ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Automated Video Capture Bird Feeder with Data Collection

Team #10

KEVIN LI (kli56)
JOHN GOLDEN (jgolden4)
COLTEN BRUNNER (cbrunner)

TA: Nikhil Arora

May 1, 2024

Abstract

This document details the design of an automated bird feeder with bird species identification and data collection capabilities. Nature enthusiasts enjoy watching birds using feeders, and this project aims to enhance that experience by providing remote viewing and data analysis.

Contents

1 Introduction

1.1 Problem

Many nature enthusiasts enjoy watching birds outside their windows with homemade or store-bought feeders. This practice has been going on for many years, but until recently it has been impossible to see the birds feeding without being present. With modern-day technology, it has become possible to mount cameras onto or adjacent to bird feeders to see birds feeding, but in the new era of information technology, there should be more to bird feeders than simple footage. We seek to add onto an automated video capture system by including data capture to analyze when peak feeding hours occur. This problem occurs for common bird watchers and ornithologists alike. Whether it is knowing when to sit in front of your bird feeder or wanting to collect feeding data in specific areas, this is a problem that necessitates a solution.

1.2 Solution

The solution we propose involves a bird feeder that has a camera to turn on when motion is detected. The idea is to have an passive infrared sensor that would trigger a camera to record for a given set of times if motion is detected. In addition, specific data points that would benefit nature enthusiasts would be acquired and stored. These would include time intervals when birds arrive to identify peak bird times. We also want the end user to be able to view live footage of the bird feeder via a website URL. Our solution would implement all of this by using a power pack located on the bird feeder that supplies power to the camera, motion detector, and microcontroller that all work in tandem to create our final product.

1.3 High Level Requirements

- Activate the camera within 10 seconds of a bird landing, correctly identifying bird presence with 90 percent accuracy.
- Deliver a live video feed to the user device with a resolution of 720p or higher, while displaying the feed with a delay of no more than 15 seconds.
- Turn off the camera within 30 seconds after the bird departs. The camera should stay on if motion is detected within the birdhouse during the timeout period, while minimizing false positives triggered by wind, leaves, or other non-bird movements.

1.4 Subsystem Overview

The subsystems are separate systems that will work in tandem to ensure the high level design requirements are met. Subsystems in this project are divided into hardware and software categories. Namely, the video capture, data collection and real-time video feed are mainly software while the power, sensing, and microcontroller are hardware oriented. On the software side, the video capture has an input from the microcontroller to turn on the camera when the infrared sensor outputs detection. The detection will feed into one of the Raspberry Pi GPIO pins which triggers the video live stream to start recording. This live stream is displayed on a web server hosted on the Pi, which also displays a graph of bird activity across time. The hardware focused subsystems are the power system and sensing system. The bird feeder housing holds the raspberry pi as well as the power, microcontroller, and PIR sensor. Like stated previously the PIR sensor will be connected to the microcontroller and the raspberry pi via GPIO pins. The power system will be connected mainly to the microcontroller circuit via the pcb as the raspberry pi will be on a separate wall to avoid protect it from damage.

1.5 Block Diagram

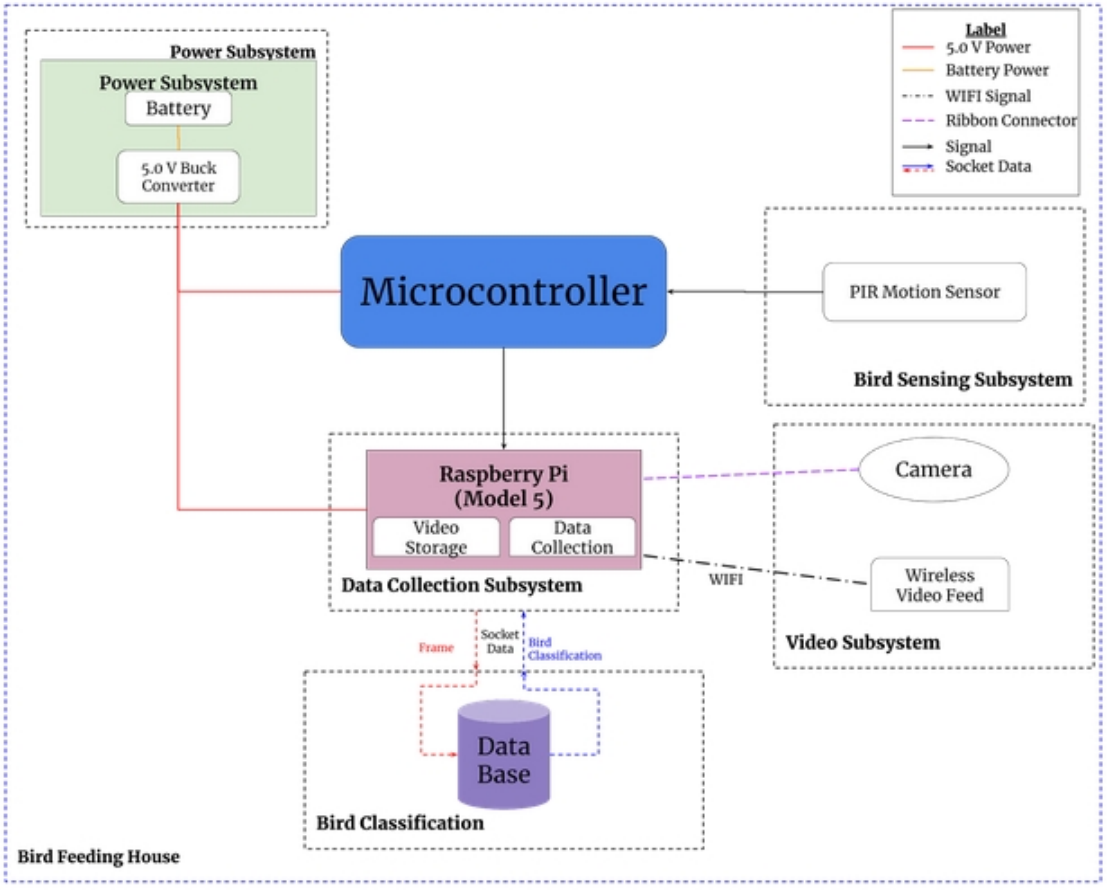

Figure 1: Block Diagram

2 Design

2.1 Design Procedure

The block diagram for our project went through several main changes throughout the semester. Significant changes were made to the power, sensing, classification and data collection subsystems.

