Automatic Pet Door System

ECE 445 Design Document

Fall 2023

Team 27

Haijian Wang (haijian4)

Haoran Zheng (haoranz8)

Zhihao Xu (zhihaox4)

TA: Wang, Yixuan

Professor: Arne Fliflet

February 23th, 2023

Contents

1. Introduction

1.1 Problem

For those people living near small natural ecosystems, human-wildlife conflicts could frequently happen, and one type of such conflict is intruding wild animals. As pet doors are designed for free entry and exit for medium-sized pets like cats and dogs, wild animals of similar size like raccoons, Skunks, opossums, and even coyotes can potentially intrude into people's houses through pet doors [1]. Although most intruding animals' incidents are not fatal, some of these wildlife animals can pose a threat to pets and humans: KCAL9 News reported that a coyote trespassed into a house in Buena Park through the pet door and killed a pet dog in 2019 [2]. Since wildlife animals' intruding is hard to predict, the most direct method to prevent wildlife animals from intruding is to lock the pet door. However, locking the door diminishes the benefits of pet doors as pets can no longer freely enter or leave the house on their own, so people need to make a tradeoff between safety and convenience. Other common countermeasures like equipping pets with protective vests could enhance the survival rate of pets, but they do not prevent wild animals from intruding.

1.2 Solution

After analyzing the problem, we make a consensus that the fundamental cause for such a problem is that, unlike the pet's owner, a pet door can neither distinguish between wild animals and pets nor lock/unlock on itself. Our goal is to design a pet door that can automatically make the correct decision: permitting pets freely enter and leave while preventing wild animals from intruding. In our planned design, when an object approaches our pet door, it should be able to execute the following four sequential steps without human intervention: detection, evaluation, decision, and action.

Detection: Our pet door should equip a variety of sensors to know whether an object has approached close enough for the system to make a locking/unlocking decision. Our main sensor is a camera installed on the Raspberry Pi, which will capture the outside view of the door in real-time. It will send the captured graph information to the Raspberry Pi for later evaluation processes. Additionally, we will also use an ultrasonic distance sensor mounted on the outside of the door to detect the presence of the object. It will send the detected distance to the microcontroller for later evaluation.

Evaluation: After receiving the graph from the camera, a Raspberry Pi with a pre-trained AI model will be responsible for analyzing the photo and judging whether the approaching object is a pet or wild animal. After the analysis, if the object is labeled as a pet, then Raspberry Pi will send a signal to the microcontroller for the later decision-making process.

Decision: The microcontroller will receive two previously mentioned signals: one from the ultrasonic distance sensor, and one from the Raspberry Pi. If the signal from Raspberry Pi is high, and the distance value sent by the ultrasonic distance sensor is within the predefined range, the microcontroller will send a high signal to gates of three MOSFETs that respectively control the electricity flows to an LED, a speaker, and an electrical-powered latch for a duration of 10 seconds, leaving enough time for pets to enter the house through our pet door. The duration of the signal will be constantly refreshed back to 10 seconds if the microcontroller keeps receiving a high signal from Raspberry Pi and a proper distance from the ultrasonic distance sensor.

Action: After receiving the signal from the microcontroller, the MOSFETs will be turned on and allow the electricity to flow to the latch installed at the inner side of our pet door. Then the latch will unlock itself and let the pet door be freely opened. At the same time, the LED will also be lit up, and the speaker will make sounds to indicate our pet door is currently unlocked once the MOSFETs are turned on. Ideally, the unlocking duration will be at least 10 seconds, and the timer will be constantly refreshed to 10 seconds if pet stays in front of the door without entering the house.

