

**OUTCOME SPECIFIC ASSIGNMENTS
USED TO ASSESS OUTCOMES**

ASSIGNMENTS FOR OUTCOME “1”

An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematic.

**ELECTRICAL AND COMPUTER ENGINEERING
DEPARTMENT
OUTCOMES SPECIFIC ASSIGNMENT COVER SHEET
ELEG 4248 SENIOR DESIGN & PROFESSIONALISM II
SPRING 2022**

Instructor: VJ Turner & Sarhan Musa

Title of Assignment

Written Final Report used to evaluate this outcome

Outcome Measured Using this Assignment

Outcome 1

An ability to identify, formulate, and solve **complex engineering problems** by applying principles of engineering, science, and mathematic.

Check the complex engineering problem characteristics related to this assignment

Assignment problem	
X	involves wide-ranging or conflicting technical issues, has no obvious solution
	Addresses problems not encompassed by current standards and codes
	involves diverse groups of stakeholders
	includes many component parts or sub-problems
	includes many component parts or sub-problems
	involves multiple disciplines
	has significant consequences in a range of contexts
Assignment problem involves the application of the principles of	
X	engineering
	science
	mathematics

Brief justification of the suitability of assignment for the outcome

This section will demonstrate students' ability to define the project and project scope. Along with some key deliverables.

ELEG 4248 SENIOR DESIGN & PROFESSIONALISM II SPRING 2022

Title of Presentation: Outcome 1						
Team Number and Members:						
Date of Presentation:						
Name of Evaluator and Affiliation (University or Company name)						
	Excellent 99	Very Good 89	Average 79	Below Avg. 69	Poor 54	Comments
Identify and Formulate						
Able to understand the given problem and identify the subject area and concepts involved						
Able to identify what information is known and what needs to be determined						
Able to convert the problem into a well labeled sketch or appropriate representation (free body diagram, flow chart, functional block diagram, schematic diagram, etc.)						
Able to formulate the complex engineering problem into an engineering model or design – showing any math or science or technology needed to validate the design						
Solve Complex Problem						
Able to design a practical solution to address the problem.						
Able to identify parts and other needed capability (like coding) to solve the problem or proposal						
Able to deliver a final solution that integrates all aspects of the design from hardware to software						
The final solution should be consistent with project deliverables and able to demonstrate the objective has been achieved						

Electrical and Computer Engineering Department
ELEG 4248 Senior Design II
Spring Semester

TABLE OF CONTENTS

Project Executive Summary	i
Table of Contents	ii
List of Figures	iii
List of Tables and Graphs	iv
List of Team Members and Roles	v

- 1. PROJECT SCOPE** Please Number All Pages
- 1.1 Problem Statement, Business Objective and Target Audience of your Project
 - 1.2 Client identification and recognition of need
 - 1.3 Project goals and objectives
 - 1.4 Contemporary issues relevant to project
 - 1.5 Potential impact of project on society (Safety, Env, Tech, Eco)
 - 1.6 Initial Project constraints
- 2. PROJECT PLANNING AND TASK DEFINITION**
- 2.1 Task identification
 - 2.2 Gantt Chart – detailed chart not a high level summary
- 3. LITERATURE REVIEW**
- 3.1 List of all sources of information
 - 3.2 Statement of Learning from each source
- 4. PRELIMINARY DESIGN**
- 4.1 Concept Generation - describe other designs you considered
 - 4.2 Concept Evaluation and selection (evaluation other designs you considered / selection criteria)
 - 4.3 Design Constraints and Applicable Codes and Standards
Used in the Preliminary Design
 - 4.4 Preliminary Analysis - must show the science/math
 - 4.5 Preliminary Design
- 5. DETAILED SYSTEM DESIGN**
- 5.1 Project Parts Summary
 - 5.2 Engineering Analysis and/or Simulation must show the science/math
 - 5.3 Hardware Layout/Detailed Drawings/Equipment selection
and specifications (Coding only projects replace Hardware with Software and skip 5.3.1)
 - 5.3.1 Show Overview of Code Developed to Support 5.3 (only if you developed any code)
 - 5.4 Application of Engineering Codes and Standards used in Final Design
 - 5.5 Economic Analysis
 - 5.6 Final Design – show pictures or other example of working design
- 6. RESULTS & DISCUSSION**
- 6.1 Summary of Goals and Objectives – show deliverables and compare final deliverables with original
 - 6.2 Summary of Constraints and Codes met by Design – elaborate on codes/how you used in your project
 - 6.3 Discussion of Results and Other Use Cases – results summary and add other ideas/use cases

7. CONCLUSIONS AND RECOMMENDATIONS

LIST OF REFERENCES

APPENDIX (Include user manual or coding documentation if appropriate)

ITEMS in RED are Informational only – do not include the comments in red in your submission

ABOVE AVERAGE

Complex Engineering Problem Evaluation Form
ECE Senior Design & Professionalism
Outcome 1

Title of Presentation:

Team 5 – Object Detection

Joshua Littleton, Xavier Dukes, Arthur Quinney, Tri'Heem Neville, Christopher Rollerson

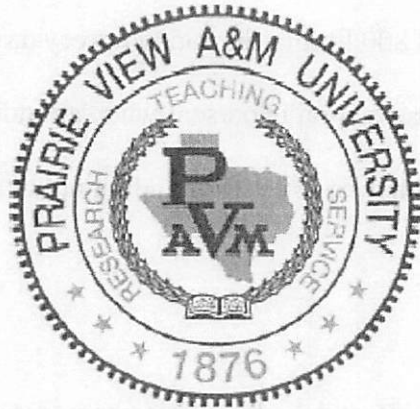
Date of Presentation:

Name of Evaluator and Affiliation (University or Company name)

	Excellent 99	Very Good 89	Average 79	Below Avg. 69	Poor 54	Comments
Identify and Formulate						Chapter 1
Able to understand the given problem and identify the subject area and concepts involved	✓					
Able to identify what information is known and what needs to be determined						
Able to convert the problem into a well labeled sketch or appropriate representation (free body diagram, flow chart, functional block diagram, schematic diagram, etc.)	✓					p4 25/29 Great
Able to formulate the complex engineering problem into an engineering model or design – showing any math or science or technology needed to validate the design	✓					Sections VV
Solve Complex Problem						Chapter 6
Able to design a practical solution to address the problem.	✓					
Able to identify parts and other needed capability (like coding) to solve the problem or proposal	✓					p431
Able to deliver a final solution that integrates all aspects of the design from hardware to software	✓					
The final solution should be consistent with project deliverables and able to demonstrate the objective has been achieved	✓					

Avg 99

PRAIRIE VIEW A&M UNIVERSITY
ROY G. PERRY COLLEGE OF ENGINEERING



Object Detection on Raspberry Pi

ELEG 4247: Senior Design & Professionalism II

Topic: Final Report

Team Number 5

Submitted by:

Joshua Littleton (Team Lead)

Tri'Heem Neville

Arthur Quinney

Xavier Dukes

Christopher Rollerson

Instructor: Professor Turner

Faculty Sponsor: Dr. Dong

Date of Submission: May 1, 2022

*Excellent
Report*

Project Executive Summary



Problem Statement

On average, approximately 4,800 burglaries happen every day. One-third of all robberies result in injury and 39% of the time a weapon is present when an individual is harmed. Arrest rates for Weapon offenses have doubled since 1965 including unlawful possession and trespassing.

Proposed Deliverables

- We will be using a Raspberry Pi attached with a camera to survey the area and identify potential threats
- The machine learning (ML) algorithm we implement detects the threatening objects on an individual such as weapons in real-time
- Once an object defined as a threat is detected, a text notification is sent to the owner/operator's phone allowing them to determine the course of action

Budget

- The total budget for our project is \$600.00
- The total amount spent on our project is \$340.00

Project Executive Summary (continued)

Customer requirements And Engineering solutions

- Mass producible by using easily obtainable/cost-effective parts
- Lightweight parts for mounting purposes
- Train machine learning algorithm to detect various weapons
- The camera needs a resolution of 1080p for increasing the range of detection
- Text notification alerts of potential threats
- The algorithm classifies threats and non-threats
- Detect threats in real-time

Major Constraints

- The housing for the Raspberry Pi and camera must be mountable
- The camera resolution affects detection accuracy and distance
- The limited frame rate of the Raspberry Pi camera

Benefits

- **Societal:** This project will offer greater security to those that cannot afford other options. Businesses will be capable of guaranteeing the safety of customers without infringing upon a customer's 2nd amendment right to carry while within the public space.
- **Safety:** The primary goal of this project is to create an affordable security system for small businesses and homeowners to proactively respond to potential threats however they see fit.
- **Technical:** Machine learning has not been utilized within security systems to this day. Our project will pioneer a smart security system that will inform you of potential dangers

Project Executive Summary (continued)

at the first sighting of a weapon that would provide a danger to security or the owner.

Recommendations

We recommend that the unit is mounted to a stable elevated surface, ensure there is nothing blocking the camera lens, make sure the Raspberry Pi is located in a well-insulated area to prevent overheating the unit, provide a power source for the Raspberry to connect to, and lastly to get the best usages of our product do not forget to set up a Twilio account for all alerts and notifications.

Conclusion

Our design will identify active threats by revealing displayed weapons. This artificial intelligent threat detection system will impact many from universities and church organizations to businesses and homeowners.