Initially, Our project used an ESP32 microcontroller to handle the GPIO from a sensor interfaced with the Raspberry Pi, as well as the power rails. We used the ESP32 because it was an easily accessible Dev Kit in the lab and was very well documented. After discussion in our design review we decided to switch to an ATMega328 for a couple of reasons. The first reason for the switch was to power the ATMega with 5V instead of the 3.3 required for the ESP32. This greatly simplified the power system. The second reason for the switch was for soldering considerations. The ESP32 had more pins in less area than the ATMega, and also required greater auxiliary components which crowded the board and over complicated the soldering process. Additionally, we had already had experience with the ATTiny from the soldering assignment, so using the ATMega offered a simplicity that the ESP32 did not.

In the first iteration of our project, we used an HC-SR04 ultrasonic sensor to determine object proximity to the sensor. This choice was due to having past project experience with this sensor, and its known simplicity. During our design review Professor Gruev pointed out that birds may be sensitive to ultrasonic frequencies and be deterred from our bird feeder. To avoid this unforeseen complication, we decided to switch to the HC-SR501 Passive Infrared Sensor. The switch to this sensor solved two problems for our group. Firstly, the potential for scaring away birds was removed, and more importantly the nature of the PIR sensor allowed us to prevent incidental camera triggers. With the ultrasonic sensor, any motion would be detected, even if it was just a leaf passing in the wind, but with the PIR sensor only living creatures would activate the control signal because of the body's infrared radiation being emitted.

The final significant change for our project was the power system. Initially we had debated between using wall power and using battery packs without ever fully confirming our decision. Either option involved us using voltage regulators to step down voltage to usable levels. After our design review, we used the advice of the course staff to switch to using a battery pack which takes advantage of Buck Converters to manage our voltage. These allow for much better heat regulation as well as control when stepping down from higher voltages. This also greatly reduced the current draw from voltage regulators that would have strained our battery pack life. One significant note for our power system is that the Raspberry Pi is powered separately using wall power for two main reasons. Firstly, we never planned for the Pi to be a permanent component as it was independently financed, and we wanted to avoid permanent integration. Secondly, the Pi has the largest current draw and would require a much larger battery back to sustain a respectable battery life of at least one day.

2.2 Design Details

In this section, we will provide a detailed overview of the key subsystems for our project. We will examine the design principles, justifications, and technical requirements, and verification of each component, highlighting their roles in facilitating the operation of the overall system.

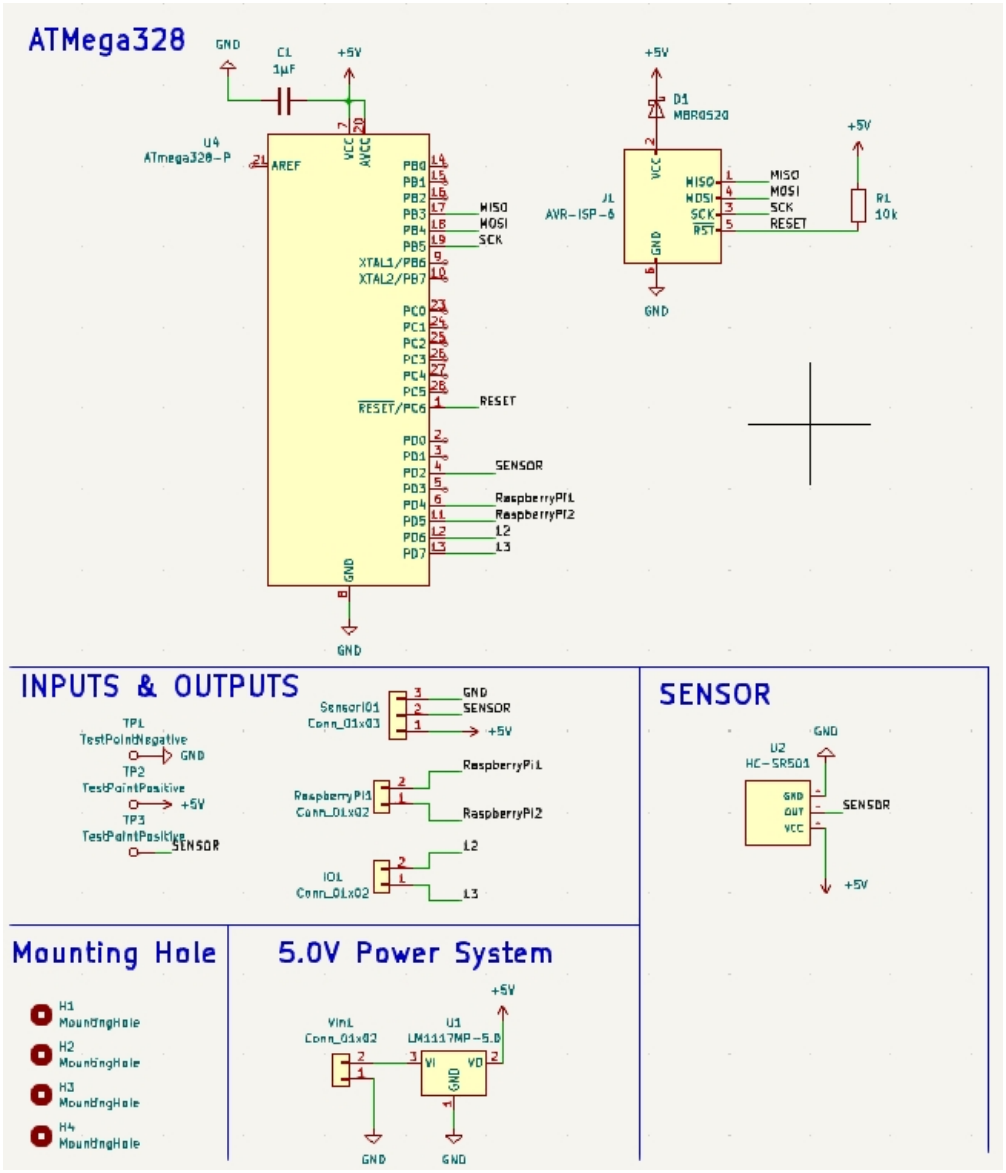

Figure 2: Full PCB Schematic

2.2.1 Video Capture

The subsystem, composed of a Raspberry Pi, camera, camera ribbon, and SSD, is responsible for initiating video recording upon receiving a signal from the microcontroller and live streaming the video to the dedicated web server. [1] [2] We have focused on three primary aspects: timeliness of recording start, video data processing rate, and video quality.

