Sit & Chow: The Smart Dog Training Feeder

Andres Arzola, Andrew Balmes, Natalie Nguyen, and Maria Tran

University of Central Florida, Department of Electrical and Computer Engineering, Orlando, FL 32816

***Abstract* — Sit & Chow is an intelligent dog feeding system that integrates computer vision and embedded systems to automate dog feeding through behavioral reinforcement. Using an ESP32-S3 microcontroller and the OV5640 camera, the system performs real-time image processing to recognize when the dog approaches and assumes a sitting posture. Once this behavior is detected, the system triggers the motor-driven dispensing mechanism, transforming traditional timer-based feeding into a training tool. Additionally, the system allows for remote monitoring and control through a mobile application, improving usability while promoting safer and more consistent pet care.**

***Index Terms* — automated feeding, computer vision, embedded system, ESP32, mobile application, object detection, pet care.**

I. Introduction

Smart pet care devices have become increasingly popular as owners seek automated solutions for managing daily routines such as feeding. Conventional automatic feeders rely solely on time-based dispensing and do not incorporate behavioral awareness, limiting their ability to support training and reinforce desired actions.

Sit & Chow is an intelligent dog feeding system that integrates embedded hardware, computer vision, and cloud-based machine learning to enable behavior-dependent feeding. The system dispenses food only when a dog is detected in a sitting posture, combining scheduled feeding with real-time posture verification to reinforce obedience training.

This work demonstrates the integration of machine learning with embedded systems to create a practical, behavior-aware feeding solution for smart pet care applications.

II. System Components

A. Microcontroller

The microcontroller chosen for our project is the ESP32-S3, featuring a dual-core 240 MHz processor, allowing for parallel tasks to be handled simultaneously, such as posture detection and motor control. The ESP32-S3 was selected for its wide range of built-in features, most notably UART, I2C, DVP, and PWM, which enable direct communication with components such as the camera and motor driver without requiring additional hardware. Additionally, the ESP32-S3 includes built-in Wi-Fi and Bluetooth for communication with a mobile application.

B. Camera

The camera module used in our project is the OV5640, a 5-megapixel sensor that is able to capture high-quality images in real-time, essential for detecting a dog's posture. It features a DVP interface, which is compatible with the ESP32-S3, allowing direct data transfer without requiring additional hardware [1]. The OV5640 also contains its own image signal processor (ISP), which will handle its exposure and balance without relying on the ESP32-S3 to process. Additionally, the camera provides reliable low light performance for the detection of the dog while under unique light conditions.

C. Dispensing Mechanism

The DRV8825 motor driver is widely used in an automatic dog feeder project to control stepper motors with high precision. In this application, the driver interfaces with a microcontroller like an ESP32-S3 to regulate the rotation of a stepper motor that dispenses food portions. The DRV8825 supports multiple microstepping modes, which improves motion smoothness and positional accuracy, leading to more consistent food dispensing. It also features adjustable current limiting (from 1.5 to 2.2 A/phase with cooling), allowing the motor to operate safely within its rated specifications [2]. Additionally, the driver includes protection mechanisms such as overcurrent protection and thermal shutdown, enhancing system reliability during continuous or unattended operation.

A NEMA-17 bipolar stepper motor (42×38 mm, 1.7 A/phase, 200 steps/rev, holding torque $\approx$37 N·cm) drives a slotted dispensing wheel [9]. The DRV8825 stepper driver supports up to 1/32 microstepping, a supply voltage range of 8.2–45 V, and up to 2.2 A per phase with adequate heatsinking [2]. Step and direction signals from the ESP32-S3 GPIO pins directly control the driver, and onboard current-limiting trimpots allow precise adjustment of motor current without firmware changes.

D. Audio Subsystem

The LM386 audio amplifier is commonly used in an automatic dog feeder project to generate audible feedback signals. In this application, the LM386 amplifies low-power audio signals from the ESP32-S3 microcontroller to drive a small speaker. This allows the system to produce sounds like alerts or feeding cues, which can help train pets to associate audio signals with feeding times [4]. The LM386 operates at low voltage, requires minimal external components, and provides sufficient gain for small speaker output, making it suitable for compact embedded designs [5].