Table of Contents

Perfect ✓

PROJECT EXECUTIVE SUMMARY	II
TABLE OF CONTENTS	V
LIST OF FIGURES	VII
LIST OF TABLES AND GRAPHS.....	VIII
LIST OF TEAM MEMBERS AND ROLES	IX
SECTION 1 - PROJECT SCOPE	1
1.1 Problem Statement, Business Objective and Target Audience.....	2
1.2 Client Identification and Recognition of Need.....	3
1.3 Project Goals and Objectives.....	6
1.4 Contemporary Issues Relevant to the Project.....	7
1.5 Potential Impacts of the Project.....	8
1.5.1 Societal Impact:	8
1.5.2 Safety Impact:.....	8
1.5.3 Technical Impact:	9
1.6 Initial Project Constraints	11
SECTION 2 - PROJECT PLANNING AND TASK DEFINITION.....	12
2.1 Task Identification	13
2.2 Gantt Chart.....	14
SECTION 3 - LITERATURE REVIEW.....	16
3.1 List of All Sources	17
3.2 Statement of Learning from Each Source	20
SECTION 4 - PRELIMINARY DESIGN	22
4.1 Concept Generation.....	23
4.2 Concept Evaluation and Selection.....	24
4.3 Constraints and Applicable Codes and Standards Used in Preliminary Design	26
4.4 Preliminary Analysis.....	27
4.5 Preliminary Design.....	28
SECTION 5 - DETAILED SYSTEM DESIGN	30
5.1 Project Parts Summary.....	31
5.2 Engineering Analysis	32
5.3 Hardware Layout/Detailed Drawings/Equipment Selection and Specifications	35
Hardware Layout.....	35
Detailed Drawings	35
Equipment Selection and Specifications.....	44
5.4 Overview of Code	49
5.5 Application of Engineering Codes and Standards used in Final Design.....	54
5.6 Economic Analysis	55
5.7 Final Design - Show Pictures or Other Examples of Working Design.....	56
SECTION 6 - RESULTS AND DISCUSSION	63
6.1 Summary of Goals and Objectives	64
6.2 Summary of Constraints and Codes met by Design	65
6.3 Discussion of Results and Other Use Cases.....	67

Table of Contents (continued)

SECTION 7 - CONCLUSIONS AND RECOMMENDATIONS	68
LIST OF REFERENCES	R1
APPENDIX.....	A1
CODE TO RUN CAMERA AND SEND NOTIFICATION:	A2
CODE FOR THE TRAINING OF THE MODEL	A8

List of Figures

Outstanding

FIGURE 1.1: EXAMPLE OF WEAPON DETECTION OF A FIREARM.....	10
FIGURE 1.2: EXAMPLE OF WEAPON DETECTION OF A KNIFE	10
FIGURE 2.2: GANTT CHART (PART 1).....	14
FIGURE 2.2: GANTT CHART (PART 2).....	14
FIGURE 2.3: GANTT CHART (PART 3).....	15
FIGURE 2.4: GANTT CHART (PART 4).....	15
FIGURE 3: PRELIMINARY DESIGN.....	28
FIGURE 4: SOFTWARE FLOW CHART	29
FIGURES 5.1-5.8: IMAGES OF PARTS.....	31
FIGURE 6: LABELING SOFTWARE.....	32
FIGURE 7: HARDWARE LAYOUT.....	35
FIGURE 8: HOUSING DESIGN	36
FIGURE 9.1: FRONT CAMERA PLATE.....	37
FIGURE 9.2: SIDE CAMERA COVER.....	37
FIGURE 9.3: CAMERA VISOR	38
FIGURE 9.4: BACK CAMERA PLATE	39
FIGURE 9.5: HINGE.....	39
FIGURE 9.6: UPPER & LOWER ARM	40
FIGURE 9.7: LONG & SHORT SCREWS	41
FIGURE 9.8: PI CASE TOP.....	42
FIGURE 9.9: PI CASE BOTTOM.....	43
FIGURE 10.1: CAMERA DIMENSIONS	49
FIGURE 10.2: LOADING LABEL MAP	50
FIGURE 10.3: LOADING IN THE MODEL.....	50
FIGURE 10.4: BOXES, SCORES, CLASSES, AND NUMBER OF BOXES.....	51
FIGURE 10.5: FRAME RATE CALCULATION.....	51
FIGURE 10.6: RUNNING DETECTION MODEL	52
FIGURE 10.7: VISUALIZATION OF RESULTS	52
FIGURE 11.1: TWILIO SETUP	53
FIGURE 11.2: SMS NOTIFICATION EXECUTION	53
FIGURE 12.1: RIFLE DETECTED WITH 95% CONFIDENCE.....	56
FIGURE 12.2: RIFLE DETECTED WITH 100% CONFIDENCE	56
FIGURE 12.3: COVERED RIFLE DETECTED WITH 97% CONFIDENCE	57
FIGURE 13.1: HANDGUN DETECTED WITH 84% CONFIDENCE	57
FIGURE 13.2: HANDGUN DETECTED WITH 97% CONFIDENCE	58
FIGURE 13.3: BARREL POINTED AT CAMERA DETECTED WITH 75% CONFIDENCE.....	58
FIGURE 13.4: PARTIALLY COVERED HANDGUN DETECTED WITH 93% CONFIDENCE	59
FIGURE 13.5: BLACK HANDGUN DETECTED ON BLACK JACKET WITH 99% CONFIDENCE.....	59
FIGURE 14.1: BLADE DETECTED WITH 91% CONFIDENCE	60
FIGURE 14.2: BLADE DETECTED WITH 84% CONFIDENCE	60
FIGURE 14.3: NARROW PROFILE OF BLADE NOT DETECTED	61
FIGURE 15: EXAMPLE NOTIFICATION	61
FIGURE 16.1: CELL PHONE FALSE-POSITIVE TEST (PART 1)	62
FIGURE 16.2: CELL PHONE FALSE-POSITIVE TEST (PART 2)	62

List of Tables and Graphs

TABLE 1: MATRIX	5
GRAPH 1: PERCENTAGES OF WEAPONS USED IN ROBBERIES.....	9
TABLE 2: PROJECT COMPARISON CHART	25
TABLE 3.1: RASPBERRY PI 4 MODEL B SPECIFICATIONS.....	44
TABLE 3.2: RASPBERRY PI CAMERA SPECIFICATIONS	45
TABLE 3.3: CORAL USB ACCELERATOR SPECIFICATIONS	46
TABLE 3.4: POWER SUPPLY SPECIFICATIONS.....	46
TABLE 3.5: MICRO-HDMI TO HDMI CABLE SPECIFICATIONS	46
TABLE 3.6: MICROSD CARD SPECIFICATIONS.....	47
TABLE 3.7: PI-FAN SPECIFICATIONS	48
TABLE 3.8: POLYLACTIC ACID SPECIFICATIONS.....	48
TABLE 4.1: PHASE 1 COST.....	55
TABLE 4.2: SMART THREAT DETECTION SYSTEM COST.....	55

List of Team Members and Roles

Joshua Littleton Team Lead ✓

Arthur Quinney Parts Procurement Lead

Christopher Rollerson Design Analysis Lead

Tri'Heem Neville Scribe and Presentation Manager

Xavier Dukes Software Developer

Section 1 - Project Scope

1.1 Problem Statement, Business Objective and Target Audience

Across the United States, a burglary happens every 15 seconds. This equates to a staggering 4800 burglaries a day on average. A burglary is defined as an unlawful entry of a structure to commit a felony or theft. Two-thirds of these burglaries affect residential properties while the remaining third affect small businesses. In 2020, 33,296 robberies involved a handgun, and an additional 8,641 involved a knife. A robbery can be more severe than a burglary, as it must involve threats or actual violence. Arrest rates for weapon offenses have doubled since 1965 including unlawful possession and trespassing. However, according to the U.S. Department of Justice, Bureau of Justice Statistics, a substantial portion of violent crimes that take place in homes have taken place during a burglary incident.

The goal of this project is to design an intelligent monitoring system that will enhance our ability to survey public and private spaces by using object detection to identify various threats that could potentially lead to security risks and/or bodily harm.

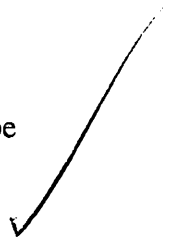
The target audience for this project is small businesses, homeowners, hospitals, schools/universities, and churches.

1.2 Client Identification and Recognition of Need

- i. The device should be mass-producible; to achieve this we plan to use easily obtainable parts that are cost-effective. Due to the target audience being homeowners and small businesses, we want to make this device available to everyone at a reasonable price.
- ii. The final product must be mountable and lightweight; this will be achieved by using lightweight materials and cheaper parts. This will ensure that it can be mounted on a wall for extended periods of time without any complications. The final product will be placed in conventional security camera locations, to survey the area.
- iii. This device must detect a variety of potential threats as weapons come in various shapes, sizes, and calibers. This is a perfect application for machine learning (M.L.) as there is not any set of definite rules to classify the myriad of types of weapons there are. The flexibility required for this particular use case lends itself very well to the use of artificial intelligence (A.I.).
- iv. The detection of potential threats at a distance is imperative to the preemptive Smart Detection System. A high degree of accuracy is required to minimize false positives. Poor resolution can affect accuracy and may not give an adequate warning since the potential threat will have to be closer to the camera when compared to a higher resolution camera. The camera will have a minimum resolution of 1080p to accomplish the goal of high accuracy, but we are also limited by the processing power of the Raspberry Pi.
- v. The device needs to be able to communicate with other devices such as a phone or monitor to deliver the alert notification system. A text message should be sent

1.2 Client Identification and Recognition of Need (continued)

immediately, via Twilio, to the home or business owner for appropriate action to be taken.