1.3 Visual Aid

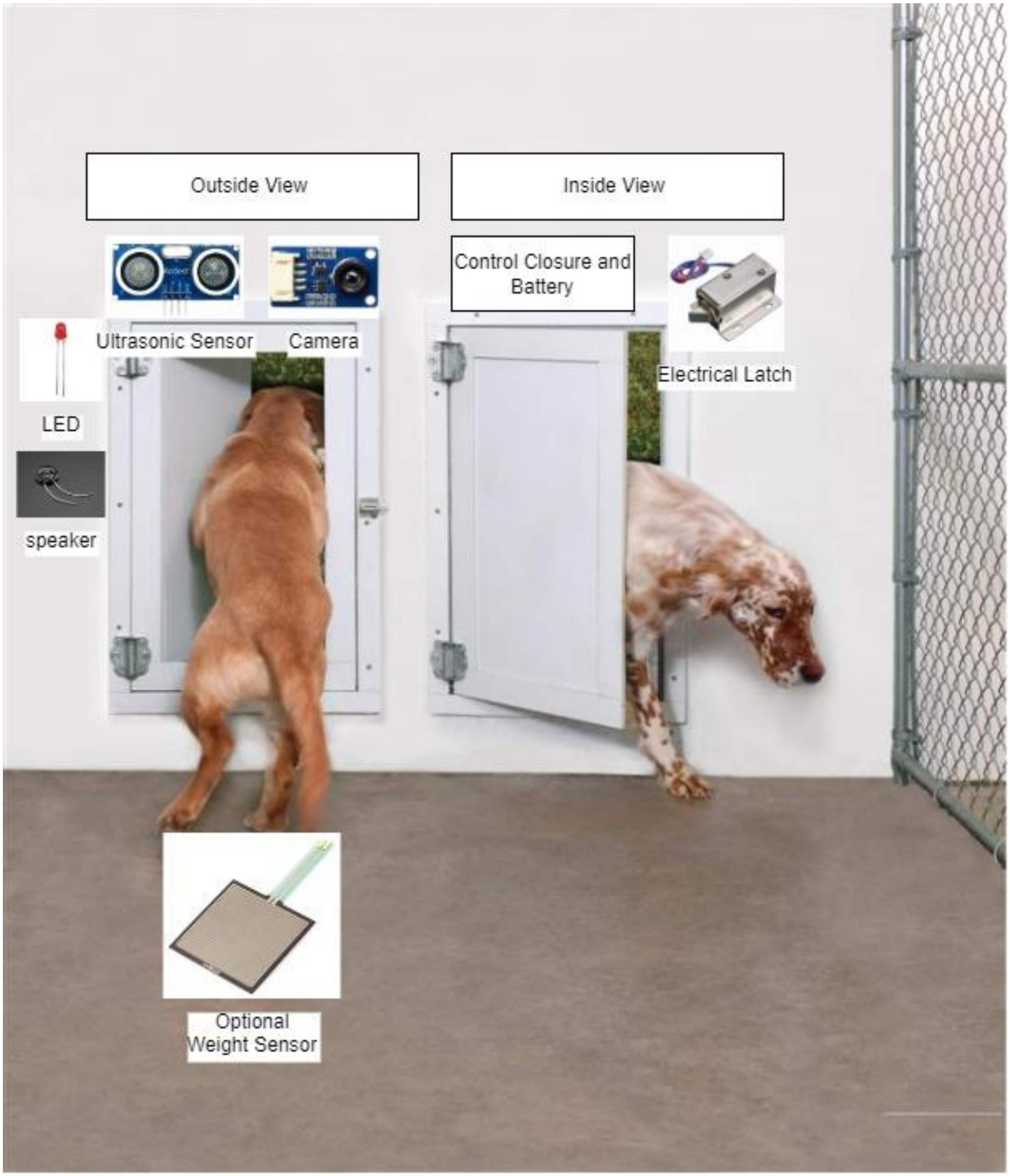


Figure 1: The demonstration of our pet door on both sides

1.4 High-level Requirements

To achieve our goal, we set the following standards for evaluating our final product. If all these requirements are satisfied, we will consider our project a success.

1. The pre-trained AI on Raspberry Pi should correctly categorize the object as pets with high accuracy. If the approaching object is a pet, the AI should categorize it as a pet and send a signal to unlock the latch at least 95 percent of the time. If the approaching object is not a pet, the AI should correctly categorize the object as non-pet and keep the latch locked 100 percent of the time.

2. Sometimes the pet may be far from the door and have no intention of entering the house, but the camera may still recognize it and send a signal to unlock the door. To prevent this, we add the ultrasonic distance sensor to ensure that the latch will only unlock if the pet is near the door. Therefore, the latch should only be unlocked if our microcontroller receives a high signal from Raspberry Pi and a proper small distance from the ultrasonic distance sensor. If any of these two conditions misses, the latch should stay locked.

3. If the electrical-powered latch is in unlocked state, it should automatically start the locking process 10 seconds after the camera fails to capture an image of pets. If the pet stays in front of the door, our camera should keep capturing the image of pets successfully, and the timer should be constantly refreshed to 10 seconds, leaving enough time for pets to enter the house through the pet door.

4. Pets should not wait for the unlocking process for too long, so whole Detection-Evalution-Decision-Action process should be completed in no more than 3 seconds. The electrical-powered latch according to its specifications could finish locking and unlocking process nearly instantly, so the processing time of Raspberry Pi should be within 1 to 1.5 seconds, and the microcontroller's response time should be within 1 to 1.5 seconds.