E. Button

The MPB06 push button switch is used in an automatic dog feeder project as a manual override input for emergency operation. In situations where the system loses power or the microcontroller fails, the button allows the user to manually trigger the food dispensing mechanism. When pressed, it can bypass the automated system, ensuring that food can still be dispensed even during faults or low-energy conditions. This adds a layer of reliability and user control to the overall design, making the system more robust and fail-safe.

F. Food Storage Sensor

The RCWL-1601 ultrasonic sensor is utilized in an automatic dog feeder project to monitor the remaining food level inside the storage container. It operates by emitting ultrasonic waves and measuring the time it takes for the echo to return from the surface of the stored food, allowing the system to estimate the distance and, consequently, the fill level [6]. In this application, the sensor is interfaced with the ESP32-S3 microcontroller, which processes the distance data to determine whether the food level is below a predefined threshold. When a low-level condition is detected, the system can trigger an alert to notify the user for refilling. Another benefit to use ultrasonic sensor is non-contact method which is hygienic and not affected by food type or color. This implementation enhances system automation and reliability by enabling real-time monitoring and reducing the risk of running out of food.

G. Depth Module

The TMF8701 ToF (Time-of-Flight) depth module is used in an automatic dog feeder project to detect the presence of a dog and trigger food dispensing. The sensor measures distance by emitting infrared light and calculating the time it takes for the reflected signal to return, enabling accurate proximity detection. The module is positioned to monitor a specific area in front of the feeder. When a dog sits within a predefined distance range, the sensor output is processed by the ESP32-S3 microcontroller to initiate the feeding mechanism. Compared to conventional sensors, the ToF module offers higher accuracy, faster response time, and better immunity to ambient lighting conditions. This improves the reliability of presence detection and ensures that food is dispensed only when the dog is properly positioned.

H. Programmer Chip

The CP2102 USB-to-UART bridge is used in an automatic dog feeder project to enable communication between a computer and the ESP32-S3 microcontroller. It converts USB signals from the computer into UART serial signals that the ESP32 can interpret. The CP2102 is primarily used for programming (flashing firmware) and serial debugging. It allows engineers to upload control algorithms, monitor sensor data (e.g., food level or presence detection), and troubleshoot system behavior in real time [7].

I. Power System

The LM2576-ADJ voltage regulator is used in an automatic dog feeder project to efficiently step down a 12V power supply to regulated 5V and 3.3V levels required by system components. It is a switching (buck) regulator, which offers higher efficiency compared to linear regulators, especially in applications with significant voltage drop [3]. In this design, the LM2576-ADJ can be configured with external resistors to produce adjustable output voltages for powering devices such as the ESP32-S3, sensors, and driver modules. Its ability to handle relatively high current and maintain stable output under varying load conditions makes it suitable for embedded systems with multiple peripherals. Overall, it provides a reliable and energy-efficient power management solution for the feeder system.

III. System Concept

The Sit & Chow system is designed as an intelligent feeding platform that reinforces positive behavior by dispensing food when a dog is detected in a sitting posture. The system integrates embedded hardware, cloud-based machine learning, and a mobile application into a unified architecture.

The embedded subsystem is built around the ESP32-S3 microcontroller, which is responsible for real-time interaction with the physical environment.

The system consists of three main subsystems:

- Embedded subsystem: handles sensing, image capture, and actuation
- Cloud backend: performs machine learning inference and decision making
- Mobile application: enables user interaction, scheduling, and monitoring.

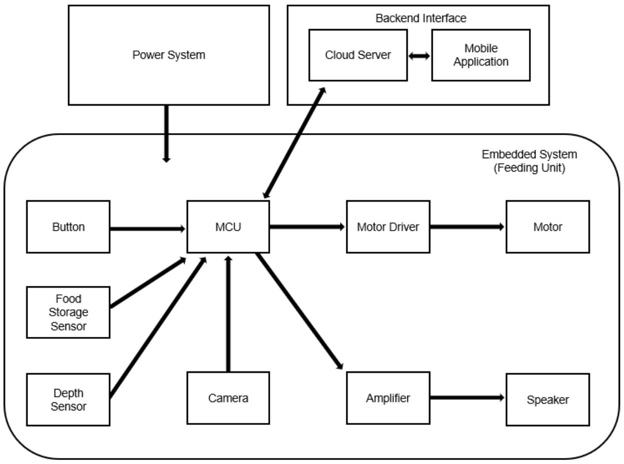

Fig. 1. Block diagram representing the connection of major system components and backend systems.