- vi. Object detection must be able to distinguish between a threat and a non-threat entity. The Raspberry Pi has limited processing power and should not waste valuable CPU power on low-risk or non-threatening individuals. The M.L. algorithm should focus on items that are deemed as potentially dangerous.
- vii. Real-time tracking is essential as the camera will give real-time data for the environment it is monitoring. The delay between the image processing and the time the data is received should be minimized for the most effective results. Raspberry Pis is particularly suited for real-time environments as they are often used in robotics due to their real-time clock (RTC) functionalities.

1.2 Client Identification and Recognition of Need (continued)

Marketing Requirements	Engineering Requirements	Justification
Mass Producible	Easily obtainable and cost-effective parts	Target audience is homeowners and small business owners
Mountable/Lightweight	Lightweight parts sturdy enough to be mounted for extended periods	Will be placed in conventional security camera locations
Detect a variety of potential threats	Train machine learning (M.L.) algorithm for weapon detection	Threats come in various forms, and this is an abstract problem with no clear formula. Requiring the flexibility of Artificial Intelligence
Detect potential threats at a distance	Camera has minimum resolution of 1080p	Poor resolution images are more difficult to decipher and can cause inaccuracies
Alert Notification System	Send an alert through a text message	Alerts the home/business owner of the potential threat near the camera
Distinguishable between threat and no threat	M.L. model will be trained to detect items deemed as threats	Camera's tracking will not focus on low-risk individuals and will dismiss them
Real Time Tracking	Use Raspberry Pi for its real time clock (RTC) capabilities	Raspberry Pi utilizes RTC which is essential for real time environments

Table 1: Matrix

1.3 Project Goals and Objectives

The goal of this project is to design an intelligent monitoring system that will enhance our ability to survey both public and private spaces by using object detection to identify various threats. These threats could lead to security risks and or bodily harm. Utilizing machine learning (M.L.) within the Raspberry Pi, we will monitor and detect any potential threats and non-threats that would enter the surveyed area. Once an object is determined to be a threat, the homeowner or business owner will be alerted so that they may take action that they deem as necessary. The notification will be sent before any violent action takes place, making the system more proactive rather than reactive. The warning that is received is essential to the safety of the occupants of the property and any type of warning in a potentially dangerous situation can aid in the mitigation of severe bodily harm.

1.4 Contemporary Issues Relevant to the Project



Security is an essential part of our society, and our project is aimed at providing better security to small businesses, homeowners, hospitals, schools/universities, and churches. In our society, we have seen a drastic increase in gun violence and gun violence rates. In 2020, there was a 34% increase in gun violence deaths from the previous year, this also results in a 75% increase over the previous decade. Another important statistic to point out is that 79% of murders were carried out with a gun. If our project were to be implemented in our society these rates would start to decrease because people would be notified when there is a weapon present in their area.

1.5 Potential Impacts of the Project

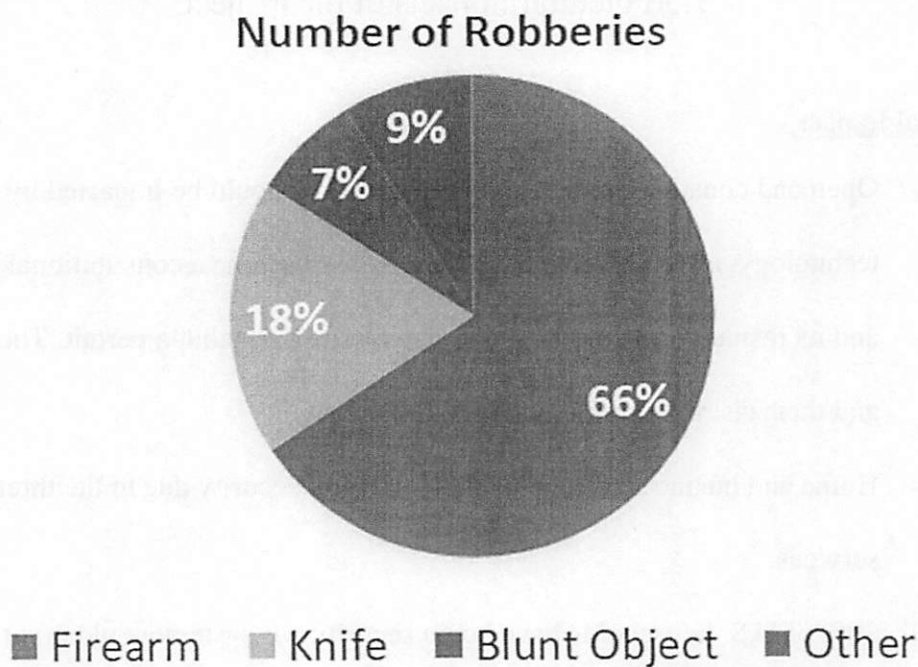
1.5.1 Societal Impact:

- I. Open and constitutional carry in several states could be impacted by this technology, as of September 1, 2021, Texas became a constitutional carry state, and its residents could openly display firearms without a permit. These weapons and their visibility in society will increase
- II. Home and business owners will have greater security due to the threat detection services
- III. 30% of U.S. households have home security our system would grant greater security

1.5.2 Safety Impact:

- I. In 2019, there were an estimated 1,117,696 burglaries in the United States. 62.8% of those burglaries were residential properties, 37.2% of those burglaries were businesses.
- II. According to Statista, there were a total of 124,381 robberies that involved weapons in 2019. 66% of those robberies involved a firearm and 34% involved a knife or some other weapon.
- III. Statistics show that someone is present 27.6% of the time when a burglary is taking place, 7.2% of these burglaries result in someone being injured.
- IV. Graph 1 shows the percentage of a certain type of weapon is used in robberies. This influenced our decision on what weapons our device will detect.

1.5 Potential Impacts of the Project (continued)



Graph 1: Percentages of Weapons used in Robberies

1.5.3 Technical Impact:

- I. Our project utilizes machine learning algorithms for the smart detection of weapons. The benefit of using M.L. is that our system will be able to learn on its own with the ability to detect new weapons
- II. Weapon detection using M.L will be done in real-time, which has not yet been implemented in conventional home and small business security systems, some examples of weapon detection are in Figures 2.1 and 2.2
- III. Weapon detection allows for the security system to be more proactive rather than reactive. This will help the homeowner/business owner to prevent the possibility of injury to themselves or others
- IV. The figures below what a weapon detection interface would look like.

1.5 Potential Impacts of the Project (continued)

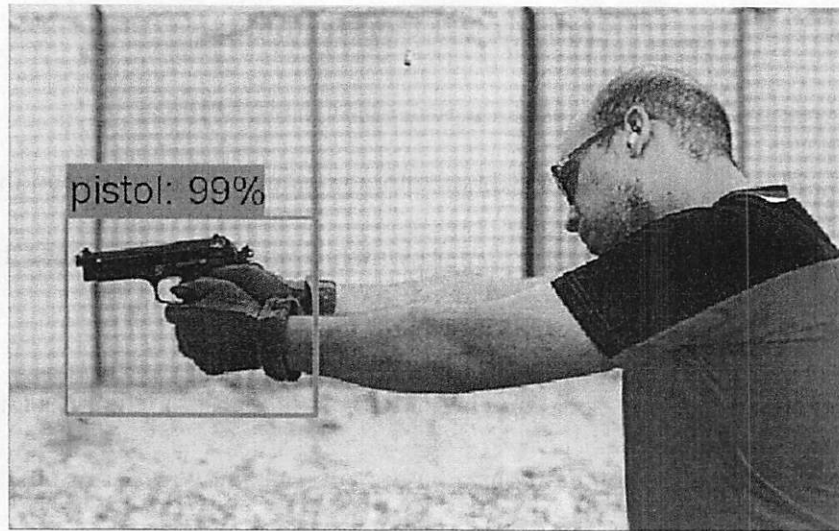


Figure 1.1: Example of Weapon Detection of a Firearm

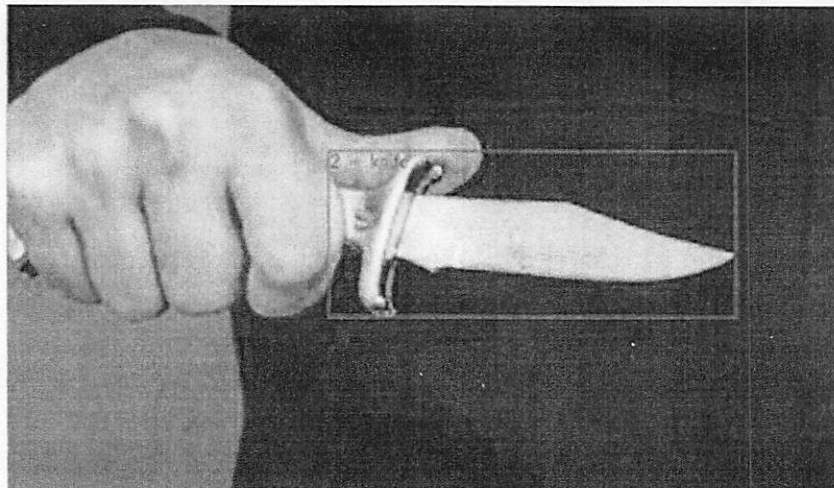


Figure 1.2: Example of Weapon Detection of a Knife

1.6 Initial Project Constraints

There are a few constraints that will influence the success of our project.

- The first is that the housing for the Raspberry Pi and camera must be mountable. This is due to our project acting as a security camera which is normally mounted to the wall in an unreachable location.
- The second constraint for our project is the camera resolution, which affects detection accuracy and distance. If our camera has a bad resolution, we will not be able to clearly detect the weapons at a distance and will have inaccurate readings.
- Our final constraint is the limited frame rate of the Raspberry Pi camera, which will also affect our detection accuracy and has an impact on our alert notification system.