This subsystem was the key component in our product and so we designed it with the intent to function separately from the microcontroller. In the following, the justification and quantitative results will be discussed for the Video Capture subsystem.

Having a manual input of a key press we were able to verify that the startup time was within 1-2 seconds via a timer. The average startup time ignoring the web server refresh rate on the client side, was 1.1 seconds on average with a standard deviation of .4 most likely contributed to human error of using a timer. Throughout all tests, we had the activation time well below the 10-second limit. In regards to the video data processing rate, this was verified through the lib-camera library native on the Raspberry Pi as well as sampling small videos. The camera was recording in 1536x864[3] at 30 frames per second which is well in the range of HD resolution and much above the SD resolution. With this fact, we made several small video recordings of 10 seconds with the average being around 105MB given that the .mp4 extension has 264 colors. The different lighting had no effect on video size and the processing rate was close to 10MB if 30 frames per second is selected and before any video compression is done. The video quality was ensured through the same process when verifying the video data processing.

The requirements and verification methods can be found in the appendix C

2.2.2 Data Collection

The data collection subsystem focuses on collecting and storing data. The system is primarily to be tested via the software we are writing for this subsystem as the hardware systems should still be tested but involve the Raspberry Pi functioning correctly in tandem with the MicroSD associated with this device. The software involves creating CSV files on the Raspberry Pi at regular intervals, ensuring they arrive on time, containing the correct information in the proper format, and ultimately if they are securely saved on the SSD. Additionally, due to further complexity requirements, we will use a PyTorch model designed by MIT. [4]

Again like the Video Capture subsystem, this was built modularly in a separate Python file and later merged into one main file once properly verified. The subsystem was tested before any implementation with the microcontroller or any other subsystems. The justification and quantitative results will now be discussed for the Data Collection subsystem.

The CSV file generation had exact timestamps of when the camera was activated with an encoded 5-second cool down to not record the same time-stamp on the same activation event. This meant every recorded timestamp was well within 1 second between the expected and actual creation times. This meant the distributions were trivial as the discrepancies were only due to small deviations in the code runtime. An analysis of the CSV file showed a column of all timestamps recorded with the name of the file being unique but however, due to the lack of a long-term field test, we kept the timestamps all in one csv. A quick edit can be made to store a CSV for each day but due to the aforementioned lack of a need, we omitted this for the demo. In regards to model accuracy, we first selected an image of the Northern Mockingbird among others and ran them through the selected model to get a baseline accuracy. We then printed these images and presented them in front of the camera which then parsed the image and sent it to the model for classification. All clear images that were in front of the camera with no background noise due to the image not covering the camera were within 10 percent between the baseline and our experiment. A worthy note is we excluded any birds that were unable to be classified by the baseline or were consistently misidentified in the experiment due to the obvious gap of using a printed image instead of a real bird. We also tested during the demo tilting the images which resulted in a complete inability to classify correctly which was from the model being trained on non-tilted images of birds facing the camera.

Visuals on this can be found in appendix A.

The requirements and verification methods can be found in the appendix D

2.2.3 Power System

The power system ensures consistent voltage delivery for the Raspberry Pi and the microcontroller, guaranteeing smooth operation for both components. The Raspberry Pi requires 5.0 volts [5] and the microcontroller requires 5.0 volts. By measuring and monitoring the voltage during various operating conditions, we can verify its stability and avoid potential power-related issues. In the initial development stages, this subsystem takes power from a wall outlet and for the final product it will be using a battery. Below is the specific requirements and verification methods for each key aspect: [4]

The Power Subsystem was also isolated from the rest of the subsystems when were performing voltage testing. The following are the justification and quantitative explanations for the aforementioned subsystem.

The Raspberry Pi verification became trivial [5] as we were allowed to use the wall power for the Raspberry Pi due to the small computer being personal property and extremely sensitive to voltage changes could directly damage an expensive component. Nonetheless, the DC voltage supplied from the wall bank never sent off any low voltage alarms which begin when the voltage threshold is below 4.75 volts likewise for high voltage alarms when it rises above 5.25 volts. The power supply is rated at supplying 5.1 volts at 5 amps. The ATmega328p when tested with various loads using combinations of resistors to simulate loads found that the voltage regulator was able to keep between 4.67 to 5.44 volts.

The requirements and verification methods can be found in the appendix E

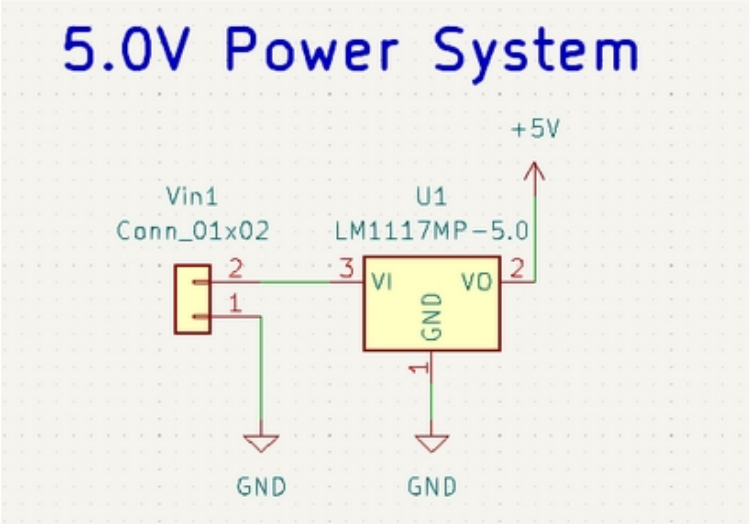

Figure 3: Power System Schematic

2.2.4 Real-time Video Feed

This system uses Raspberry Pi as a broadcasting server. Either via a web server or a direct connection a user should be able to see video streaming.