The primary functions include:

- Capturing image frames using an onboard camera
- Communicating with the cloud server over Wi-Fi
- Controlling the dispensing mechanism
- managing the sensing components (e.g., ToF sensor, ultrasonic sensor, speaker, LEDs)

A. *Feeding Schedule Workflow*

A key feature of Sit & Chow is its ability to dispense food at predefined times while enforcing posture-based conditions. Unlike traditional automatic feeders that dispense food purely based on time, this system introduces a behavioral requirement that the dog must be in a sitting posture before food is dispensed. Feeding schedules are configured through the mobile application and stored in a database (e.g., Firebase).

The scheduling workflow:

1) The user defines feeding times and portion sizes via the mobile application
2) At each scheduled time, the camera is activated, and the dog is checked if it is in range (within 1.5 meters)
3) Once in range, the system initiates the inference pipeline
4) Food is dispensed if the dog is classified as “sitting”
5) The outcome is logged to the backend

If the condition is not met, then the system will:

1) Check if the dog is in range
2) Retry inference
3) After three minutes, the outcome is logged and food is dispensed

This workflow ensures that feeding is both time-controlled and behavior-dependent.

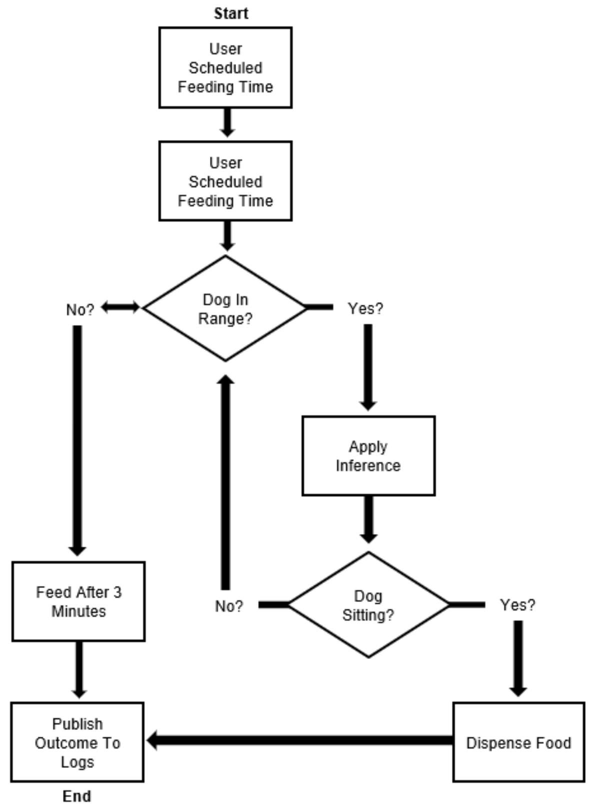

Fig. 2. Diagram of feeding schedule workflow.

Once inference is completed, the cloud server returns a response containing the classification result and associated confidence values. The embedded system then interprets this response to determine whether the feeding condition has been satisfied. If the dog is classified as sitting for a sustained duration, the system actuates the motor to dispense food. Otherwise, the food will automatically dispense after three minutes has passed.

IV. Hardware Details

A. *Power Supply System*

The system operates from a regulated 12 V, 3 A switching wall adapter (Facmogu, 36 W), chosen for its compact size, high efficiency (~85–90%), Level VI compliance, and universal 100–240 VAC input. A CUI PJ-102A barrel jack (5.5/2.1 mm) provides the PCB power entry point [8].

To supply most of the components, the 12V supply is adjusted to 3.3V and 5V based on their voltage range by using the LM2576-ADJ chip.