2. Design and Diagram

2.1 Block diagram

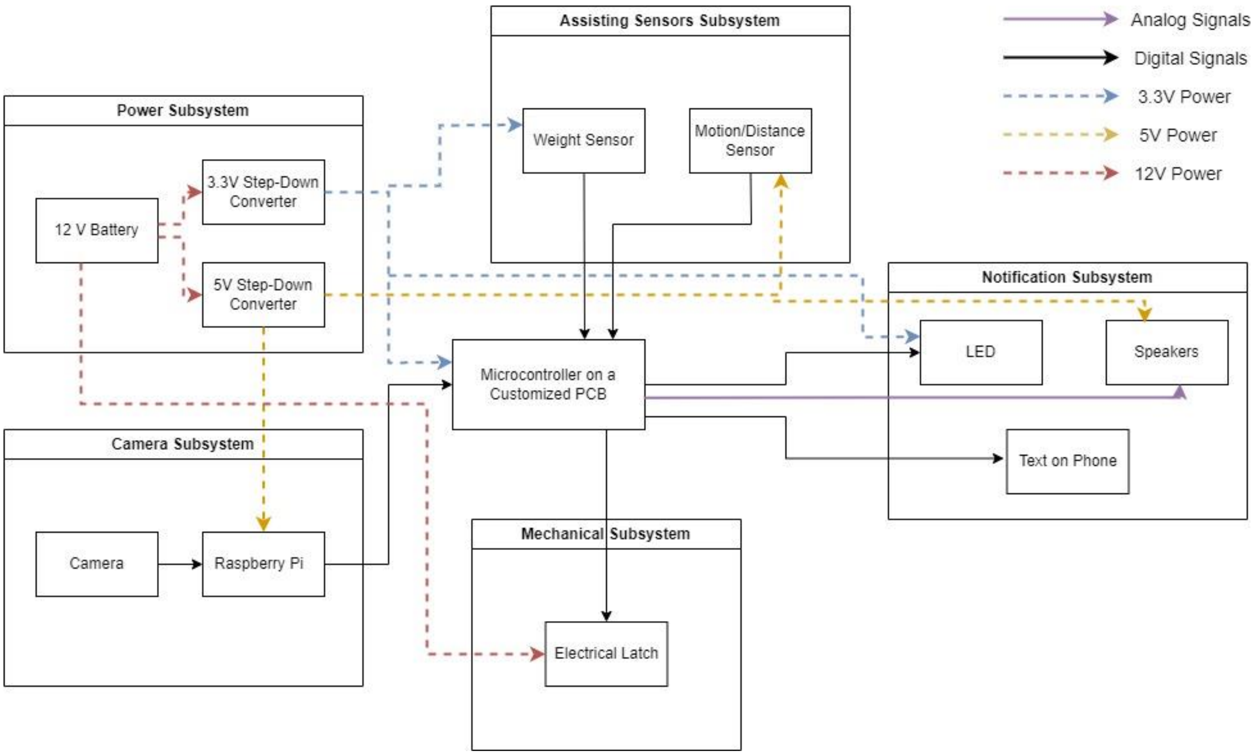


Figure 2: The block diagram of the whole system

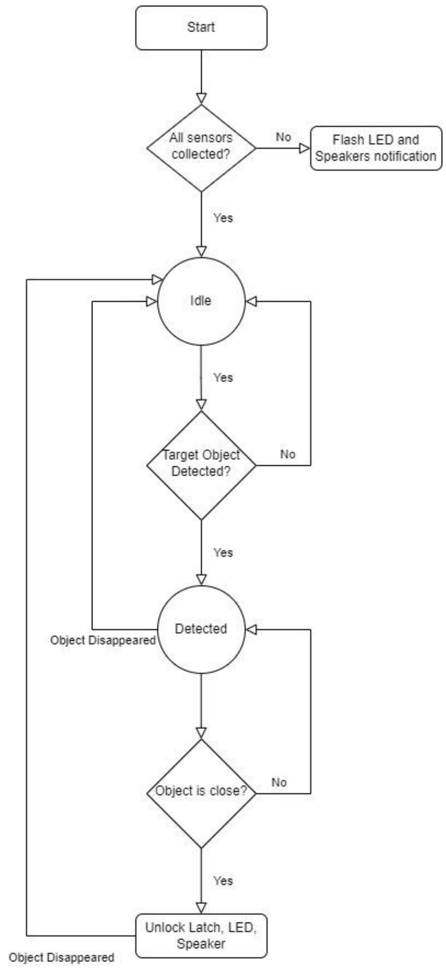
Small electrical components such as MOSFETs and resisters are ignored in the block diagram.

2.2 Crucial Component Specification

- HWE 12.8V 7Ah Battery
  - Nominal power capacity: 89.6Wh
  - Life for rated power: 7 hours
  - Size: 5.94 inch * 2.56 inch * 3.7 inch
  - Weight: 1.87 lbs
- RED WOLF Adjust DC 12V to 3.3V 5V 6V 9V DC Step Down Converter
  - Able to convert 12V to 5V for powering Raspberry Pi and speaker
  - Able to convert 12V to 3.3V for powering ESP32 microcontroller
- ESP32-S3--WROOM-2 Microcontroller:
  - CPU: Xtensa® dual-core 32-bit LX7 microprocessor, up to 240 MHz
  - Memory: 384 KB ROM, 512 KB SRAM
  - Important Peripherals: Camera interface, LCD interface, remote control, USB 1.1 OTG
- Ultrasonic Distance Sensor - HC-SR04:
  - Operating Voltage: 5V DC
  - Operating Current: 15mA

    - Ranging Distance: 2cm - 4m
    - 4 Pins: VCC (Power), Trig (Trigger), Echo (Receive), and GND (Ground)
- Raspberry Pi Camera Module 3:
    - 12-megapixel, Full HD video with autofocus at 50fps
    - Natural support by Raspberry Pi
- Raspberry Pi 4 Model B:
    - 5V 2.5A power supply would be the minimum requirement without connecting any USB peripherals.
    - OS, TensorFlow Lite, Python support
- Uxcell DC 12V 1.1A Electric Lock Cabinet Door Lock
    - Stroke & Force: 11.4mm, 300g
    - Lock Tongue Size: 11.4 x 10 x 10mm
    - Power off to lock, power on to unlock
- Uxcell a15080600ux0275 Metal Shell Round Internal Magnet Speaker
    - Resistance: 8ohm;
    - Power: 2W
    - Diameter: 28mm/ 1.1"; Material: Metal, Plastic
    - Weight: 29g
- Optoelectronics LED Indication – Discrete
    - Power dissipation: 75mW
    - Reverse voltage: 5 V
    - Forward current: 30mA
    - Pulse Forward Current: 100mA
    - Operating temperature range: -40 to 80
- IRF540PBF MOSFET
    - Drain-source voltage limit: 100V
    - Gate-source voltage: ± 20V
    - Turn-on delay time: 11ns
    - Turn-off delay time: 53ns
    - Operating temperature range: -55 to 175

2.3 Flow Chart



3. Subsystem Description

3.1 Camera Subsystem

This subsystem consists of a Raspberry Pi 4 Model B responsible for the Evalution step and a Raspberry Pi Camera Module 3 responsible for the Detection step.

Training mechanism: An AI model will be trained and tested on recognizing pets' facial images with many batches of manually labeled photos as inputs. The photos should include both the facial image of pets and other non-pet objects correctly labeled. The input photos will be randomly divided into training sets, development sets, and testing sets with no intersection between each set for easy overfitting detection. Each photo will be interpreted as an input vector, and each pixel will be interpreted as one element of this input vector. The training process could be manifest as a multi-layered neural network built on a linear regression model. In this neural network, adjacent layers are convolutionally connected. To reduce the linearity, we will choose proper activation functions such as ReLU to provide a transition between outputs from the previous layer and inputs of the current layer. In forward propagation, the AI will make its judgment and generate its own labeling. Then the AI will compare it to the manually assigned correct label. Then in the backward propagation, the AI will adjust the weight of each pixel to reduce the error gradually and thus improve labeling accuracy. The training process will be complete if the AI could correctly label the photos in the test sets with an accuracy higher than 95%. The above-mentioned training process will be done on our computer. The layout could be seen in figure 3.

