data. Distance, along with 3D space, can also be computed by the network using computer-vision estimation algorithms, such as the one outlined in this paper: "Accurate and Efficient 3D Human Pose Estimation Algorithm using Single Depth Images for Pose Analysis in Golf.[5]"

[A].3: IMU Sensor System and Direct Linearity Calculation



Demonstration of IMU placement on the leading arm. Base image courtesy of howardsgolf.com

A primary mode of feedback given to golfers during swings is the maintaining of the straightness of their driving arm during the swing until after the follow-through stage of the swing. For the AutoCaddie to determine the straightness of the user's arm during the swing, a system of three IMU sensors will be used. Using the absolute position of the IMU sensors, three points in space are defined. Using one IMU sensor as an origin point, the other two IMU sensors' position data points can be defined as vectors from the "origin" IMU. If the angle between the two IMU position vectors is less than a defined threshold, the arm is deemed to be straight. If the angle is at or greater than a given threshold, the arm is considered bent and feedback will be given to keep the arm straight. The AI is used to define the optimal angle threshold for effective feedback and error anticipation, and controls the time frame for when the arm straightness is calculated, ending after the follow-through of the swing, as the arm may bend when the club's momentum is stopped.

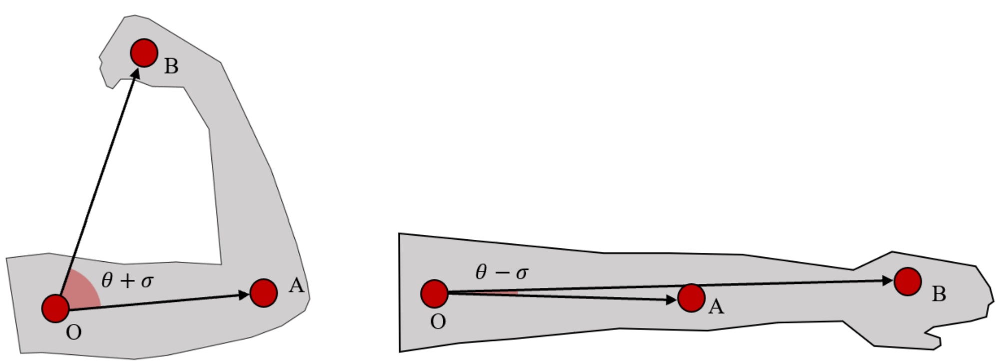


Figure A: Angle between point A and point B is greater than a defined threshold, θ. Arm is deemed bent.

Figure B: Angle between point A and point B is less than a defined threshold, θ. Arm is deemed straight.

Demonstration of arm linearity calculation results

As described earlier, the angle between the two IMU vectors from an "origin" in three-dimensional space can be used to determine the arm's state of straightness. Defined and potentially tuned by the AI for the user, the threshold angle, θ, is used to determine whether the arm is straight. To do so, the IMU placed near the user's shoulder is selected as the origin, and is used to translate the other IMU points relative to the new origin.

O: $\vec{O} - \vec{O} = \vec{0}$ **A**: $\vec{A} - \vec{O} = \vec{A'}$ **B**: $\vec{B} - \vec{O} = \vec{B'}$

From here, the dot product formula applied to three-dimensional vectors can be used to calculate the angle, σ, between the vectors.

$$\varepsilon = (\vec{A'} \cdot \vec{B'} / |A'||B'|)$$

If the calculated angle, ε, is less than θ, the threshold, the arm is considered straight, and if the angle ε is greater than θ, the arm is considered bent, and feedback to straighten the arm is given to the user by means of some audible or visual feedback mechanism.

Regarding the hardware system of AutoCaddie, the three-IMU system will be worn by the wearer on the leading arm of the swing. One IMU will be placed on the upper arm near the shoulder using a modified strap-mount, and another IMU will be worn on the forearm using the same mount type. The final IMU will be worn on the back of the hand, attached to a modified glove. Each of the IMU's wires will be fed through a flexible sheath and given enough slack to avoid impeding the movement of the wearer. The wires of the IMUs will be fed into a wireless transmitter secured on the back of the wearer, where the IMU data can be wirelessly transmitted using wireless transmitter modules to a nearby computer for the relevant calculations and feedback display to occur.