Section 2 - Project Planning and Task Definition

2.1 Task Identification

The team began planning first by conversing with the faculty sponsor to choose the specialization for object detection we would implement. This built a foundation on what type of research will be needed, as well as how much data is necessary for the project to start, beginning with the understanding of object detection and the procurement of parts. The research period allowed for each group member to learn about Raspberry Pi and the operating system interface that went along with it making for the future steps to have a smoother transition.

The design and coding phase of the project allowed the group to learn the coding language, Python, and design the script for the project to run. While designing the project we began early-stage testing to determine if the code worked as expected and find if there were any errors that needed to be fixed before moving to the next step. After testing the code and determining it works as intended, the next phase was to specialize the project in weapon detection instead of just object detection. To accomplish this, an edit to the code is necessary as well as training for the machine learning model to be capable of learning and recognizing set weapons that would appear before the camera. After changing the code and training, the first prototype was complete; however, the next step was to solve the framerate issues and implement a notification system to the code.

The final phase was completing the notification system for the Raspberry Pi and solving the framerate constraint to accomplish real time tracking and provide accurate detection for when a weapon appeared on camera. These tasks ranged between being completed within a day and taking weeks, so the delegation of time and work needed to be allocated appropriately so that the project is capable of being completed before the allotted due date.

2.2 Gantt Chart

Excellent

Throughout the year, time management is important for guaranteeing the project will be completed within a timely manner and that each member is working on their respective part by delegating the workload for the team to work as efficiently as possible. The following images are the complete Gantt chart displaying the detailed time delegation for each task to ensure the project progressed smoothly within the time limit.

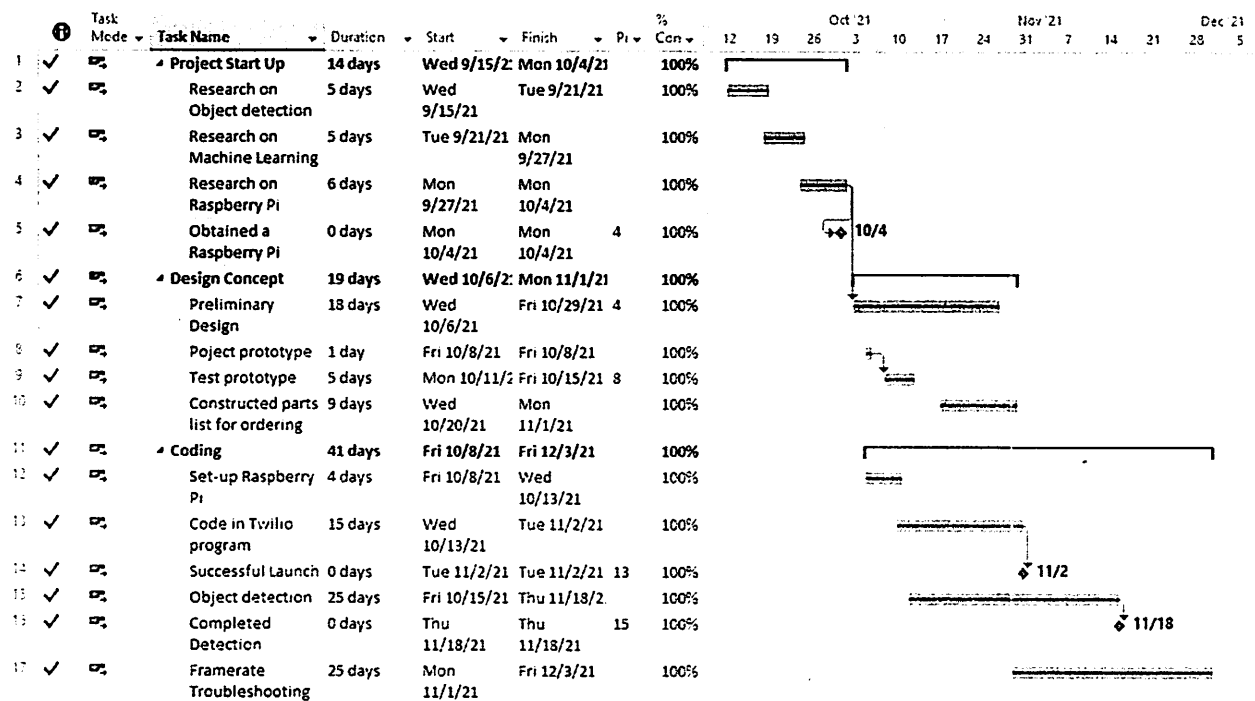


Figure 2.2: Gantt Chart (part 1)

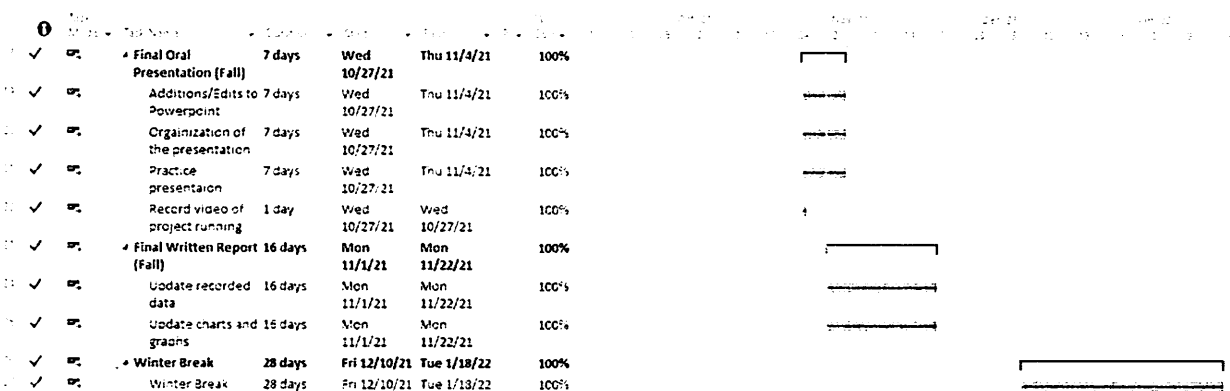


Figure 2.2: Gantt Chart (part 2)

2.2 Gantt Chart (continued)

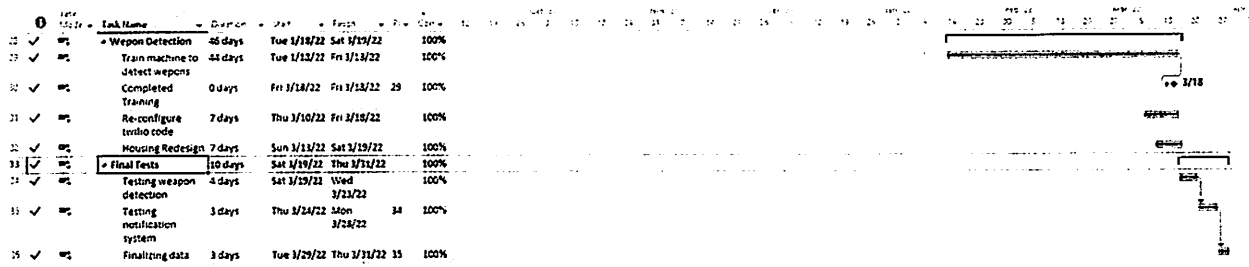


Figure 2.3: Gantt Chart (part 3)

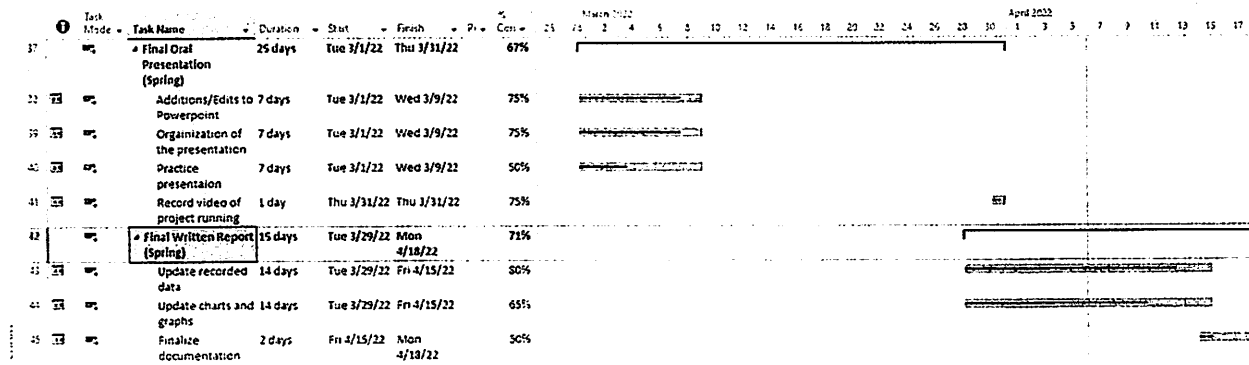


Figure 2.4: Gantt Chart (part 4)

Section 3 - Literature Review

3.1 List of All Sources

1. Best Home Security System Company & Security Cameras,” Deep Sentinel, 24-Sep-2021. [Online]. Available: <https://www.deepsentinel.com/>. [Accessed: 04-Oct-2021].
2. “Burglary Statistics: The Hard Numbers,” Alarms.org, 19-Dec-2019. [Online]. Available: <https://www.alarms.org/burglary-statistics/>. [Accessed: 04-Oct-2021].
3. “Frequently Asked Questions About Small Businesses,” U.S. Small Business Administration Office of Advocacy, Oct-2020. [Online]. Available: <https://cdn.advocacy.sba.gov/wp-content/uploads/2020/11/05122043/Small-Business-FAQ-2020.pdf>. [Accessed: 04-Oct-2021].
4. “Home Security Systems,” SimpliSafe, 2021. [Online]. Available: <https://simplisafe.com/>. [Accessed: 04-Oct-2021].
5. “Introducing VEDX: GPT-3 For Vision,” Visual one. [Online]. Available: <https://www.visualone.tech/index.html#python>. [Accessed: 04-Oct-2021].
6. J. Arora, “Theft Detection Using Machine Learning,” Medium, 25-Sep-2020. [Online]. Available: <https://medium.com/analytics-vidhya/theft-detection-using-machine-learning-a4232ea51f1c>. [Accessed: 04-Oct-2021].
7. L. Blain, “Vaak's AI theft-detection system is already 81% accurate,” New Atlas, 11-Apr-2019. [Online]. Available: <https://newatlas.com/vaak-vaakeye-ai-theft-detection/59263/>. [Accessed: 04-Oct-2021].
8. “More security with security cameras systems,” GW Security. [Online]. Available:

3.1 List of All Sources (continued)

<https://www.gwsecurityusa.com/>. [Accessed: 01-Oct-2021].