The use of a switching regulator instead of a linear regulator significantly improves efficiency by reducing power dissipation as heat. This is particularly important when stepping down from 12V to lower voltage levels, such as 5V and 3.3V. In addition, proper decoupling capacitors are placed near power pins to stabilize voltage levels and reduce noise introduced by switching components and motor operation.

The design of the LM2576-ADJ switching regulator, developed by Texas Instruments, was implemented using two separate PCB modules to convert a 12V input supply into regulated 3.3V and 5V outputs [3]. Each module follows a standard buck converter topology, consisting of the LM2576-ADJ IC, an inductor, a Schottky diode, input/output capacitors, and a feedback resistor network [3]. The output voltage is determined by the internal reference voltage (1.23V) and the external resistor divider, as defined by the equation:

$$V_{out} = V_{ref}\left(1 + \frac{R_2}{R_1}\right)$$

Based on this equation and some information in the Texas Instrument datasheet, the resistor values can be chosen to adjust voltage output [3]. Additional external components were chosen based on datasheet recommendations to ensure stable operation. An inductor in the range of 100–150μH was used to support efficient energy transfer, while a Schottky diode provided a low-loss current path during switching intervals. Input and output capacitors were included to reduce voltage ripple and improve transient response. The use of two independent regulator PCB boards enhances system modularity, minimizes noise coupling between voltage rails, and improves overall reliability. This design approach ensures that both the 3.3V and 5V supplies meet the electrical requirements of the automatic dog feeder system.

B. Microcontroller

The PCB for the microcontroller acts as a central control unit for the system, integrating each peripheral and component into a single board. The PCB connects the camera module, motor driver, CP2102 programmer, push button, amplifier, depth module, and ultrasonic sensor all directly to the ESP32-S3. In addition to these components, there is also circuitry for EN and push-button for resets, as well as LEDs to indicate the status of the regulators that we are working with. The hardware design of the microcontroller PCB includes various supporting components to ensure a stable and reliable system operation. Series resistors are used to reduce noise and improve signal integrity on high-speed signals like the OV5640's DVP D0-D7 lines. Decoupling and bulk capacitors are placed close to the power supply lines to stabilize voltage fluctuations when supplied to the ESP32-S3. These capacitors are implemented at the power inputs of the camera, motor driver, depth module, and ultrasonic sensor.

Pull-up resistors are used to enable proper communication on interfaces such as the SCL and SDA lines of the OV5640 camera module. The EN and IO pins incorporate pull-up resistors along with push buttons to provide reset functionality and control of the boot mode for programming. The PCB also includes two LEDs to indicate the status of the 3.3v and 5v regulators. A BS170 MOSFET is used as a switching device to control power to external components. Finally, multiple headers, ranging from 2 to 16 pins, are used to connect and interface the various PCBs to the microcontroller PCB.

C. Dispensing Mechanism

The DRV8825 motor driver PCB holds the primary components needed for safe and reliable control of the stepper motor. It features the DRV8825 driver chip, which receives logic signals from the ESP32-S3 and translates them to control the motor. Two resistors are individually connected to the ISA and ISB pins, which are crucial for current regulation to prevent overcurrent.

A potentiometer forms a voltage divider between the 3.3V input, VREF, and ground, allowing tuning of the motor's current limit by turning the dial [10]. Both bulk and decoupling capacitors are implemented across the board to reduce noise and smooth out voltage fluctuations that occur. In addition, a capacitor is connected between CP1 and CP2 to ensure an efficient current through the motor by storing and transferring charge to increase voltage. Without it, the motor's current would become weak and ultimately lead to malfunction.

To interface from the ESP32-S3 PCB to the DRV8825 PCB, then finally to the NEMA17 motor, an 8-pin and 10-pin header were implemented onto the board. Driven by the ESP32-S3, DIR sets the rotational direction, STEP controls the increment of motion, and EN turns the motor on and off [10]. The SLP pin controls whether the driver is in low power mode, while the RST pin resets the internal logic if it is pulled low. There are also three designated pins that are used to control the resolution, dictating how smooth the motor runs. Power and ground are provided by the designated 3.3V and GND pins[10]. Finally, connecting to the NEMA17 motor header, four pins including A1, A2, B1, and B2 connect to the motor's internal coils which rotate the motor when power is driven through them.