[A].4: PCB Design and MCU Integration

As has been described previously, the design of this project will, according to the requirements set forth, include a full PCB (Printed Circuit Board) including an MCU (Microcontroller Unit). The PCB itself has yet to be designed, but will certainly include the following (and tentatively include those marked with a "*"):

Component	Data Sheet	Price ($USD Per Unit)
MSP430FR6989IPZR (MCU)	MSP430FR698x(1), MSP430FR598x(1) Mixed-Signal Microcontrollers datasheet (Rev. D) (ti.com)	10.21
Wireless transceiver module [HC-05 RF]	HC-05 Datasheet.pdf (components101.com)	~4.20
Battery Terminals	N/A	*0.50
*Voltage Regulators	In-House Design	2.00

The purpose of the PCB attached to the golf player is to receive, arithmetically operate upon, and transmit the data collected by the IMU units placed along the arm. As is made clear by the MSP430FR6989IPZR Data Sheet (contained in the table above), the MCU is more than capable of performing the functions which will be required for arithmetic analysis (see Section 3). The HC-05 RF wireless transceiver interfaces seamlessly with the MCU which has been selected, and accepts UART connectivity, a simple serial communication mode which the MSP430 and all computers interface with well. The use of UART also ensures that data display will be facile, as the vast majority of home computers (as well as cell phones) have serial terminals which dynamically display the data received via UART periodically, according to the preset Baud rate (traditionally 9600 signal elements / second).

The PCB will not be designed until much more consultation has been undergone. The PCB is central to the success of this project, and therefore must be completed as effectively as possible. The following concerns exist with regards to the functionality of the PCB:

- Power
- Component Damage
- Component Housing
- Efficiency of Design
- Voltage Conversion/Regulation
- Unobtrusive Integration onto User

Appendix: Details

Estimated Budget

Item	Amount	Cost (Individual)	Cost (Total)
IMU [BNO055 IMU]	3	$ 34.95	$ 104.85
Wireless transceiver PCB	1	$ 30.00	$ 30.00 (EST)
Wireless transceiver module[HC-05 RF]	1	$ 7.99	$ 7.99
Battery: 7.2V / 2200 mAh	1	$ 19.95	$ 19.95
Camera	1	$ 129.00	$ 129.00
Wearable mount setup	1	$ 75.00 (EST MAX)	$ 75.00 (EST)
Electrical wire	1	$ 8.00	$ 8.00
TOTAL			**$ 374.79**