The real-time video feed again was isolated from all possible subsystems but however the video capture subsystem and parts of the data collection subsystem had not been interconnected in order to properly test this subsystem. The following are the quantitative explanations and justifications for this subsystem.

The stream delay was measured through various activation events and then the connection of a second device to the webserver to check the delay of the stream. We found the stream to be a complete live feed and not differentiable via the human eye as well as the video stream always displayed within the 5-second refresh timer on the client side when connecting. The video quality depended on the stream device's display resolution however we found it was able to deliver up to its camera resolution of 1536x864 at 30 frames per second. Ignoring any problems with the network between the stream and the client device we found it to never drop below standard resolution (SD). The bird name display was always displayed once the server responded to a given frame. The classification model always attempts to classify a frame whether it be extremely inaccurate or accurate and will return the predicted bird as a string as well as the accuracy and elapsed time as a float. Typically it takes just more than 10 seconds to send the frame, receive the frame, and display classification results as it incurs some network delay and the classification time on average takes between 5-6 seconds but can take up to 10 seconds.

The requirements and verification methods can be found in the appendix F

2.2.5 Bird Feeder

The system is enclosed in a waterproof container with two wires for the camera and ultrasonic sensor exiting the container. The wires protruding from the enclosure are waterproofed with silicone to prevent water going in. The camera is placed inside the bird feeder, facing the feed, while the ultrasonic sensor is positioned nearby. When birds land, the sensor detects motion, activating the camera for recording.

This subsystem was tested last as it required the completion of all subsystems and the complete integration to be completely tested as it depended on all subsystems incorporated in the entire bird feeder. The following are the quantitative explanations and justifications for the Bird Feeder subsystem.

To test that the PIR sensor only picked up living objects we moved out of the sensing angle and used a large meter stick to wave in front of the bird feeder. We found that it never activated the camera and therefore the stream if nobody is in the viewing angle which is close to 7 meters in a conical shape from the sensor. To prevent water damage to any components we sealed all electrical components in a water-tight poly case underneath the feeder. We ensured any holes made were sealed via electrical tape as well as any connections. We also ensured the waterproof of the case by removing all electrical components and splashing water unto it and checking for any leaks. We additionally mounted the PIR sensor within the shelter of the bird feeder which protected it from any weather that would not also damage the bird feeder structure itself.

The requirements and verification methods can be found in the appendix G

2.2.6 Sensing

We decided to separate the sensor into a separate subsystem. This subsystem uses the HC-SR501 PIR motion detector which uses infrared to detect if a living creature is in front of the sensor. This subsystem directly integrates with the microcontroller using Arduino code to correctly output a signal to the raspberry pi. [3].

The Sensing subsystem was able to be tested completely separate from all other subsystems via a breadboard. The sensor we found was continually able to pick up motion via the voltmeter measuring a high output when a hand was waved in front as well as visual cue from a red LED connected to the sensing output. We found the output voltage to then immediately drop from anywhere from 3.17 to 3.33 volts to zero when no motion is detected. The LED also would turn off immediately after the hand was removed. The implementation with the microcontroller and sensor was aided by this source [6].

The requirements and verification methods can be found in the appendix H

2.3 Tolerance Analysis

One key subsystem in our project is the power for the entire project. In our power subsystem we want to ensure that the bird feeder can operate for a reasonable amount of time without a charger or battery replacement. To determine the possible lifetime from the battery pack we are using, we need to first find the total current draw from the battery pack and then compare it to the datasheet rating. Below is a table with the necessary currents from different components.

$$T = I_{total} * Rating = 2500\,mAh\,(20\,mA + 10\,mA + 15\,mA) = 55.56\,h \tag{1}$$

Using these values in the equation above, we can simply multiply our total current by our batteries' 2500 mAh rating to get the predicted time the power would last well over two days. This time is more than enough for our intended use to examine birds over a weekend with minimal intervention as food would likely run out before battery life.

component	**ATMega328p**	**PIR Sensor**	**LM117**
current draw	20 mA	10 mA	15 mA
voltage	3.3 V	5.0 V	3.3-5.0 V

Table 1: Component Voltage and Current Values

One more subsystem that required analysis was the passive infrared sensor. The concern for this system was that the sensor needs to capture the full area in the bird feeder so that no incident birds are missed by this system. For this calculation, the physical dimension of the feeder is needed as well as the placement area of the sensor. Shown in the figure below is the approximate geometry of our bird feeder. The PIR is rated for 120°conical range, so to ensure there are no blind spots in our viewing area, some geometrical calculations are necessary. Every angle needs to be less than 60°on either side of the PIR's orientation to fully capture the feeding area. Shown below are the 3D, top, and side views of the bird feeder. The PIR location is represented by the light blue box, the PIR orientation is in yellow, and the green line is used as an orientation reference.