D. Programmer Chip

The CP2102 serves as a communication bridge, converting USB signals from a host computer into UART signals compatible with the ESP32-S3. In the design, the USB connector provides VBUS (5V), D+, D-, and GND connections. The D+ and D- lines are directly connected to the corresponding USB pins of the CP2102, while VBUS is used both as a power input and as a detection signal for USB presence.

Power regulation and stability are ensured by connecting the CP2102 VDD and REGIN pins to the supply rail, with appropriate decoupling capacitors placed close to the power pins [7]. An additional bulk capacitor is used to reduce voltage fluctuations. The RST pin is pulled up to VDD through a resistor to maintain normal operation. Proper grounding is maintained by connecting all GND pins to a common ground plane, which is critical for signal integrity.

For UART communication, the TXD and RXD pins of the CP2102 are cross connected to the RXD and TXD pins of the ESP32-S3, respectively, with 0Ω resistors included for signal protection and noise reduction. Other handshake pins (CTS, RTS, DTR, etc.) are left unconnected as they are not required for basic serial communication [5]. This design ensures reliable USB-to-serial data transfer, allowing firmware uploading and real-time debugging of the ESP32-S3 in the automatic dog feeder system.

E. Amplifier & Speaker

The LM386 audio amplifier is interfaced with the ESP32-S3 and a speaker to provide amplified audio output in the automatic dog feeder system. The LM386 operates as a low-voltage power amplifier capable of driving small speakers directly from microcontroller-level signals.

The amplifier is powered using a 5V supply. The ESP32-S3 GPIO pin is used as the audio input source and is connected to the non-inverting input pin of the LM386. A pull-down resistor is added to stabilize the input signal and prevent floating when no signal is present [5]. Additionally, a small capacitor is connected to ground to filter high-frequency noise, improving signal quality.

The output from the LM386 is AC-coupled to the speaker through a large electrolytic capacitor, which blocks DC components and allows only the audio signal to pass [5]. A Zobel network, consisting of a capacitor in series with a resistor, is connected from the output to ground to stabilize the amplifier and prevent oscillations. This design ensures stable operation and improved audio performance.

V. Software Details

The software architecture of the Sit & Chow system is designed as a distributed, multi-layered framework that integrates embedded control, mobile interaction, and cloud-based intelligence to achieve reliable and intelligent automated feeding. At the embedded level, firmware running on the ESP32-S3 manages real-time sensing, actuation, and communication with peripheral hardware components. The mobile application, developed using Flutter and supported by Firebase services, provides a user-facing interface for scheduling, monitoring, and device management. Completing these layers, the cloud infrastructure hosted on Google Cloud Run performs computationally intensive machine learning inference, enabling accurate dog detection and posture classification. Together, these components operate in a coordinated pipeline, where data flows from the embedded device to the cloud for processing and back to the system for decision-making, ensuring responsive, scalable, and user-friendly operation.

A. Embedded Software

The embedded software for the Sit & Chow system is implemented in C using the ESP-IDF framework on the ESP32-S3 microcontroller. This low-level approach enables deterministic real-time control, efficient memory usage, and direct hardware interaction, which are essential for coordinating sensors, actuators, and communication within a resource-constrained system.

The firmware operates as an event-driven system that transitions between idle, sensing, and actuation states. When a scheduled feeding event occurs, the system first verifies the presence of the dog using a time-of-flight sensor. This prevents unnecessary camera activation and reduces processing overhead. Once the dog is detected within range, the camera is enabled and image frames are transmitted over Wi-Fi to a cloud-based inference pipeline.

Because the ESP32-S3 lacks sufficient computational resources for real-time machine learning, the embedded system acts as a data acquisition and control node rather than performing inference locally. Upon receiving confirmation that the dog has maintained a sitting posture for approximately five seconds, the firmware activates a stepper motor to dispense a precise portion of food. A fail-safe timeout ensures that food is dispensed after a fixed duration even if posture detection fails, maintaining reliability in real-world operation.