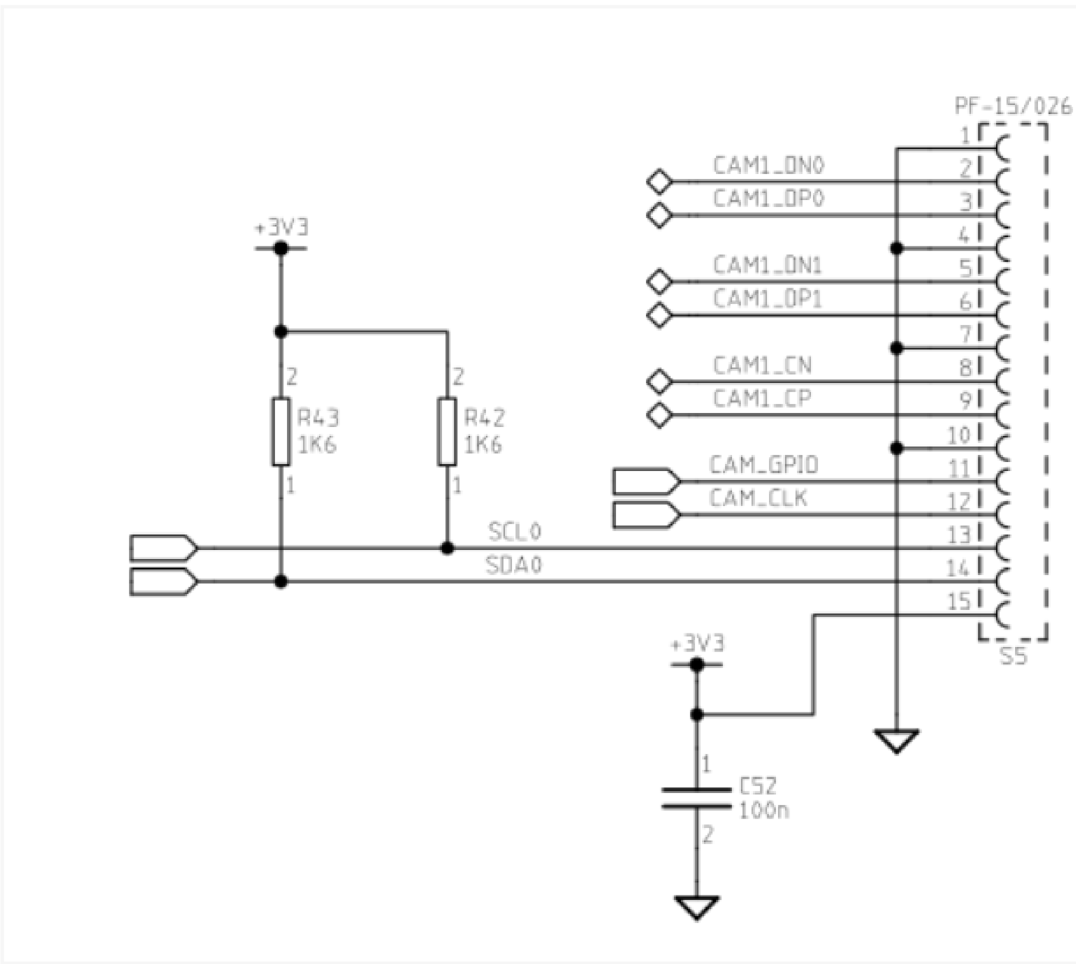


Figure 3: Schematic of the Raspberry Pi CSI camera connector

Raspberry Pi 4 Model B and Raspberry Pi Camera Module 3:

After finishing the training process on the computer, the AI will be loaded onto the Raspberry Pi 4 Model B and further tested in the real scenario. The Raspberry Pi

Camera Module 3 will monitor the outside of the door and send captured image data to Raspberry Pi as input. If AI's judgment is "pet", the Raspberry Pi will generate a high signal and send it to the ESP32 microcontroller. Otherwise, it will send a low signal. We will test the AI with several different breeds of dogs and expect our AI to label all of them as "pets". It is unrealistic to bring real wildlife for the tests, so we will bring other non-pet objects like cardboard boxes and expect our AI to label all of them as "non-pets". If the AI fails to correctly label the image sent by the camera and generates a false signal, we should further train the AI on the computer and load the improved version to the Raspberry Pi until it successfully passes all real-life test cases on Raspberry Pi. The detailed subsystem requirements and verification can be seen in Table 1.

Table 1: Camera Subsystem – Requirements & Verification

Requirement	Verification
• The camera module should function properly. It should be able to generate a full-color video stream with auto-focus and the right exposure. 1080P 60 FPS mode is preferred.	• Use the 15 Pin cable to Connect the Raspberry Pi Camera Module 3 to the Raspberry Pi 4 Model B board. • Connect the Raspberry Pi board to a monitor with a micro-HDMI cable. • Launch the Raspbian OS and run our Python test scripts. • The scripts will display the video stream and frame rate on the monitor. Check if the stream is stable and consistent. • Move around an object in front of the camera to test auto-focus.
• Our AI model should identify the category of a given object. When the AI is enabled, the video frame rate should be over 3 FPS.	• With the camera and monitor connected to the Raspberry Pi board, run our AI model's Python test script in the Raspbian OS. • The video frame rate should be above 3 FPS. Otherwise, we should optimize our code further with multithreading.

	• A colored box will be drawn around the identified object and its category will be displayed in text near the box. • Use real objects and pictures from the COCO dataset to test the accuracy of our model, which should be over 90%.
• Our AI model should generate a digital high signal while a target category object stays in the frame. In our case, the target categories are cats and dogs.	• Set the target categories in our Python script. For testing purposes, we will use common objects such as apples and bags. • Connect an LED test circuit to the output pin so that we can verify the signal visually. • Put the target object in front of the camera, and then remove the objects. Check if the behaviors of the LED satisfy our design requirements.
• Stability is important.	• Test our OS and hardware stability by running the OS for two days. • Test our program stability by running the program with full functionality for two days.

3.2 Motion Sensors Subsystem

This subsystem mainly consists of an ultrasonic distance sensor, so it is responsible for the Detection step. If conditions allow, an optional weight sensor will also be implemented in this subsystem.

Ultrasonic Distance Sensor - HC-SR04:

In some scenarios, pets may be far away from the door and have no intention to enter the house, but the camera still captures the facial image of the pets and unlock the latches, causing unnecessary power consumption and potential danger. To avoid this undesirable situation, we will add an ultrasonic distance sensor powered by one RED WOLF 12V to 5V converter to detect the presence of objects in front of our pet door. The ultrasonic distance sensor will emit sound waves as a signal and receive the reflected waves after hitting an obstacle within the practical range. If the reflected signal is different from the original signal, the ultrasonic sensor will consider it meets an obstacle. Once this ultrasonic distance sensor detects an obstacle (can be pet or non-pet), a high signal will be sent to the ESP32 microcontroller. The microcontroller will decide to unlock the latch and turn on the notification subsystem only if this signal and Raspberry Pi's signal are both high.

Optional weight sensor:

A weight sensor powered by the 3.3V converter can serve as a fail-safe: if the measured weight is lower or higher than the boundary of the expected weight range of normal cats and dogs, a signal will be generated and sent to the microcontroller. After receiving the signal, the microcontroller will ensure that latches remain locked even if the camera falsely recognizes the non-pet object as a pet. It is difficult to implement this sensor because pets may only be partially on this weight sensor, which will negatively influence the accuracy of this sensor, and some breeds of cats or dogs may have similar weight to wild animals like raccoons, so weight is probably not the most essential feature to distinguish pets from wild animals. Based on its merits and difficulty to implement, we consider this sensor optional. If we decide to implement this sensor in the future, we will make a more detailed plan.