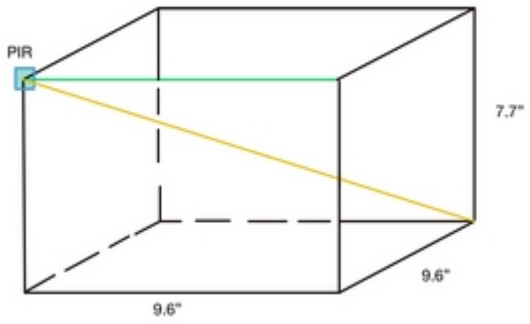

Figure 4: 3D View

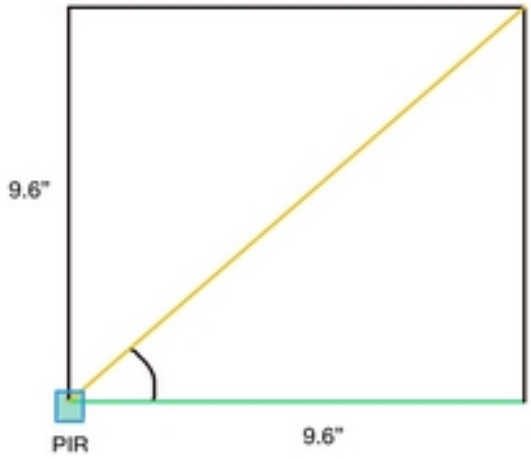

Figure 5: Top View

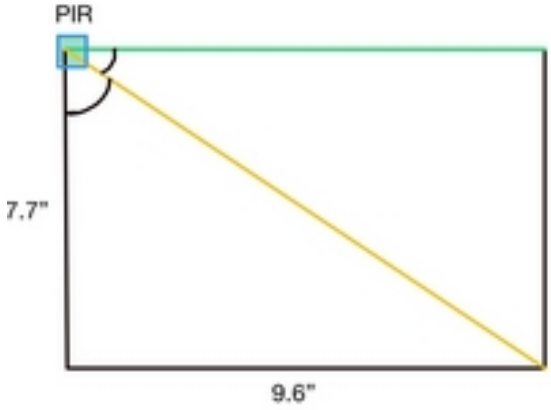

Figure 6: Side View

Based on these drawings, simple geometry can be used to solve for the angles on each side of the line of sight. In the top view, the angle on each side of the yellow line can be calculated by using:

$$\Theta_1 = \Theta_2 = \arctan\left(\frac{9.6}{9.6}\right) = 45^\circ \tag{2}$$

For the side view, the larger angle is found by using:

$$\Theta_3 = \arctan\left(\frac{9.6}{7.7}\right) = 51.3^\circ \tag{3}$$

The smaller angle is found by using:

$$\Theta_4 = \arctan\left(\frac{7.7}{9.6}\right) = 38.7^\circ \tag{4}$$

Based on these calculations, all four angles are well less than the 60°limit we found from the data sheet and there will be no geometric restrictions for the PIR sensor.

3 Costs and Schedule

This section discusses the cost and scheduling used throughout the entirety of our project. Working within the provided funding of the course and keeping in line with a strict schedule were fundamental to our groups success. The following serves as a comprehensive overview to our resource management and scheduling.

3.1 Costs

Part	**Cost**	**Part**	**Cost**
ATMega328P	$2.42	Raspberry Pi Camera	$25.00
PIR Sensor	$3.95	Bird Feeder	$25.99
Raspberry Pi Model 5	$169.95	Raspberry Pi Cooler	$5.00
Voltage Regulator	$1.09	3x2 Header	$1.75
Polycase Enclosure	$27.10	(2) 10 kΩ Resistors	$0.20
(3) 1 μF Capacitors	$0.33	(3) 1x2 Headers	$0.36

Table 2: Project Component Pricing

The total cost of the project can be calculated by adding up the costs from Table 2 with the total labor cost calculated in this equation 5:

$$P_{labor} = (3_{teammates})(16_{weeks})(\frac{12\frac{hrs}{week}}{teammate})(40\frac{\$}{hr}) = \$19,200.00 \tag{5}$$

$$P_{parts} = \$2.42+\$25.00+\$27.10+\$3.95+\$25.99+\$0.89+\$169.95+\$5.00+\$1.09+\$1.75 = \$263.14 \tag{6}$$

$$P_{total} = P_{labor} + P_{parts} = \$19,463.14 \tag{7}$$

3.2 Schedule

The project schedule (see Appendix B) proved to be a valuable guide throughout the development process. We were able to adhere to the timeline for most tasks, ensuring a steady and focused workflow. However, we encountered an unforeseen delay in receiving the second iteration of the PCB. This setback could have significantly impacted our progress. Fortunately, our commitment to ongoing testing on the breadboard mitigated this challenge. By continuing with these tests as planned, we were able to maintain momentum and achieve a functional product on the breadboard despite the delay in receiving the final PCB.

4 Conclusions

Overall, our group found this semester to be a successful venture in the technical portion of our project, as well as personal development. Even though we did not achieve every single one of our goals, we were able to present a working product that satisfied all of our high level requirements and even accomplished more. All of our group members learned the importance of teamwork, communication, and planning ahead, and we will take these skills with us in our future careers as engineers.

4.1 Accomplishments

The most important accomplishment we achieved was succeeding in our high level design. Our camera was able to communicate with the motion sensor to accurately identify bird presence without false positives, deliver a high resolution live feed to our web server, and also turn off streaming when birds departed the feeder. Beyond this, our group successfully implemented a bird recognition software program which could take a frame from our live video stream, send it to a laptop, and make a "guess" of what species the bird is using machine learning software. Although this was not field tested, when printed images of birds were used, the model accurately determined the species of many test images.

4.2 Failures

The main failure of our project was our inability to program our ATMega328p using the ISP. Our latest PCB revision did not deliver enough voltage to the chip in order for it to be programmed. We only were able to deliver 4.38 V instead of the necessary 5.0 V. On our next revision we were planning on changing voltage regulation as well as including decoupling capacitors paired with a 16 MHz oscillator to make programming more easily accomplished. This oscillator would have allowed us to not rely on the 8 MHz internal oscillator expediting the bootloading process.

One other shortcoming of our project was the lack of flexibility for our bird recognition module. Our design was able to successfully detect pictures of bird when the resolution was clear and the picture was oriented correctly. When the image was oriented in an awkward direction, the model's accuracy dropped significantly. When we tested with an image of a Northern Mockingbird, the accuracy dropped from 99.9% with the correct species listed down to the incorrect species at 27.4% accuracy.

Overall, although these two shortcomings were unfortunate, we believe with more time and resources we could successfully overcome or mitigate them.

4.3 Ethics and Safety

With a project such as ours, ethics and safety are of the utmost importance every step of the way. Addressing the ethical concerns there are two main concerns we have identified. First, we are concerned that this project has similar products that are already commercially available, which means that we need to actively ensure that we are coming up with unique ideas that are firmly our own. The second ethical concern comes from the fact that there have been other past senior design projects that have similar subsystems to ours, and we want our project to be unquestionably unique. These two problems have similar solutions, and we have identified several ways to ensure that our project is ethically compliant with the IEEE code of Ethics, mainly to comply with the tenet of "To uphold the highest standards of integrity, responsible behavior, and ethical conduct in professional activities." [7].