The firmware also integrates interrupt-driven routines for responsive operation, allowing concurrent handling of sensor inputs, communication events, and actuator control. Additional features such as speaker feedback and system notifications are coordinated within this layer, making the embedded software the real-time backbone of the system.

B. Mobile Application

The mobile application serves as the primary interface between the user and the Sit & Chow system. Developed using Flutter, it provides cross-platform support with near-native performance and seamless integration with backend services [11]. Firebase is used for authentication, real-time database management, and cloud messaging [12].

The application allows users to create and manage feeding schedules by specifying time, portion size, and active days.

These schedules are stored in the cloud and synchronized in real time with the embedded device. A manual “quick feed” option is also available for immediate food dispensing.

Device provisioning is performed via Bluetooth Low Energy (BLE), enabling users to securely transmit Wi-Fi credentials to the ESP32-S3. Each device is uniquely identified, allowing multiple feeders to be managed within a single account.

The application also supports live video streaming, allowing users to monitor their pet remotely. To optimize performance, the stream is only initiated when requested. Additionally, a two-way audio feature enables users to communicate with their pet through the feeder’s speaker. Notifications such as feeding events and low food levels are delivered in real time, enhancing usability and system awareness.

C. Machine Learning Pipeline

The machine learning algorithm is designed to determine whether a dog is in a sitting posture before dispensing food. The system follows a detection and classification pipeline that:

- localizes the dog within the frame and
- posture classification to determine if the dog is sitting.

During operation, the system processes frames in real time:

- Captures frame from the camera
- Runs YOLO detection
- Extract the ROI from the bounding box
- Classify the posture
- If "sitting" is detected for 5 seconds, trigger the food dispensing.

This design allows for improved accuracy and flexibility compared to a single end-to-end model, particularly in a controlled environment where only one dog is present.

Dog Detection Using YOLO

To detect the presence and location of the dog, a pretrained You Only Look Once (YOLO) model is utilized. YOLO is a real-time object detection model that predicts bounding boxes and class probabilities [13].

A lightweight version (e.g., YOLOv11n) is selected to ensure efficient inference on a CPU-based cloud server. The model is initialized with weights trained on the COCO dataset, which includes a “dog” class [13], thus eliminating the need for extensive fine-tuning.

During inference:

- input frames from the camera are passed to the YOLO model
- the model outputs bounding boxes, confidence scores, and class labels
- only detections that are classified as “dog are retained
- the highest-confidence bounding box is selected (since only one dog is to be expected)

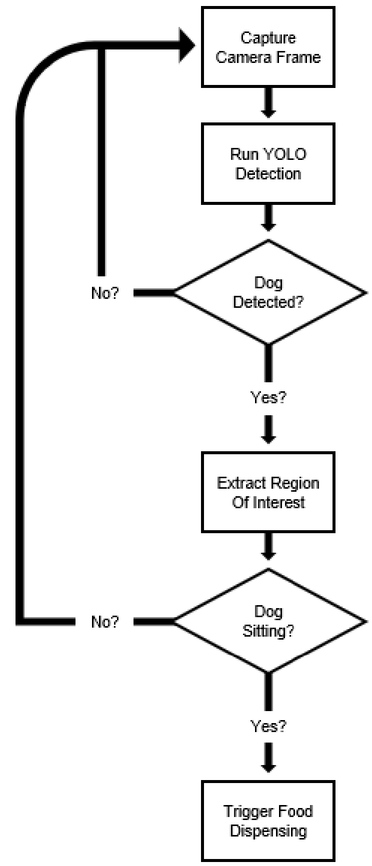

Fig. 3. Workflow diagram representing how the machine learning inference interacts with the system.

After detection, the bounding box is used to extract a Region of Interest (ROI) that corresponds to the dog. To preserve important features:

- A small padding margin is added around the bounding box
- Coordinates are used to determine where the box is located in the image
- The image is cropped and resized to match the classifier’s input dimensions

This step ensures that critical posture features are not truncated from the bounding box, improving performance.