Table 2: Motion Sensors Subsystem – Requirements & Verification

Requirement	Verification
• The ultrasonic distance sensor should receive the reflected signal after hitting an obstacle correctly	• the ultrasonic sensor should be powered by a 12 V to 5V converter to work with enough voltage • The location of ultrasonic sensor is set up near the camera and they are set up at a common horizontal line to make sure the signal is received simutaneously • the signal is sent to the microcontroller to unlock the latch which presents whether the signal is reflected correctly

3.3 Microcontroller & PCB

Microcontroller:

An ESP32-S3-WROOM-2 microcontroller powered by the 3.3V converter is the main control of the whole system, and it is responsible for the Decision step.

Specifically, this microcontroller receives signals from Raspberry Pi and obstacle's distance from the ultrasonic distance sensor. After receiving the distance of the obstacle, the microcontroller will compare it to a manually preset distance. If the obstacle's distance is smaller than the preset distance, and the signal from Raspberry Pi is high, the microcontroller will generate a high signal and send to the Gate of all three MOSFETs to power the electrical-powered latch, the LED, and the speaker. Physically, this microcontroller should be installed on the PCB board designed by us specifically for this project. The block diagram and schematics of ESP32 could be seen in figure 4 and figure 5.

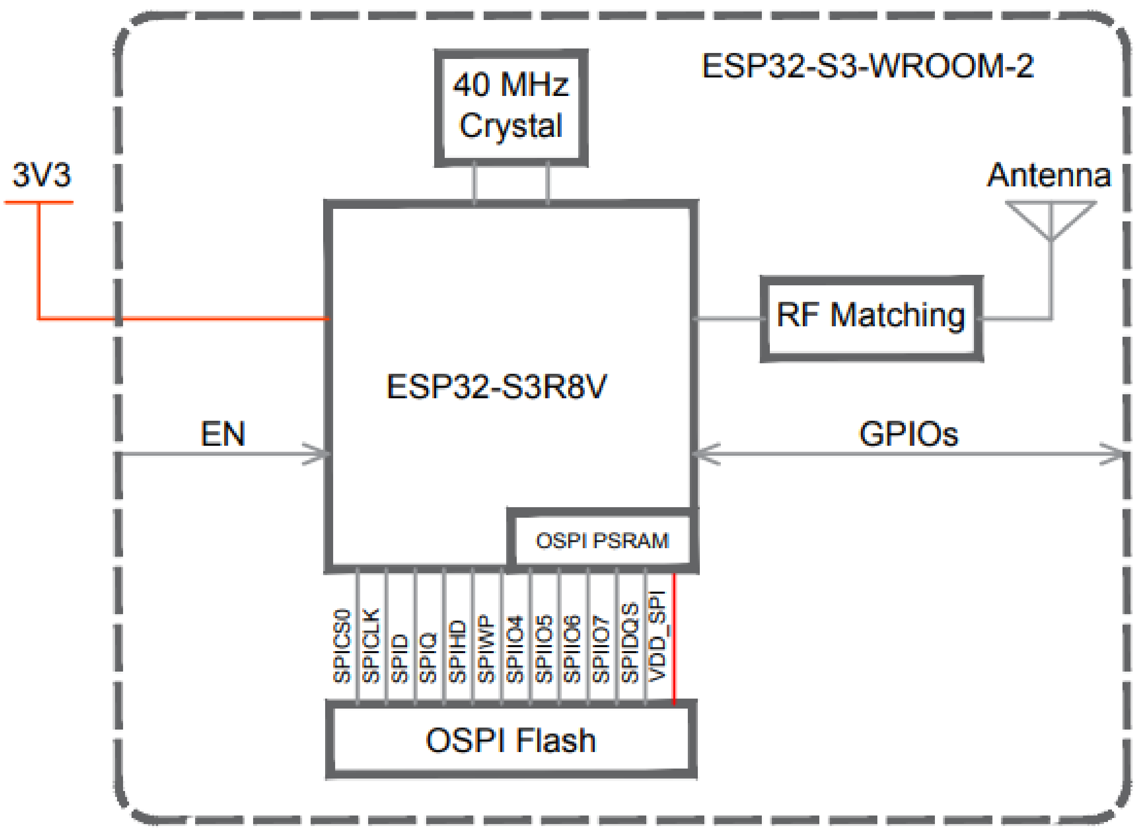


Figure 4: ESP32-S3-WROOM-2 Block Diagram

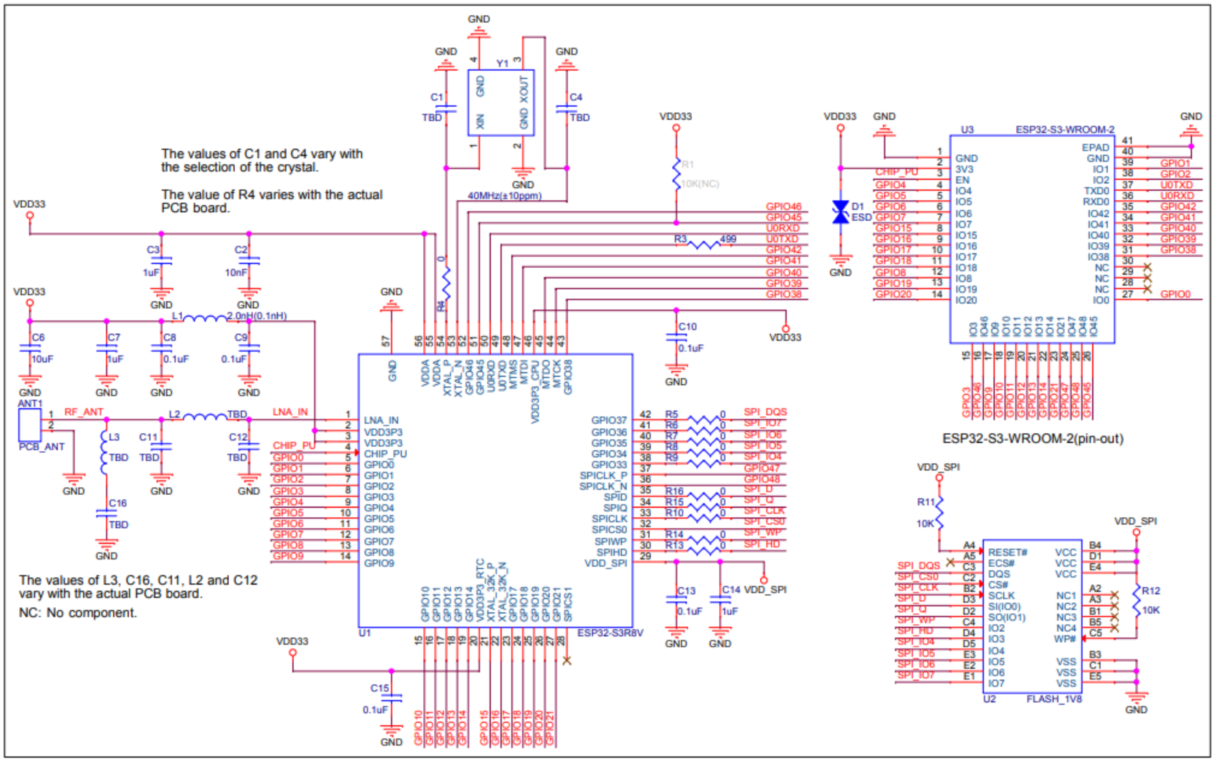


Figure 5: ESP32-S3-WROOM-2 Schematics

PCB:

Our PCB design will follow the official reference design. We will use a 40MHz crystal oscillator with two 20pF capacitors for the chip as required by the datasheet. We will have a debounced reset button at the CHIP_EN pin. The WIFI antenna design is optional. If we have extra time to implement functionality such as WIFI notification, we will go for a pi pad (attenuator) in our PCB design. For USB and UART connection, we are thinking about using a Type-C connector and CP2102 chip. If we spent way too much time debugging our PCB design, we will use an EP32 development board to proceed with the rest of our project.

Table 3: ESP32-S3-WROOM-2 Series Comparison

Ordering Code	Flash	PSRAM	Ambient Temp (°C)	Size (mm)
ESP32-S3-WROOM-2-N16R8V	16 MB (Octal SPI)	8 MB (Octal SPI)	–40 ~ 65	18 × 25.5 × 3.1

ESP32-S3-WROOM-2-N32R8V	32 MB (Octal SPI)	8 MB (Octal SPI)	–40 ~ 65	

Table 4: Microcontroller – Requirements & Verification

Requirement	Verification
• The microcontroller should generate control signals properly when the conditions are met.	• The input signals from the Raspberry Pi and distance sensor should be recorded properly. We will use an unused pin and an LED to visualize if the signals are detected and the circuits function properly.
• The PCB design should function properly	• Pass the DRC verification test • Satisfy the physical dimension limit. • Specifications of the official datasheet and hardware design manual are satisfied.
• The microcontroller should be programmed through the Type-C port	• We will use Arduino IDE with a customized pin assignment to program our microcontroller. • The USB port and UART connection should function properly. Otherwise, we will use a multimeter to test each soldered pin to debug.