The physical birdhouse comes with two main safety concerns for our project. The first is water protection for our power system and electronics against inclement weather. This concern is addressed by our water proofing of our enclosure system. The second outdoors related concern is for squirrels and other unintended wildlife from harming the system. This problem is addressed by isolating our bird feeder and protecting all electric components from the environment with adequate security to prevent intrusions.

As a low voltage project, the safety concerns with this project are the same of most typical hardware design labs and we will observe all of the necessary precautions. The main concerns we have for safety in this project is following lab procedures during fabrication and testing. On the user end, the components will be contained in the bird feeder enclosure which means that they should be properly isolated from the electronics and power supply.

The main safeguard we have against copying other products that are on market is to be active in our competitor research rather than being passive. This means we will be actively scouring the internet and other marketplaces for similar products and ensuring that the project we are completing is fully our own unique idea. We will address this during the design process by using our design notebooks to track our own research and creative processes. Tracking these dated and detailed entries will give definitive proof that we actively sought to ensure the individuality of our design process.

Overall, our project's safety and ethical concerns can be minimized and addressed through the use of active prevention techniques and engineering controls to ensure a safe, ethical project for both the fabrication team and the end user.

4.4 Next Steps

The first steps we would take in continuing this project would be fixing the failures discussed above. After fixing the PCB and improving the recognition algorithm, we would want to add a cellular notification system for the user. This would come in the form of an app that notifies the user whenever the PIR sensor is pinged, and would also display the live feed of the bird feeder. In addition to this, we would also add in frame optimization, meaning we would implement a script to identify the optimal frame for our bird detection algorithm and only send that to be classified. Ideal frames would be oriented with the bird facing the camera and oriented right-side up. The last change we would make is including a more dedicated video processing unit than the Pi. We would want to research better options to ensure that our project is the most efficient on the market. Overall, we are confident that these next steps would improve our project greatly, and open the door for even more innovation for our team to improve upon.

References

[1] Instructables. ""How to Make Raspberry Pi Webcam Server and Stream Live Video —— Motion + Webcam + Raspberry Pi"." (2018), [Online]. Available: https://www.instructables.com/How-to-Make-Raspberry-Pi-Webcam-Server-and-Stream-/ (visited on 05/01/2024).

[2] R. N. Tutorials. ""Video Streaming with Raspberry Pi Camera"." (2018), [Online]. Available: https://randomnerdtutorials.com/video-streaming-with-raspberry-pi-camera/ (visited on 05/01/2024).

[3] ArduCam. ""Arducam 5MP OV5647 1080p Mini Camera Module for Raspberry Pi 5/4/3B+/3"." (2021), [Online]. Available: https://www.arducam.com/product/arducam-ov5647-standard-raspberry-pi-camera-b0033/ (visited on 05/01/2024).

[4] Moddy2024. ""Bird-Classification"." (2023), [Online]. Available: https://github.com/Moddy2024/Bird-Classification?tab=MIT-1-ov-file (visited on 05/01/2024).

[5] HDMI. ""Raspberry Pi 5 Datasheet"." (2023), [Online]. Available: https://datasheets.raspberrypi.com/rpi5/raspberry-pi-5-product-brief.pdf (visited on 05/01/2024).

[6] ElectronicWings. ""PIR sensor interfacing with ESP32"." (2020), [Online]. Available: https://www.electronicwings.com/esp32/pir-sensor-interfacing-with-esp32 (visited on 05/01/2024).

[7] IEEE. ""IEEE Code of Ethics"." (1990), [Online]. Available: https://ieeexplore.ieee.org/document/65865 (visited on 05/01/2024).