Posture Classification Model

The cropped ROI is passed into a binary image classifier that predicts whether the dog is “sitting” or “not sitting.” A pretrained convolutional neural network (CNN), such as ResNet [14], is utilized where the final fully connected layer is replaced to output a single probability score. The model’s features include:

- Output: sigmoid activation representing the probability of "sitting"
- Loss Function: binary cross-entropy

- Optimization: Adam

Then, the classifier is trained using a dataset that consists of images of dogs that are sitting, standing, lying down, or some ambiguous position. To improve generalization:

- Images are collected from both online datasets (e.g., Stanford Dogs) and camera captures from the overall system
- Data augmentation techniques are applied, such as scaling and brightness adjustments.

Training and Evaluation

The detection model is evaluated using mean Average Precision (mAP@50-95) [13] to measure the localization accuracy.
The classifier performance is evaluated using standard metrics:

- Accuracy
- Precision and recall
- F1-score
- Confusion matrix

Since the overall system will be in a controlled environment (single dog, fixed environment), the classifier is expected to achieve high accuracy (>70%).

Separating the detection and classification into a two-stage pipeline provides several advantages. First, it localizes the dog using object detection, so the classifier operates on a focused region of interest, reducing background noise and improving classification accuracy. This modular approach also allows each model to be independently optimized and updated without affecting the overall system.

D. *Cloud Functionality*

The cloud infrastructure enables advanced features such as machine learning inference, real-time synchronization, and scalable system operation. Google Cloud Run is used to host the machine learning pipeline in a serverless environment, providing an HTTPS endpoint [15] for processing image data and returning results to the ESP32-S3. By offloading computation to the cloud, the system overcomes the hardware limitations of the ESP32-S3 while still achieving near real-time performance.

The inference server is implemented as a containerized Python application that is deployed on Google Cloud Run. This platform enables automatic scaling, low operational overhead, and on-demand resource allocation [16]. The server is built using:

- FastAPI for lightweight REST API endpoints [17]-[18]
- Machine learning libraries, such as PyTorch and Ultralytics YOLO for inference
- Containerization for deployment

The server uses a RESTful endpoint that takes in an image frame as input and outputs a JSON response containing the detection confidence and posture classification ("sit" or "not sit").

Firebase Firestore is used as the central database for storing user data, schedules, and system logs. Its real-time synchronization [19] ensures that updates made in the mobile application are immediately reflected on the device. Cloud-based event handling also supports notifications and logging, allowing users to track feeding outcomes and system behavior.

Overall, the cloud layer extends the system's capabilities by providing scalable computation, centralized data management, and seamless communication between the embedded device and the mobile application.

VI. System Integration

A. *PCB Layout*

Six PCBs were designed: the main MCU board, 2 power supply units, the motor driver board, output sound board, and programming board. The main PCB hosts the ESP32-S3 module, camera header (16-pin DVP with 22Ω series resistors for signal integrity), amplifier header, TMF8701 header, RCWL-1601 header, motor driver header, tactile button, and dual status LEDs (3.3 V and 5 V rail indicators). A solid copper ground plane on both layers minimizes EMI and thermal resistance. Trace widths of 0.508 mm were used for signal lines. Power traces were sized for $\geq$2A continuous current.

The PSU utilizes two LM2679 for 5V and 3.3V regulators, and the PJ-102A barrel jack. The motor driver board contains the DRV8825 with decoupling capacitors (100μF bulk and 100nF ceramic per VMOT rail), a current-sense resistor, and a heatsink pad for thermal management. The output sound uses LM386 to connect ESP32 to speaker. Lastly, programmer board uses CP2102 to program for ESP32. Boards are interconnected via female/male header pairs.

B. *Mechanical Integration*

The OV5640 camera and TMF8701 sensor are mounted at the front center of the feeder enclosure to maximize field of view and proximity sensing coverage. The NEMA-17 motor is positioned on the right side, driving a slotted dispensing wheel that deposits food onto an angled ramp leading to the bowl at center-front. This geometry keeps moving parts away from the dog bowl and below the camera's minimum obstruction height. The RCWL-1601 ultrasonic sensor faces downward into the food hopper from above.