Project Milestones

First Semester

September 4th: Database resources found

October 1st: Artificial Intelligence methodology researched

October 16th: Artificial Intelligence methodology defined

October 1st: Hardware resourced and validated

October 30th: Hardware systems designed

Second Semester

Third Week: Direct linearity calculation systems complete

Fourth Week: Hardware systems integrated

Third Week: Artificial Intelligence vision system completed

Fifth Week: Complete Artificial Intelligence trained

Sixth Week: Feedback systems integrated

Eighth Week: Testing results gathered

Twelfth Week: Final product completed

Citations

1. P. Cintia, “Coach2vec: autoencoding the playing style of soccer coaches.” Accessed: Sep. 26, 2023. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/2106/2106.15444.pdf
2. G. Searle, “Singlearity: Using A Neural Network to Predict the Outcome of Plate Appearances,” *Baseball Prospectus*, Jul. 09, 2020. https://www.baseballprospectus.com/news/article/59993/singlearity-using-a-neural-network-to-predict-the-outcome-of-plate-appearances/ (accessed Sep. 26, 2023).
3. “Home,” *www.alfaswing.ai*. https://www.alfaswing.ai/ (accessed Sep. 26, 2023).
4. “AI Coach | 18Birdies,” *18birdies.com*. https://18birdies.com/aicoach/ (accessed Sep. 26, 2023).
5. “Sportsbox AI,” *SB*. https://www.sportsbox.ai/ (accessed Sep. 26, 2023).
6. P. Carroll, “Apple iPhone XR camera review: Top-ranked single-lens phone,” *DXOMARK*, Dec. 06, 2018. https://www.dxomark.com/apple-iphone-xr-camera-review/
7. C.-C. Liao, D.-H. Hwang, and H. Koike, “AI Golf: Golf Swing Analysis Tool for Self-Training,” *IEEE Access*, vol. 10, pp. 106286–106295, 2022, doi: https://doi.org/10.1109/access.2022.3210261.
8. “SMARTGOLF AI Coach | SMARTGOLF,” Jun. 28, 2022. https://smartgolf.biz/en/product/smartgolf-aix/
9. X. Zhu, Q. Wang, and W. D. Lu, “Memristor networks for real-time neural activity analysis,” *Nature Communications*, vol. 11, no. 1, May 2020, doi: https://doi.org/10.1038/s41467-020-16261-1.
10. A. Wright, E.-P. Damskägg, L. Juvela, and V. Välimäki, “Real-Time Guitar Amplifier Emulation with Deep Learning,” *Applied Sciences*, vol. 10, no. 3, p. 766, Jan. 2020, doi: https://doi.org/10.3390/app10030766.
11. “DeepMind - What if solving one problem could unlock solutions to thousands more?,” *Deepmind*. https://www.deepmind.com/
12. H. Dong *et al.*, “A deep convolutional neural network for real-time full profile analysis of big powder diffraction data,” *npj Computational Materials*, vol. 7, no. 1, pp. 1–9, May 2021, doi: https://doi.org/10.1038/s41524-021-00542-4.
13. G. Oppy and D. Dowe, “The Turing Test (Stanford Encyclopedia of Philosophy),” *Stanford.edu*, Apr. 09, 2003. https://plato.stanford.edu/entries/turing-test/
14. W. Mcnally, K. Vats, T. Pinto, C. Dulhanty, J. Mcphee, and A. Wong, “GolfDB: A Video Database for Golf Swing Sequencing.” Accessed: Sep. 27, 2023. [Online]. Available: https://arxiv.org/pdf/1903.06528.pdfhttps://arxiv.org/pdf/1801.04381.pdf
15. “MobileNet V2 — Torchvision main documentation,” pytorch.org. https://pytorch.org/vision/main/models/mobilenetv2.html (accessed Sep. 27, 2023).
16. Network Architecture of SwingNet. 2019. Accessed: Sep. 26, 2023. [Online]. Available: https://arxiv.org/pdf/1903.06528.pdf https://arxiv.org/pdf/1412.6980.pdf
17. PyTorch, “PyTorch documentation — PyTorch master documentation,” Pytorch.org, 2019. https://pytorch.org/docs/stable/index.html

18. D. Kingma and J. Lei Ba, "ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION," Jan. 2017. Available: https://arxiv.org/pdf/1412.6980.pdf

19. S. Park, J. Chang, H. Jeong, J.-H. Lee, and J.-Y. Park, "Accurate and Efficient 3D Human Pose Estimation Algorithm using Single Depth Images for Pose Analysis in Golf." Accessed: Sep. 27, 2023. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017_workshops/w2/papers/Park_Accurate_and_Efficient_CVPR_2017_paper.pdf

20. A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale Video Classification with Convolutional Neural Networks." Available: https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/42455.pdf

21. K. Simonyan and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos." Available: https://papers.nips.cc/paper_files/paper/2014/file/00ec53c4682d36f5c4359f4ae7bd7ba1-Paper.pdf

22. S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, "End-to-end Learning of Action Detection from Frame Glimpses in Videos." Accessed: Sep. 27, 2023. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2016/papers/Yeung_End-To-End_Learning_of_CVPR_2016_paper.pdf

23. S. Giancola, M. Amine, T. Dghaily, and B. Ghanem, "SoccerNet: A Scalable Dataset for Action Spotting in Soccer Videos." Accessed: Sep. 27, 2023. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2018_workshops/papers/w34/Giancola_SoccerNet_A_Scalable_CVPR_2018_paper.pdf

24. NumPy, "NumPy Documentation," numpy.org. https://numpy.org/doc/

25. "SciPy — SciPy v1.4.1 Reference Guide," Scipy.org, 2019. https://docs.scipy.org/doc/scipy/reference/

26. "DevDocs — pandas 0.25 documentation," devdocs.io. https://devdocs.io/pandas~0.25/ (accessed Sep. 27, 2023).

27. "Models & datasets," TensorFlow. https://www.tensorflow.org/resources/models-datasets

28. "torchvision.models — Torchvision 0.10.0 documentation," pytorch.org. https://pytorch.org/vision/stable/models.html