9. "Product Help," SimpliSafe. [Online]. Available: <https://support.simplisafe.com/hc/en-us>. [Accessed: 04-Oct-2021].
10. "Weapons Offenses and Offenders", U.S. Department of Justice. [Online]. Available: <https://bjs.ojp.gov/content/pub/pdf/woofccj.pdf> [Accessed: 29-Mar-2022]
11. S. Catalano, "National Crime Victimization Survey: Victimization During Household Burglary," Office of Justice Programs, 28-Sep-2010. [Online]. Available: <https://bjs.ojp.gov/content/pub/ascii/vdhh.txt>. [Accessed: 04-Oct-2021].
12. S. Cho and N. Tin, "Using infrared imaging technology for concealed weapons detection and visualization," TENCON 2010 - 2010 IEEE Region 10 Conference, 2010, pp. 228-233, doi: 10.1109/TENCON.2010.5685995.
13. S. K. Bandyopadhyay, B. Datta, and S. Roy, "Identifications of Concealed Weapon in a Human Body," Arxiv. [Online]. Available: <https://arxiv.org/ftp/arxiv/papers/1210/1210.5653.pdf>. [Accessed: 04-Oct-2021].
14. S. Perkins, "Heating up the search for Hidden Weapons," Science News for Students, 03-Dec-2019. [Online]. Available: <https://www.sciencenewsforstudents.org/blog/eureka-lab/heating-search-hidden-weapons>.
15. S. Stasha, "18+ Intriguing Home Ownership Statistics for 2021," Policy Advice, 13-Sep-2021. [Online]. Available: <https://policyadvice.net/insurance/insights/home-ownership-statistics/>. [Accessed: 04-Oct-2021].

3.1 List of All Sources (continued)

16. "Self-installed DIY Home Security Systems," Cove Security, 04-Oct-2021. [Online]. Available: <https://www.covesmart.com/>. [Accessed: 04-Oct-2021].
17. Statista Research Department, "Number of robberies in the United States in 2019, by weapon used," Statista, 01-Oct-2020. [Online]. Available: <https://www.statista.com/statistics/251914/number-of-robberies-in-the-us-by-weapon/>. [Accessed: 04-Oct-2021].
18. T. Covington, "Burglary Statistics, Research, and Facts," The Zebra, 08-Jul-2021. [Online]. Available: <https://www.thezebra.com/resources/research/burglary-statistics/>. [Accessed: 04-Oct-2021].
19. Uniform Crime Reporting Program, "Crime in the U.S. 2019 Table 23," FBI: UCR, 29-Aug-2019. [Online]. Available: <https://ucr.fbi.gov/crime-in-the-u.s/2019/crime-in-the-u.s.-2019/tables/table-23>. [Accessed: 27-Sep-2021].

3.2 Statement of Learning from Each Source



1. Reference to another security device for monitoring of area; Deep Sentinel device
2. Break-in and burglary statistics regarding injuries and the presence of weapons provided via FBI Uniform Crime Reporting Statistics
3. Economic significance and statistics regarding the presence of small businesses
4. Project comparison to SimpliSafe products
5. How to integrate and set up a web client via python to send message to Twilio messaging service
6. Discussion of crime statistics and how ML can be used in burglary prevention and identification
7. Implementation of Vaak's AI theft-detection system and the methods of training
8. Project comparison to GW security camera products and their various prices and capabilities
9. Various SimpliSafe products for comparison to our product
10. Bureau of Justice Statistics Weapons Offenses and Offenders statistics on firearm related criminal offense over the years and its growth
11. The victimology of burglary victims according to the Department of Justice statistics, rates of burglary and harming of victims
12. Exploring the possibility of infrared cameras and object detection, for this project to detect concealed weapons
13. How to detect concealed weapons against the human body and when direct sightlines are obstructed

3.2 Statement of Learning from Each Source (continued)

14. The possibility of using infrared and heat sensitive wavelengths of light with various photo processing techniques to detect concealed weapons
15. 18+ homeowner statistics, touches on various aspects of homeownership such as reports on increasing cost, less overall homeowners, and disparities in the housing market
16. Cove Smart products and how they potentially compare to our product sensitive wavelengths of infrared light
17. Visual data and statistics regarding the number of times a weapon was present during robberies and the types of weapons during these occurrences
18. Stats on the number of burglaries, how often homes had security measures preinstalled and the economic effect such acts have on the victims
19. Statistics regarding the type of locations affected by burglaries

Section 4 - Preliminary Design

4.1 Concept Generation

There are several applications that machine learning can be used for such as object detection, image classification, natural language processing, facial recognition, and much more. As a group, we decided to focus on a specific implementation of object detection in relation to threat detection. The project shifted more into the development of a pre-emptive warning and threat detection system with associated hardware and third-party text notification alert system.

4.2 Concept Evaluation and Selection

As a company, we have a favorable position to produce our goods and services at a differential value. We have demonstrated a greater comparative value than that of our competitors and conveyed this information to our target market. Production of our unique technology that offers features our competitors are not marketing paired with a lower production cost will ensure our competitive advantages.

Our technology includes a weapon detection algorithm whereas our rivals offer a general object detection algorithm. We have machine learning and real-time tracking software whereas some of our rivals only offer machine learning software. Our organization will not charge a continuous monthly service charge as many of our rivals include in their packages. Our single unit product is remarkably simple, easy to install, and lightweight, whereas our rivals have multiple heavier hardware devices and equipment to install.

With every great invention comes a challenge, and for our system to be complete, we must rely on a third-party notification company by the name of Twilio. Twilio plays a large factor in our equation as to when a threat is determined by our system. Twilio is responsible for sending out the proper notifications to the selected entities. Twilio is an independent company that the consumer will need to purchase a package plan from for our product to be utilized to its fullest capacity.

4.2 Concept Evaluation and Selection (continued)

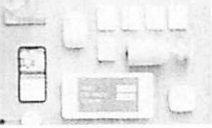
Design Options	Initial Cost	Continuous Fees	Pros	Cons	References
Smart Threat Detection System (Our Design)	\$340	No Monthly Fee	- Weapon Detection - Machine Learning (ML) - Real Time Tracking - Light Weight - Text Notifications	- No remote monitoring - Twilio subscription needed	N/A
GW Security 	Starts at \$1400 to \$6000	No Monthly Fee	- 4k Resolution - Uses ML - Night Vision	- No Remote Viewing - Heavy Camera Equipment - No Weapon Detection	GWsecurityusa.com BBB.com Amazon.com
Deep Sentinel 	\$400 to \$1200	Monitoring Fee \$60 to \$200 per month	- Uses ML - Motion Detection - Wi-Fi or Ethernet connection	- No Weapon Detection	Deepsentinel.com
SimpliSafe 	Starts at \$700	Monitoring Starts at \$15 per month	- Smoke, water, carbon & motion sensors - Smart home integration	- No ML feature to enhance system	Simplisafe.com
Cove Smart 	Starts at \$1200	Monitoring \$26 per month	- A+ rating - Environmental Sensors - Text Notifications	- No smart ML features available	Covesmart.com BBB.com

Table 2: Project Comparison Chart



4.3 Constraints and Applicable Codes and Standards Used in Preliminary Design

One of the major design constraints for our project was the frame rate of the Raspberry Pi camera. The frame rate affects Twilio alert times as it works via a counter of frames with a weapon present. Frame rates are essential to our project because we need to be able to detect weapons clearly and accurately before an individual is harmed. To fix this issue we implemented the Google Coral USB Accelerator, this allowed our system to go from one to two frames a second to around 30 frames a second.