• The whole board, especially the microcontroller, should operate at the recommended temperature range.	• We will use an IR thermometer to measure the temperature of the device under high loads and light loads. • If the temperature went above 90 °C, we will add a copper heat pipe in our design to dissipate heat.

3.4 Notification Subsystem

This subsystem is responsible for the Action step and consists of an LED, an electrical-powered latch, a speaker, and three MOSFETs. If conditions allow, an optional LCD screen will also be implemented in this subsystem.

Uxcell DC 12V 1.1A Electric Lock Cabinet Door Latch & MOSFET 1:

This latch will be in locked state if no voltage is applied. Once receiving a 12V input voltage, it will unlock the pet door and let the pets freely enter the house. Our HWE 12.8V 7Ah Battery should provide enough voltage and power for this latch. MOSFET 1 will control the electricity flow to this latch, and if the microcontroller sends a high signal to the Gate of MOSFET 1, it will close the circuit and allow latch to receive the 12V voltage and unlock the pet door. The latch should only be installed on the inner side of the pet door, so if the pet door is only locked from the outside, and the pets in the house can freely leave the house through the pet door from inside.

Optoelectronics LED Indication & MOSFET 2:

This LED will be lit up if a 3.3V voltage is applied. MOSFET 2 will control the electricity flow to this LED, and if the microcontroller sends a high signal to the Gate of MOSFET 2, it will close the circuit and allow this LED to receive a 3.3V voltage and be lit up. When the LED is lit up, it indicates that the object in front of the door is recognized as a pet and the latch is unlocked. Otherwise, it will be turned off and indicates that the latch is locked. The LED should be installed on both sides of the door, so the houseowner could read if the status of the pet door conveniently from both inside and outside.

Uxcell a15080600ux0275 Round Internal Magnet Speaker & MOSFET 3:

This speaker will generate sounds if a 5V voltage is applied. MOSFET 3 will control the electricity flow to this speaker, and if the microcontroller sends a high signal to the Gate of MOSFET 3, it will close the circuit and allow this speaker to receive a 5V voltage and emit sounds. When sounds are generated, it indicates that the object in front of the door is recognized as a pet and the latch is unlocked. Otherwise, no sounds will be generated and indicate that the latch is locked. The speaker should be directly mounted on the PCB board inside the house.

Optional LCD device:

An LCD screen will be connected to the microcontroller, and if the latch is unlocked, the word “UNLOCKED” will be displayed; Otherwise, the world “LOCKED” will be displayed. This device is very difficult to implement and may consume too much energy, which may negatively affect the performance of the whole system, so we consider the LCD screen optional.

An application on the pet's owner’s phone could remotely check the pet door status and send control signal to the pet door. This application is expected to be extremely difficult to implement on our own, so we consider this application optional. If we decide to implement this in the future, it should be wirelessly connected to the microcontroller and powered by the cellphone battery.