Appendix A Data Collection

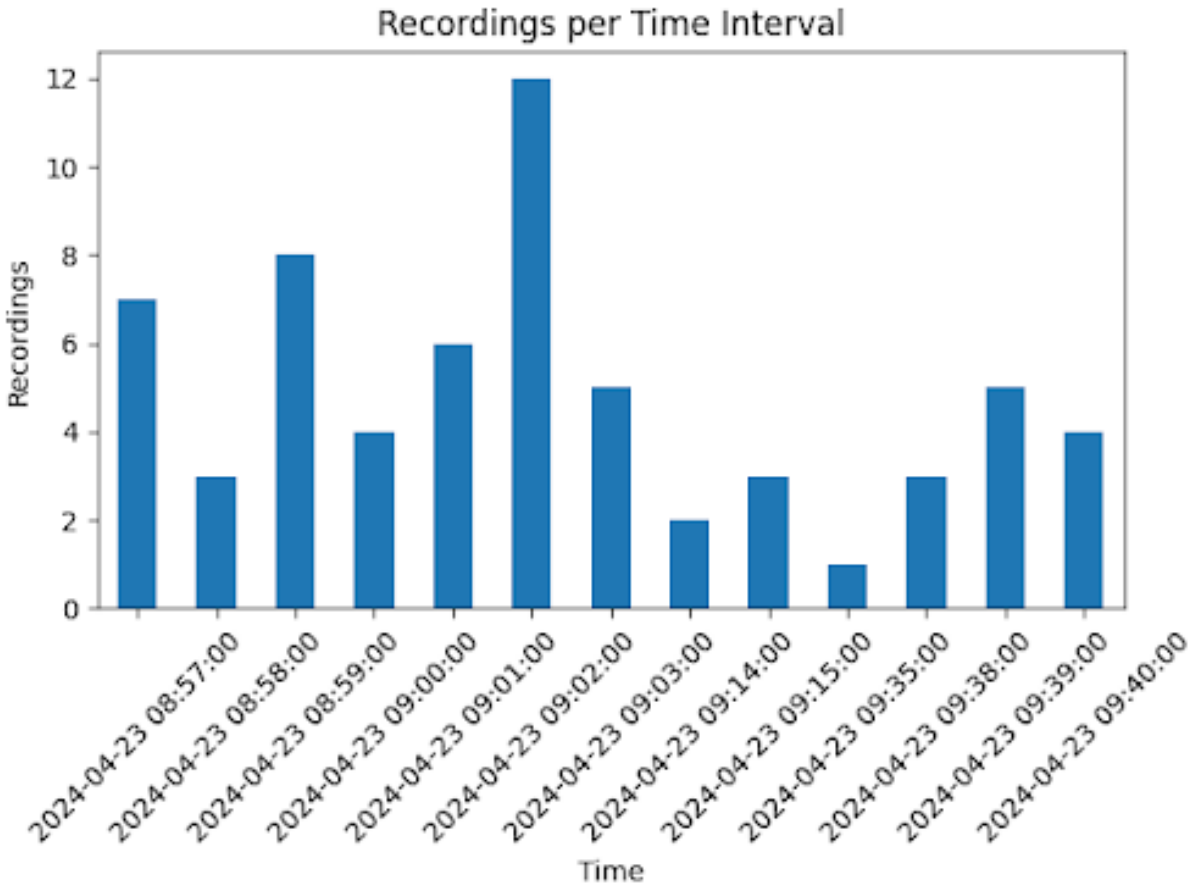

Figure 7: CSV example display

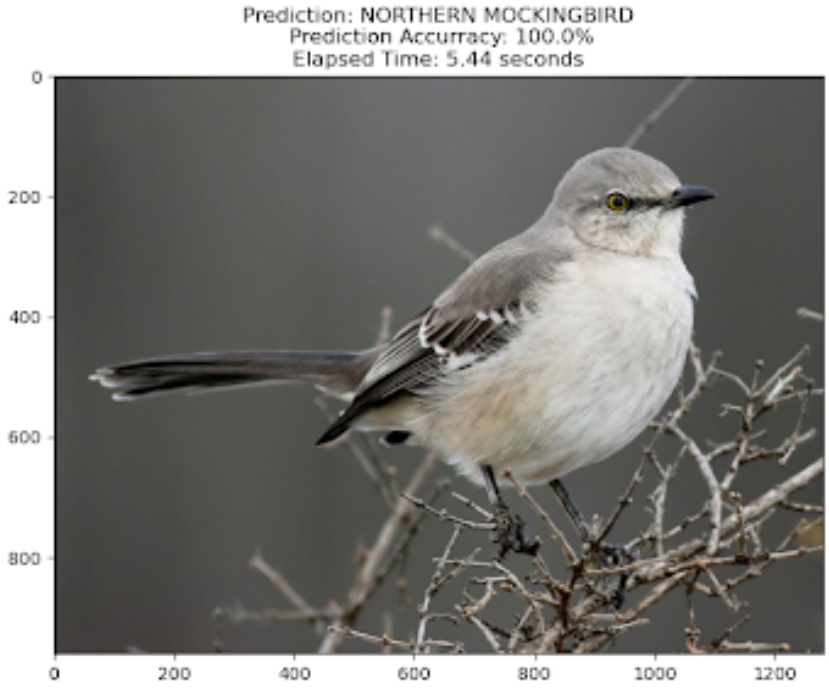

Figure 8: Baseline

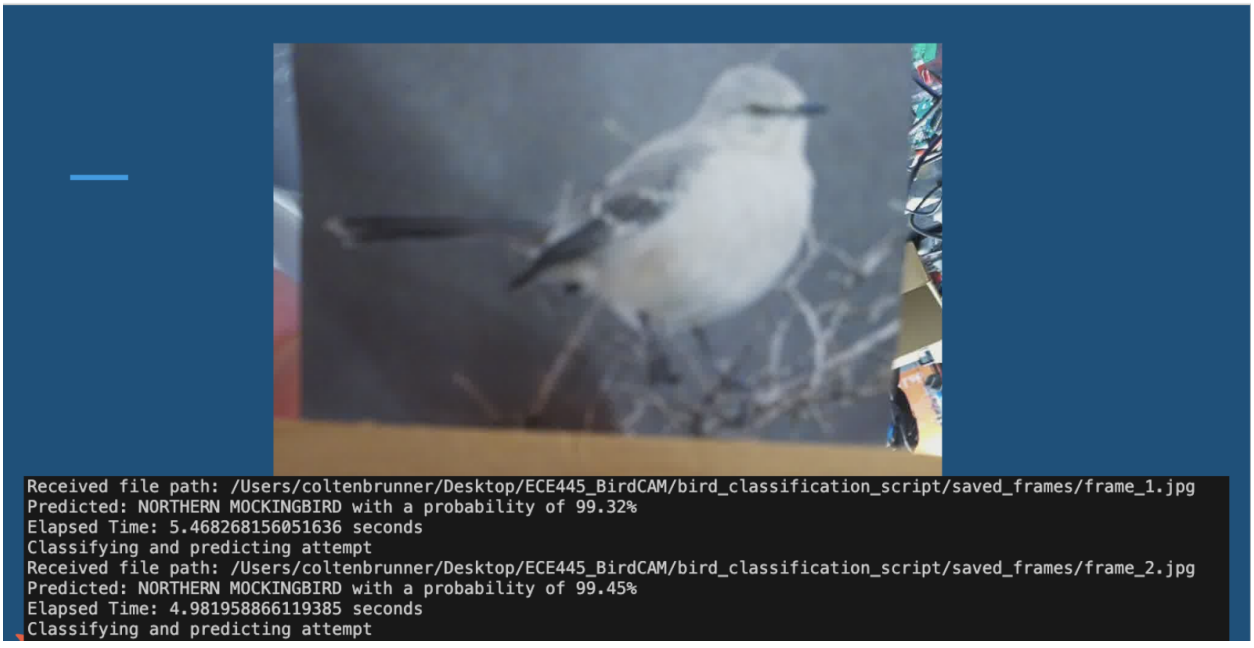

Figure 9: Experiment

Appendix B Schedule Table

Week	**Tasks**	**Team Member**
Week of 2/19	Finalize Project Proposal	Everyone
	Write Design Document	Everyone
	Breadboard test ESP32 and ultrasonic transducer	Kevin John
	Order Raspberry Pi & Raspberry Camera	Colten
Week of 2/26	Design Review with Instructor and TAs	Everyone
	PCB Review	Everyone
	Unit test voltage regulators/design power subsystem	John
	Finalize PCB design for submission by Friday	Kevin
Week of 3/4	Start on Camera setup and implementation with Pi	Colten Kevin
	Order PCB/Teamwork Evaluation	Everyone
	Test power subsystem on breadboard	John
	Make sure PCB has been ordered	Kevin
Week of 3/11 (Spring Break)	Start on data collection software on raspberry pi side	Colten Kevin
	Get the waterproof container from machine shop	Kevin
	Buy Bird Feeder	Kevin
	Interface power subsystem with sensing subsystem and microcontroller, everything should be assembled except for the pi camera (½)	John
	Test PCB to see if it works	Kevin
Week of 3/18	Start on web server on Raspberry Pi and if time allows implement data on web server as well as video feed	Colten Kevin