The following Codes and Standards were used in implementing our preliminary design:

IEEE 3652.1-2020: Federated Machine Learning - Our machine learning model can be constructed from data distributed across various repositories owned by various companies

IEEE P2671: Online detection with machine vision - This standard specifies through the general requirements of online detection based on machine vision

IEEE P2247.3: Recommended Practices for Evaluation of Adaptive Instructional Systems - best practice incorporates and promotes the principles of ethically aligned design for the use of artificial intelligence

IEEE 829-2008: IEEE Standard for Software and System Test Documentation - We used this standard when testing our software for detecting weapons

IEEE 2755.2-2020: Recommended Practice for Implementation and Management Methodology for software Based Intelligent Process Automation

IEEE 7.8: IEEE: Code of Ethics

- 3) To be honest and realistic in stating claims or estimates based on available data
- 5) To improve the understanding of technology, its appropriate application, and potential consequences

4.4 Preliminary Analysis



After determining the deliverables for our project, we were then able to research and investigate ways to complete our project. We decided that we should start with object detection on our Raspberry Pi 3b+, object detection is to detect instances of visual objects within a certain class in digital images. With object detection, we were able to detect preprogrammed objects such as chairs, cellphones, laptops, a person, and many other objects. Next, we would test object detection on our Raspberry Pi 3b+ and investigate the machine learning code so that we could determine how to go from object detection to weapon detection, which is the basis of our project. The way to do this was by “labeling”, labeling is the process of identifying raw data and adding meaningful and informative labels to provide context so that a machine learning model can learn from it. This will be addressed later in the paper. Referring to our object detection tests, we noticed that we were only getting around 1.1 to 2.5 frames per second, this would cause a delay in the timing of our text message alert system. This issue is due to the Raspberry Pi 3b+ being an older model machine that could not handle the video processing that we needed it to do. Therefore, we had to do more research to find a solution to address this problem, we arrived at the Google Coral USB Accelerator which can be plugged into our Raspberry Pi to give us a higher frame rate. The accelerator would then bring our frame rate from a maximum of 2.5 frames per second to around a maximum of 30 frames per second. We also found a newer model Raspberry Pi that was more powerful than the current one we were given; the new model was the Raspberry Pi 4b. We decided to purchase these new products that had a greater benefit on our overall project success.

4.5 Preliminary Design

Once we were given our project at the start of the fall semester, we began to consider ways that we could create the design of our project. We knew that we would need a Raspberry Pi and the associated camera, because that was the required hardware specification for our project and the camera allows us to monitor the area. Aside from the hardware of our project we also decided that we would have an alert notification system set up to alert the owner of the device that a weapon has been identified in their area. The alert notification comes through a third-party company known as Twilio that users will have to subscribe to and provide a phone number to send an alert message.

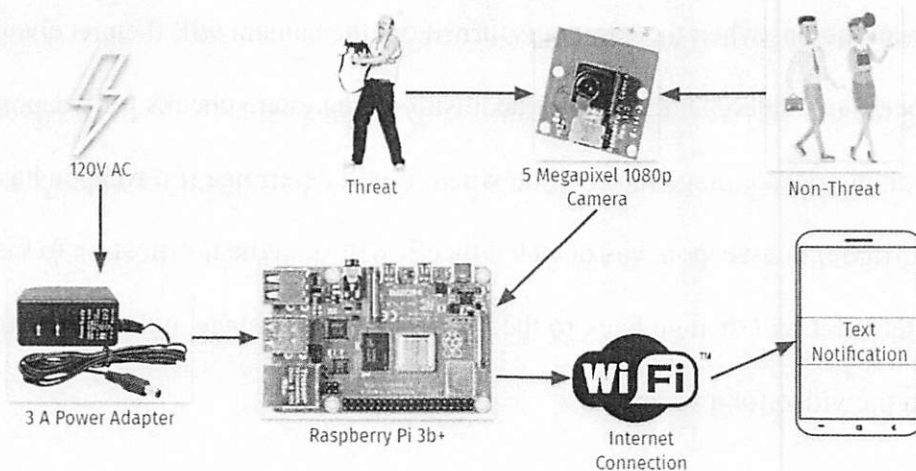


Figure 3: Preliminary Design

Once we obtained the hardware for our preliminary design, we put it together so that we could test it; the preliminary design can be seen in figure 3 above. The Raspberry Pi 3b+ is plugged in via a 3-amp power adapter, the camera is attached to the Raspberry Pi to detect threats and non-threats, and once a threat is detected an alert text message is sent over a Wi-Fi internet connection.

4.5 Preliminary Design (continued)

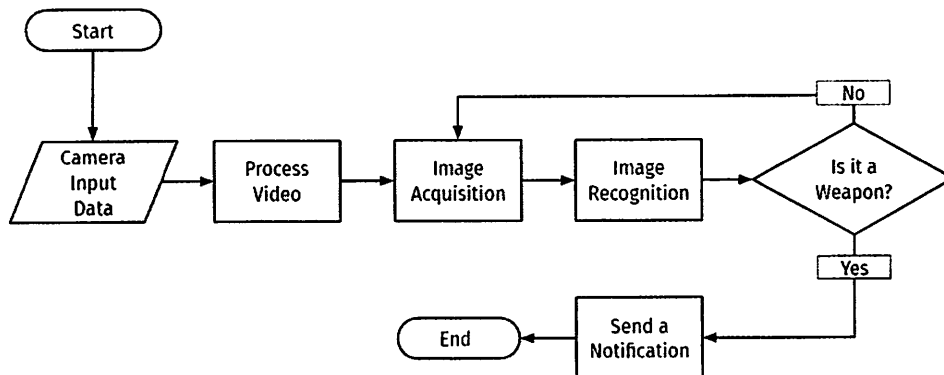


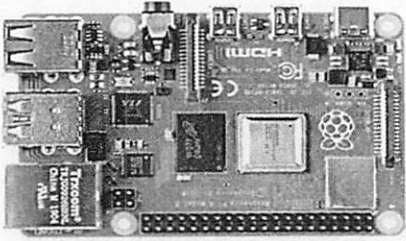
Figure 4: Software Flow Chart

In figure 4, we show the software flow chart on how the detection and notification system works. The software begins when the camera is turned on, the camera will then receive input data and will process the video. During image acquisition the system checks for weapons frame by frame, which then goes to image recognition where it will determine if a weapon has been identified in that frame. If a weapon was detected then it will send the notification to the user, if no weapon is detected it will then go back to the image acquisition stage and continue to check the next frame in the video for a weapon.

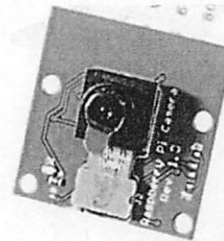
Section 5 - Detailed System Design

5.1 Project Parts Summary

All parts used in the final system design are listed below in Figures 5.1 to 5.8.



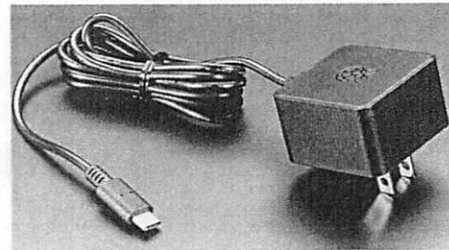
5.1: Raspberry Pi 4 Model B



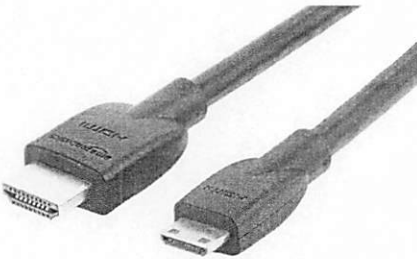
5.2: Raspberry Pi Camera Rev 1.3



5.3: Google Coral USB Accelerator



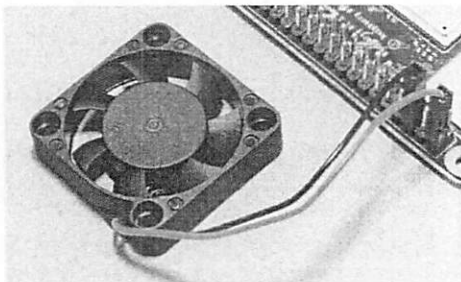
5.4: USB-C Power Supply



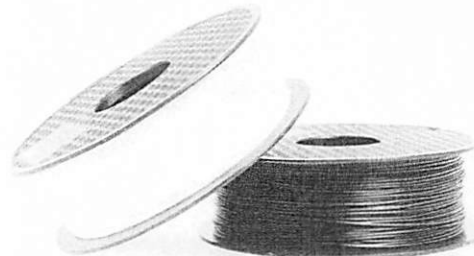
5.5: Micro HDMI to HDMI Cord



5.6: Micro SD Card



5.7: Raspberry Pi Fan



5.8: Polylactic Acid

Figures 5.1-5.8: Images of Parts

5.2 Engineering Analysis

Labeling and Training:

In order to train the model to detect weapons, we first had to label images that contained weapons. This was done using a software called LabelImg. A bounding box is drawn on the portion of the image that contains the weapon and you attach a label to that box. An example of this is shown in the figure below.



Figure 6: Labelling Software

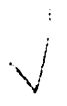
Here you can see that two handguns have been boxes and labeled as handguns. This had to be done for over 5000 images.

After the entire dataset is labeled, it is used to train the model. The model uses the information provided by the labeled images to recognize patterns and make predictions.

Quantization:

Quantization is the act of constraining an input and representing its value as a smaller value. Each node, or tensor, of the model is represented by some numerical value. Usually, a 64-

5.2 Engineering Analysis (continued)



bit floating-point value can represent an absurdly broad range of values and is computationally intensive to decipher or to use in other operations. This large computational load slows the processing of frames and therefore the frame rate. Shrinking the representation of these nodes to 8-bit unsigned integers can greatly improve the frame rate, as operations on simpler values are faster and take up significantly less space in memory.

Quantization can be applied to the model on a per tensor basis or to the outputs of individual layers which only partially optimizes the model. This process can decrease the overall size of the model when it is totally quantized by a factor of 4 times. Quantization is also essential for the implementation of the Coral USB Accelerator Tensor Processing Unit (TPU) produced by Google which will maximize the frame rate on the Raspberry Pi upwards of 25-30 frames per second (fps).

Model:

The model used for object detection was the SSD MobileNet V2 COCO model. The SSD acronym at the beginning of the model's name stands for Single Stage Detection, meaning that this algorithm for identification goes straight from pixels of an image directly to bounding box coordinates and class probabilities to be displayed around the object. This model was chosen because of its compatibility with TFlite as only a select pool of lightweight operators is available with this framework and not all models can be converted to this version. This model is also one of the few quantizable models which is necessary for frame rate solutions to be implemented such as the Coral USB accelerator.