VII. Conclusion

This paper has presented Sit & Chow, a smart dog training feeder that integrates embedded systems, computer vision, and mobile application development into a cohesive, reproducible system. The ESP32-S3 provides a cost-effective platform for real-time peripheral control and wireless communication; the OV5640 and TMF8701 supply the visual and spatial inputs required for reliable posture-gated dispensing; the YOLOv11n + binary CNN pipeline implements the behavioral detection logic; and the Flutter/Firebase mobile stack delivers schedule management, remote monitoring, and live voice interaction.

Individual subsystems have been validated, and full integration testing is underway. If successful, Sit & Chow demonstrates a viable architecture for behavior-reinforced automated pet care and offers a reference design for future smart home appliances that couple machine learning with embedded hardware control.

Acknowledgement

The authors wish to thank Dr. Saleem Sahawneh (mentor), Dr. Tong Wu, Dr. Shahana Ibrahim, and Dr. Truong Nghiem (review committee), Dr. Arthur Weeks (lab professor) of the UCF Department of Electrical and Computer Engineering for their guidance throughout this senior design project.

References

[1] OmniVision Technologies, "OV5640 5MP CMOS image sensor datasheet," *SparkFun Electronics*, [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/LightImaging/OV5640_datasheet.pdf

[2] Texas Instruments, "DRV8825 stepper motor controller datasheet," *Texas Instruments*, 2014. [Online]. Available: https://www.ti.com/lit/ds/symlink/drv8825.pdf

[3] Texas Instruments, "LM2576HV SIMPLE SWITCHER voltage regulator datasheet," *Texas Instruments*, 2016. [Online]. Available: https://www.ti.com/lit/ds/symlink/lm2576hv.pdf

[4] Espressif Systems, "ESP32-S3 datasheet," *Espressif Systems*, [Online]. Available: https://documentation.espressif.com/esp32-s3_datasheet_en.pdf

[5] Texas Instruments, "LM386 low voltage audio power amplifier datasheet," *Texas Instruments*, 2004. [Online]. Available: https://www.ti.com/lit/ds/symlink/lm386.pdf

[6] Adafruit Industries, "Ultrasonic distance sensor product page," *Adafruit*, [Online]. Available: https://www.adafruit.com/product/4007

[7] Silicon Laboratories Inc., "CP2102/9 USB-to-UART bridge datasheet," *Silicon Labs*, 2020. [Online]. Available: https://www.silabs.com/documents/public/data-sheets/CP2102-9.pdf

[8] CUI Devices, "PJ-102A power jack datasheet," *Digi-Key Electronics*, [Online]. Available: https://www.digikey.com/en/products/detail/same-sky-formerly-cui-devices/PJ-102A/275425

[9] Pololu Corporation, "DRV8825 stepper motor driver carrier," *Pololu*, [Online]. Available: https://www.pololu.com/product/2267

[10] Pololu Corporation, "DRV8825 stepper motor driver carrier," *Pololu*, [Online]. Available: https://www.pololu.com/product/2133

[11] Google, "Flutter documentation," *Flutter*, [Online]. Available: https://docs.flutter.dev/

[12] Google, "Firebase documentation," *Firebase*, [Online]. Available: https://firebase.google.com/docs

[13] Ultralytics, "Ultralytics YOLO documentation," *Ultralytics*, [Online]. Available: https://docs.ultralytics.com/

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[15] Mozilla, "HTTP overview," *MDN Web Docs*, [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview

[16] Google, "Cloud Run documentation," *Google Cloud*, [Online]. Available: https://cloud.google.com/run/docs

[17] S. Ramírez, "FastAPI documentation," *FastAPI*, [Online]. Available: https://fastapi.tiangolo.com/

[18] L. Gupta, "REST API tutorial," *RESTfulAPI.net*, [Online]. Available: https://restfulapi.net/

[19] Google, "Cloud Firestore documentation," *Firebase*, [Online]. Available: https://firebase.google.com/docs/firestore