Week	Tasks	Team Member
	Prepare presentation outline and visuals	Everyone
	Interface power subsystem with sensing subsystem and microcontroller, everything should be assembled except for the pi camera (2/2)	John
	Work and debug PCB schematic for next order	Kevin
	Order new PCB if needed	Kevin
Week of 3/25	Test and debug any problems with the Pi system	Colten Kevin
	Test PCB to see if it works	Kevin
	Confirm any mechanical alterations to the birdhouse with machine shop	John
Week of 4/1	This week and next can be used if any further debugging is needed with Pi but would prefer to use this time for prep for presentation / help teammates with anything else.	Colten
	Connect raspberry pi to microcontroller output signal	John
	Test product with battery only (Aside from raspberry Pi)	Kevin
Week of 4/8	Fix any minor bugs	Everyone
	Final presentation preparation	Everyone
Week of 4/15	Mock Demo	Everyone
Week of 4/22	Final Demo	Everyone
Week of 4/29	Final Presentation	Everyone
	Submit Final Paper	Everyone

Table 3: Schedule Table

Appendix C Video Capture R&V

Requirement	Verification
Timeliness of Recording Start: Raspberry Pi should initiate recording within 10 seconds upon signal from the microcontroller.	1. Measure the time delay between signal reception and the start of video recording using a timer. 2. Repeat the measurement multiple times and calculate the average and standard deviation of the delay. 3. Ensure the average delay is less than 10 seconds.
Video Data Processing Rate: The Raspberry Pi processes video data at an average rate of 0.4 MB/s in SD resolution.	1. Capture a short video clip (10 seconds) and measure the file size. 2. Calculate the processing rate (data size / recording time) in MB/s. 3. Repeat the measurement several times with different lighting conditions and object movements. 4. Ensure the average processing rate is close to 0.4 MB/s.
Video Quality: The recorded video clips should be at least 720p resolution	1. Capture a short video clip (10 seconds). 2. Manually review recorded video clips for artifacts, blurriness, or excessive noise.

Table 4: Video Capture Subsystem R&V

Appendix D Data Collection R&V

Requirement	Verification
Time of CSV Generation: The Raspberry Pi generates a new CSV file within 10 seconds of the designated time interval.	1. Record the timestamp of CSV creation. 2. Calculate the time difference between the expected and actual creation times. 3. Repeat the measurement multiple times and analyze the distribution of time differences.
CSV File Content: The CSV file contains all necessary data columns in the correct format.	1. Manually view the CSV file to view the data and see if the data is there.
CSV File Storage: The CSV file on the SSD has a unique and identifiable filename.	1. Manually check the filename of the generated CSV file to make sure it has the timestamp.
OpenCV: The accuracy should be within 10% of the accuracy of the model.	1. Flash images of birds to test the model before implementation and create a baseline for the model 2. Parse images from videos of live feed bird videos to test its drop from camera quality, etc.

Table 5: Data Collection Subsystem R&V

Appendix E Power System R&V

Requirement	Verification
Raspberry Pi: The power system should supply dc voltage to the Raspberry Pi between 4.75 - 5.25 volts	1. Connect a voltmeter to the 5.0V voltage regulator power output 2. Run the system through various operating scenarios (including idle, high load) 3. Ensure the voltage stays within 4.75 - 5.25 volts
ATmega328p: The power system should supply dc voltage to the microcontroller be 5.0 volts ± 0.5 volts [6]	1. Connect a voltmeter to the 5.0V voltage regulator power output 2. Run the system through various operating scenarios (including idle, high load) 3. Ensure the voltage stays within 4.5 to 5.5 volts

Table 6: Power System R&V

Appendix F Real Time Video Feed R&V

Requirement	Verification
Stream Delay: Delay between the stream and live should be within 15 seconds	1. Set up a simple test environment with the camera capturing both the live feed and recording for streaming. 2. Simultaneously introduce a visual cue (e.g., hand wave) in front of the camera. 3. Measure the time difference between the cue appearing in the live feed and its appearance in the streamed video.
Video Quality: The broadcast should as least be SD quality	1. Display the streamed video on a remote device capable of showing resolution details. 2. Continuously monitor the displayed resolution to ensure it never drops below SD quality.
Bird Name Display: The Broadcast should have a list of the current birds identified on the feed	1. If the bird can be identified check if a physical identifier ig name is displayed on the web server.

Table 7: Real Time Video Feed R&V

Appendix G Bird Feeder R&V

Requirement	Verification
PIR Performance in Simulated Weather: Sensor does not detect motion from inanimate objects	1. Connect the PIR and place it away from living objects 2. Use a tool to initiate motion from a distance while no living being is within range 3. Continuously monitor the sensor output. If any motion is detected during this process, consider it a failure
PIR Protection from weather: Transducer does not get damaged in severe weather	1. Visually inspect the mounted location of the transducer. Ensure it is fully sheltered from direct rain, snow, or hail. 2. If necessary, adjust the mounting position or add a protective cover to shield the transducer from water exposure. 3. Document the chosen protection method and its effectiveness in preventing water damage.
Environmental Protection for all electronics: The enclosure housing the electronics must be weatherproof	1. Visually inspect the enclosure for proper sealing around seams, ports, and cable entries. 2. Conduct a water spray test to verify the enclosure's resistance to rain and splashing water.

Table 8: Bird Feeder R&V

Appendix H Sensor R&V

Requirement	Verification
PIR Sensor: Motion sensing used to see if infrared can pick up hand in front of sensor	1. Put your hand in front of the motion sensor and check if the output from the sensor is high voltage which is 3.3V and once the hand is removed drop down to 0V. Voltage will be checked via a voltmeter.

Table 9: Sensing R&V