5.2 Engineering Analysis (continued)

Notification System:

Short Message Service or SMS is used as our notification system. A third-party software called Twilio is integrated into our design. Whenever a weapon is seen for a set number of frames, the notification is sent to warn anyone in danger. This type of notification was used because of its easy implementation, as well as its ability to send out notifications to any number of people with any type of phone. The messages are very quick and reliable.

5.3 Hardware Layout/Detailed Drawings/Equipment Selection and Specifications

Hardware Layout

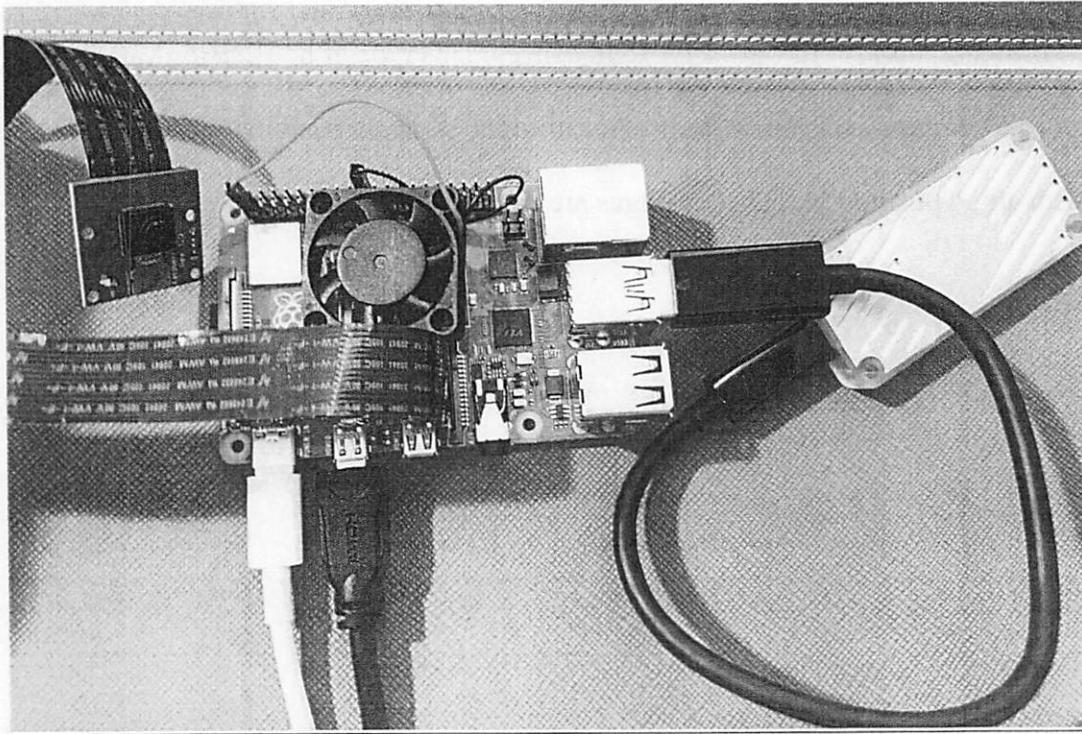


Figure 7: Hardware Layout

In the figure above, it is shown that the Raspberry Pi is connected to power via the 3-amp power supply and the Pi Camera is connected via a ribbon cable. The USB accelerator is connected to the raspberry Pi via a USB-A to USB-C cable. The fan is connected to the 5V and ground pins on the Raspberry Pi and the Raspberry Pi is connected to the monitor via the Micro HDMI to HDMI cable.

Detailed Drawings

In this section, the schematics for the housing will be shown and described. After the housing was designed. It was 3D printed out of PLA. The housing is encasing the Raspberry Pi,

5.3 Hardware Layout/Detailed Drawings/Equipment Selection and Specifications (cont.)

fan, Coral USB, camera, and the camera ribbon cable. The full housing can be seen in the figure below.

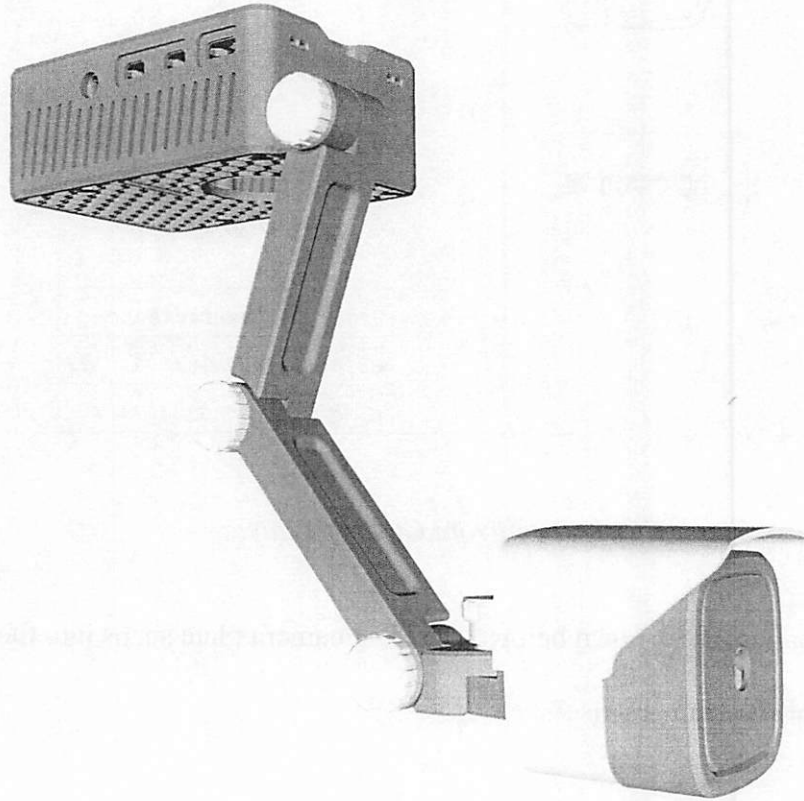


Figure 8: Housing Design

Listed below will be detailed drawings of the individual pieces of the housing. To start with we have the front camera plate. The Pi Camera snaps into place via the 4 pins that stick out on the inside. The lens of the camera sits in the whole in the center.

5.3 Hardware Layout/Detailed Drawings/Equipment Selection and Specifications (cont.)

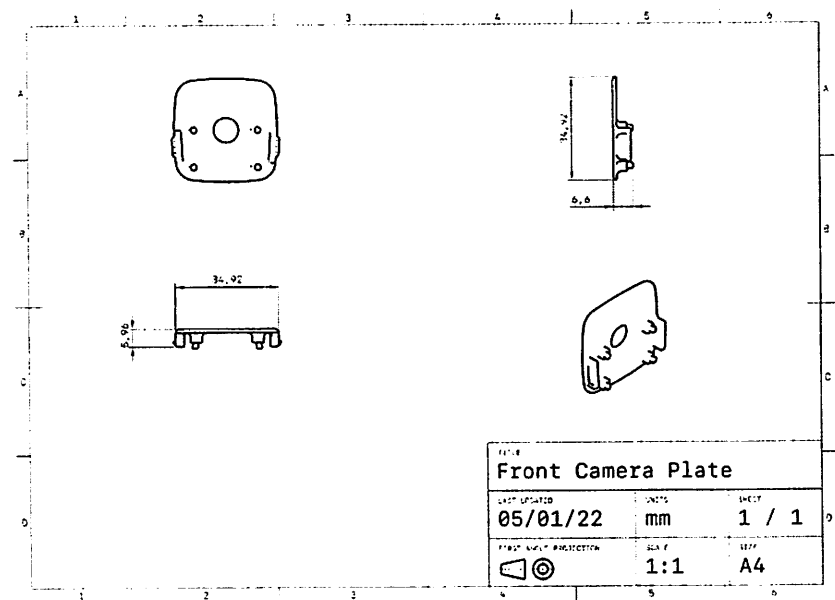


Figure 9.1: Front Camera Plate

The side camera cover is shown below. The front camera plate snaps into the side camera cover and his snaps into the camera visor.

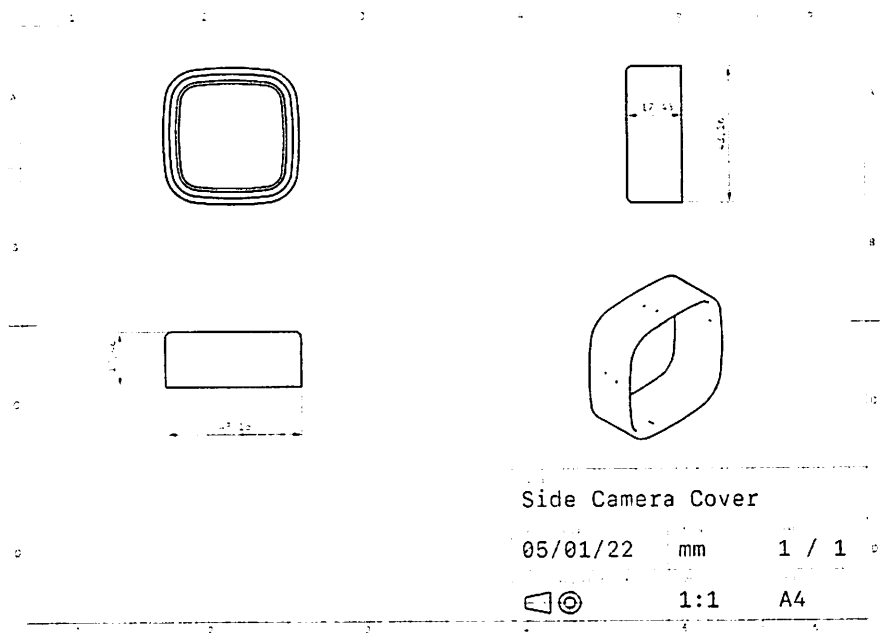


Figure 9.2: Side Camera Cover

5.3 Hardware Layout/Detailed Drawings/Equipment Selection and Specifications (cont.)

The camera visor is shown below. The visor snaps together with the side camera cover. There is a slot on the back to allow the ribbon cable of the camera to pass through as well as a larger slot that connects the back camera plate.