Table 5: Notification Subsystem – Requirements & Verification

Requirement	Verification
•The Latch, LED and speaker should be controlled by the microcontroller’s signal	•They should be powered by different converters to satisfy enough voltage. LED should be connected to a 12V to 3.3V converter. Speaker, the latch should be connected to 12 V to 5 V converter • The Latch, LED, and speaker should be connected to diverse MOSFETs separately. All three MOSFETs are connected to the microcontroller • microcontroller sends a high signal to MOSFET, the Latch will be unlocked, the LED will light up and the speaker will generate sound. Otherwise, the latch will keep locked, LED will not light and there are no sound generated

3.5 Power Subsystem

This subsystem consists of an HWE 12.8V 7Ah Battery and two RED WOLF Adjust DC converters. Although this subsystem does not directly participate in any of the four steps, it provides power to the rest of the system, so all other components can function properly.

HWE 12.8V 7Ah Battery:

This battery serves as the main power source of the system. When powering to the electrical-powered latch, it can directly supply its 12V voltage. When powering the ESP32 microcontroller, it should be connected with a RED WOLF Adjust DC converter and set the converted voltage to 3.3V for safety concerns. When powering the Raspberry Pi, the speaker, and the ultrasonic distance sensor, it should be connected with a RED WOLF Adjust DC converter and set the converted voltage to 5V for safety concerns.

Table 6: Power Subsystem – Requirements & Verification

Requirement	Verification
• The power subsystem should provide enough 3.3 V/5 V to the microcontroller, optional weight sensor, LED / Raspberry Pi, Ultrasonic Distance Sensor, latch, and the speaker	•Use a 12.8V 7Ah Battery and connect it with a DC step-down converter from 12 V to 3.3 V / 5V. •Use the voltmeter to measure the voltage after converting to ensure that it reaches the requirement of the microcontroller, optional weight sensor, LED / Raspberry Pi, Ultrasonic Distance Sensor, latch, and the speaker

3.6 Tolerance Analysis

One of the risks is that the checking of our systems will vary with the moving of pets. Every sensor system, motion system, and weight system should be satisfied at the same time as unlocking the latch. However, the deviation for the same pet will also oscillate among motion, sensor, and weight systems. Those systems will vary with the movement of pets. Pets' motions are not fixed. Another limit is our camera subsystem. The AI identification is unstable if the camera cannot catch the face of pets completely and clearly. The movement of the animal will influence the operation of the camera system and the door will also be limited because of this factor.

Symbol	Parameter	Min	Max	Unit
VDD33	Power supply voltage	–0.3	3.6	V
T_{STORE}	Storage temperature	–40	105	°C

Table 7: Absolute Maximum rating of Microcontroller

Symbol	Parameter	Min	Typ	Max	Unit
VDD33	Power supply voltage	3.0	3.3	3.6	V
I_{VDD}	Current delivered by external power supply	0.5	—	—	A
T_A	Operating ambient temperature	–40	—	65	°C

Table 8: Operating Conditions of Microcontroller

ABSOLUTE MAXIMUM RATINGS (T_C = 25 °C, unless otherwise noted)					
PARAMETER			**SYMBOL**	**LIMIT**	**UNIT**
Drain-source voltage			V_{DS}	100	V
Gate-source voltage			V_{GS}	± 20	
Continuous drain current	V_{GS} at 10 V	T_C = 25 °C	I_D	28	A
		T_C = 100 °C		20	
Pulsed drain current [a]			I_{DM}	110	

Table 9: Absolute maximum rating of IRF540PBF MOSFET

For the power supply system, we contain a 12 V battery with 7Ah.

The capacity of the battery = 12V* 7Ah = 84 Wh

The power consumption of the Raspberry Pi Camera Module 3 is between 0.4W to 1.4W.

For the DC-DC Step-Down 3.3V Converter with 300mA:

P = 3.3V* 0.3A = 0.99 W

For the ultrasonic Motion Sensor, its power consumption ranges from 1W to 5W.

The power consumption of Raspberry Pi 4 Model B is 15W

The power consumption of the latch is 12V *1.1A = 13.2 W

The power most of our chips and devices consume is the sum of the above devices:

Maximum: Ptotal = 1.4W + 0.99W + 5W + 15 W +13.2W= 35.59W

Minimum: Ptotal = 0.4W + 0.99W + 1W + 15W +13.2W= 30.59W

The minimum expected usage time between charging will be:

T = 22.2 Wh / 35.59W =0.623 Hour

The maximum expected usage time between charging will be:

T= 22.2 Wh / 30.59W = 0.725 Hour

Note that the usage time is calculated on the basis that the door is fully in unlocking state. Normally, the pet door will not always be unlocked, so the usage time will be significantly longer than the value calculated.

4. AI Design

We are going to design our AI pet identification system on Raspberry Pi OS. Since we are going to rely on Python libraries such as OpenCV and PyTorch, a lightweight Debian-based Linux system such as Raspberry Pi OS is ideal for our use case. As of right now, we decided to install the latest 32-bit Raspberry Pi OS n a microSD card for our Raspberry Pi 4 Model B board. We will change our OS version if we run into compatibility issues. To test our concept, we will first use a YOLOv3 pre-trained on the COCO dataset. YOLOv3 is a fast real-time object detection algorithm. COCO is a large-scale object detection, segmentation, and captioning dataset. COCO is great for our project because it has over 80 object categories including dogs, cats, hairs, and other common objects with context. E. Even if we might not be able to bring real pets for the demonstration, COCO would allow us to show the functionalities of our project with other objects. After we finish a working prototype, we will try to improve our system with the latest search results such as YOLOv7 and M3I-Pre-training. We think we won't have enough time to create a customized dataset, but it is certainly possible to customize the training for a specific dog or cat. We will pre-train our model and load the weights to the microSD for instant use. In our python script, we will have to help methods such as drawing colored boxes around the identified object to help us.

5. Cost and Schedule

Schedule:

Week	Task	Person
February 20 – February 24	Finish design document	Everyone
	Design review and order components	Everyone

February 27 – March 6	Prototype design on the bread board and check the problems in our design	Haoran Zheng
	Revise our design and check each module’s feasibility	Haijian Wang
March 6 – March 13	Begin PCB design and PCB review	Haoran Zheng
	Revise our PCB design	Zhihao Xu
	Place the order for PCB and team evaluation	Everyone
March 13 – March 20	Spring break	N/A
March 20 – March 27	Coding our AI part for our project and building connection with the sensor	Haijian Wang
	Integrate sensors with PCB and test the whole system	Haoran Zheng
	Revise the PCB design according to the testing results	Zhihao Xu
March 27 – April 3	Place the second order of PCB	Everyone
	Individual progress reports	Haijian Wang
April 3 – April 10	Finalize the camera system and integrate other sensor modules	Zhihao Xu
	Check the functionality of the whole system and debug the system	Haijian Wang
April 10 – April 17	Finalize the document and notebook	Everyone
	Prepare for the mock demo	Everyone
April 17- April 24	Mock demo	Everyone
April 24 – May 1	Finish final paper	Everyone

Cost analysis:

Description	Quantity	Manufacturer	Extended price
HWE 12.8V 7Ah Battery	1	HWE Energy	$34.56

RED WOLF Adjust DC 12V to 3.3V 5V 6V DC Step Down Converter Power Supply Voltage Adapter Reducer Regulator Fit	2	Red Wolf	$13.99
ESP32-S3--WROOM-2 Microcontroller	1	Espressif	$7.5
HC-SR04 ultrasonic distance sensor	1	Seeed Technology Co., Ltd	$4.3
Raspberry Pi Camera Module 3	1	Adafruit	$35.5
Raspberry Pi 4 Model B	1	Raspberry Pi	$128
Uxcell DC 12V 1.1A Electric Lock Cabinet Door Lock	1	Uxcell	$12.9
Uxcell a15080600ux0275 Metal Shell Round Internal Magnet Speaker	1	Dragonmarts - BISS	$10.55
Optoelectronics LED Indication – Discrete	10	American Opto Plus LED	$1.14
IRF540PBF MOSFET	3	ECE supply center	$1.97
Total costs: $250.41			

The Average hourly salary for a graduate student is $40 per hour, and we expect to spend 12 hours per week for 10 weeks.

$$(\$40/\text{hour}) \times 2.5 \times 10 \times 12 = \$12000$$

$$\text{Total costs} = 12000 + 250.41 = \$12250.41$$

6. Ethics and Safety

Ethics and safety are the most significant factors when identifying a design. We consider every possible risk related to our subsystem components including diverse sensors, latches, power supply, and motors. It is important to prevent the pitfalls of products that cause damage. For our product, the purpose is to protect the safety of pets and prevent the disturbance of other wild animals. The electrical components in our system will be kept away from dangerous factors including water, fire, and sharp objects to prevent harm. According to the 7.8 IEEE Code of Ethics I. 1, we need to hold paramount the safety, health, and welfare of the public and it is necessary to comply with ethical design and sustainable development practices. In order to protect the privacy of others, we also need to disclose promptly factors that might endanger the public or the environment. We will set the appropriate size of the door which is suitable for the crossing of pets. Meanwhile, the door will not be allowed to let other people or wild animals enter, preventing harm to privacy.

According to the 7.8 IEEE Code of Ethics III. 10, we need to strive to ensure this code is upheld by colleagues and co-workers. We should make sure every teammate follows the code of ethics and be honest with each other. Any information related to our design will be open to each other. Each teammate in our group finishes the Laboratory Safety Training and we will keep in mind every dangerous factor when working in the lab. We will keep at least two people staying in the laboratory and be careful about the actions of soldering and wiring. Meanwhile, our power supply system contains a 3.7 V battery to supply electrical energy for the operation of our product. It will be in an isolated environment around the door to prevent unpredictable dangers.

7. Reference:

[1] "Keep Raccoons from Using Pet Doors." *The Humane Society of the United States*, https://www.humanesociety.org/resources/keep-raccoons-using-pet-doors.

[2] Caplan, Christy. "Coyote Snuck through 'Doggie Door' to Kill Family Dog, Injure Another." *Pets*, 14 July 2022, https://www.wideopenpets.com/coyote-kills-dog-after-entering-through-doggie-door/.

[3] Winningham, Cathleigh. "Caught on Camera: Coyote Sneaks into Home through Doggie Door." *WKMG*, WKMG News 6 & ClickOrlando, 30 May 2022, https://www.clickorlando.com/news/weird-news/2022/05/30/caught-on-camera-coyote-sneaks-into-home-through-doggie-door/.

[4] "IEEE Code of Ethics." *IEEE*, https://www.ieee.org/about/corporate/governance/p7-8.html.

[5] Raspberry Pi. “Buy A Raspberry Pi Camera Module 3.” *Raspberry Pi*, https://www.raspberrypi.com/products/camera-module-3/.

[6] Raspberry Pi. “Raspberry Pi 4 Model B Specifications.” *Raspberry Pi*, https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/.

[7] Damen, Ryder. “Let the Dogs out! How to Make a Raspberry Pi Pet Detector.” *Tom's Hardware*, Tom's Hardware, 3 Apr. 2021, https://www.tomshardware.com/how-to/raspberry-pi-pet-detector.

[8] “03. Predict with Pre-Trained YOLO Models¶.” *03. Predict with Pre-Trained YOLO Models - Gluoncv 0.11.0 Documentation*, https://cv.gluon.ai/build/examples_detection/demo_yolo.html.

[9] “How to Make Your Own esp32 Breakout Board with Minimal Circuit - PCB Design Tutorial.” *PCBway*, https://www.pcbway.com/blog/PCB_Design_Tutorial/How_to_make_your_own_ESP32_breakout_board_with_minimal_circuit.html.

[10] BOY, Electro. “How I Made Own ESP32 Development Board.” *Hackster.io*, 12 May 2022, https://www.hackster.io/electroboy001/how-i-made-own-esp32-development-board-8af733.

[11] “ESP32-S3-WROOM-2-N16R8V ESPRESSIF Systems: Mouser.” *Mouser Electronics*, https://www.mouser.com/ProductDetail/Espressif-Systems/ESP32-S3-WROOM-2-N16R8V?qs=Rp5uXu7WBW%2Fbmj6VRWU1MQ%3D%3D.

[12] "Common Objects in Context." *COCO*, https://cocodataset.org/#home.