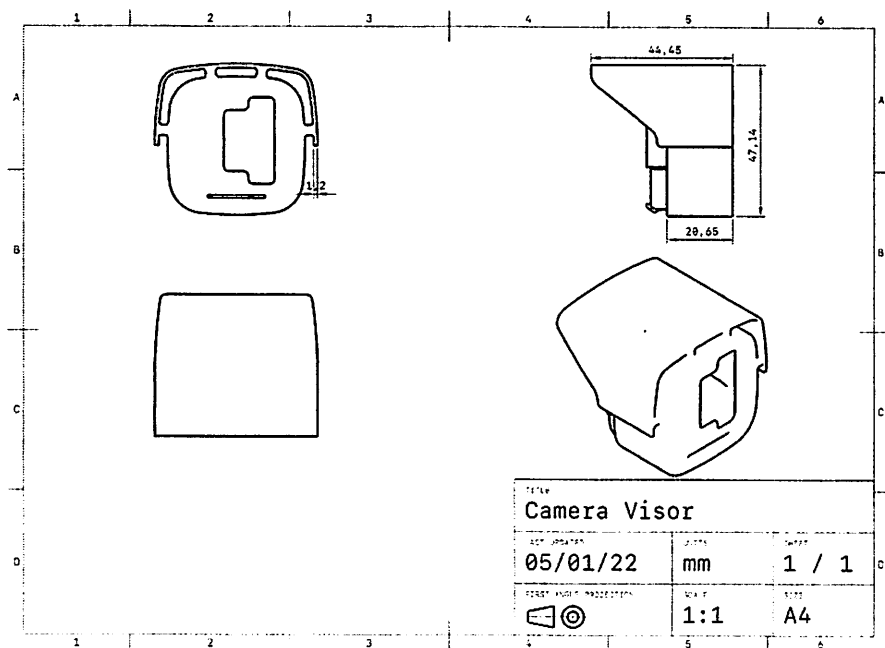


Figure 9.3: Camera Visor

The back camera plate is shown below. The interior side of the back camera plate has a notch that slides into the opening on the camera visor. There is a threaded hole to allow for a short screw attachment to the hinge.

5.3 Hardware Layout/Detailed Drawings/Equipment Selection and Specifications (cont.)

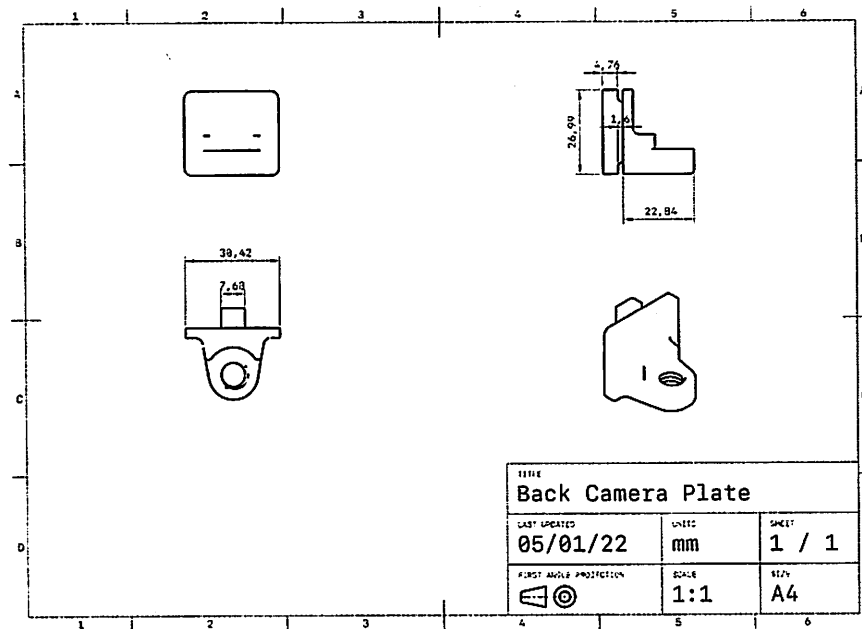


Figure 9.4: Back Camera Plate

The hinge is shown below. The hinge allows for the camera to pan left and right. The hinge is connected to the lower arm via a long screw.

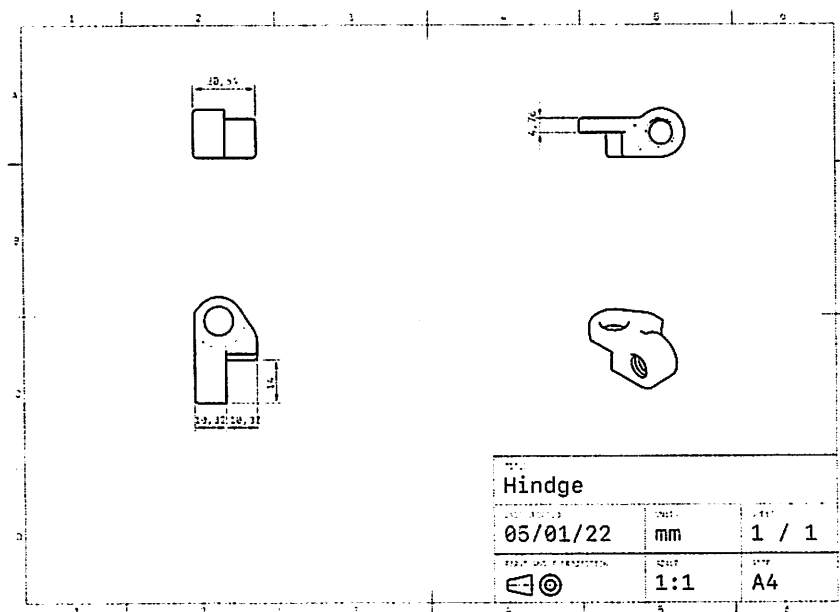


Figure 9.5: Hinge

5.3 Hardware Layout/Detailed Drawings/Equipment Selection and Specifications (cont.)

The lower arm and upper arm are identical and shown below. They are connected via 2 long screws at either end. They have a slit that allows for the ribbon cable to run through the interior of the arm.

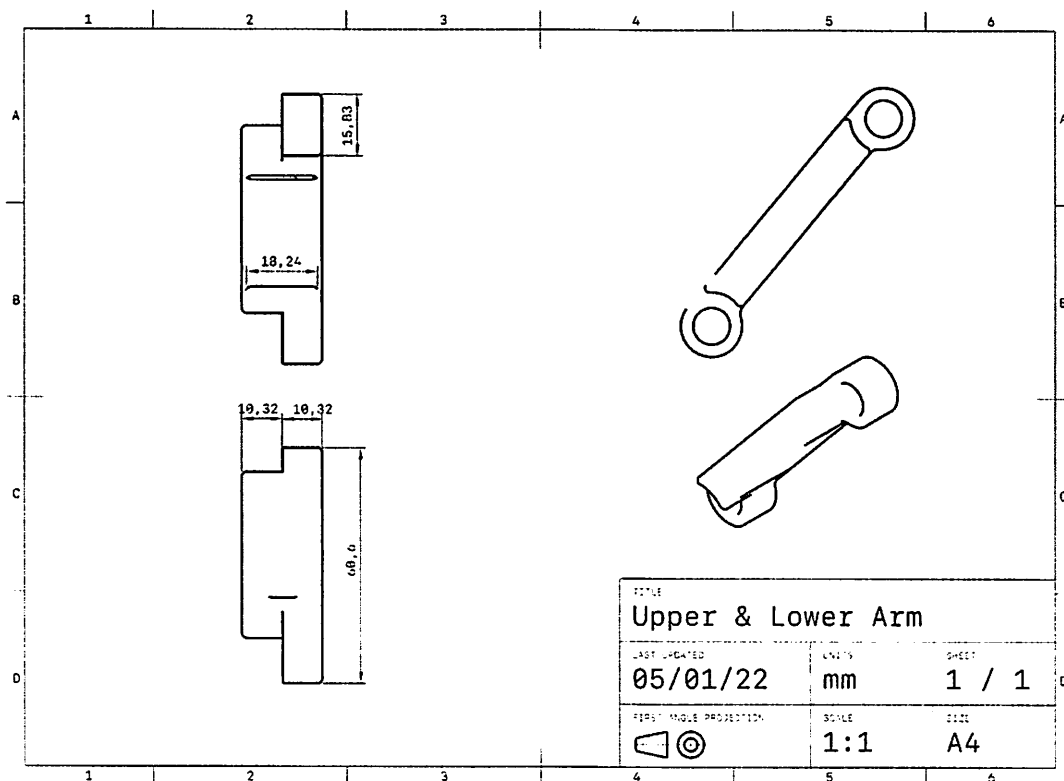


Figure 9.6: Upper & Lower Arm

The long and short screws are threaded the same way. The long screw is shown below. The short screw is $\frac{3}{4}$ the length of the long screw.

5.3 Hardware Layout/Detailed Drawings/Equipment Selection and Specifications (cont.)

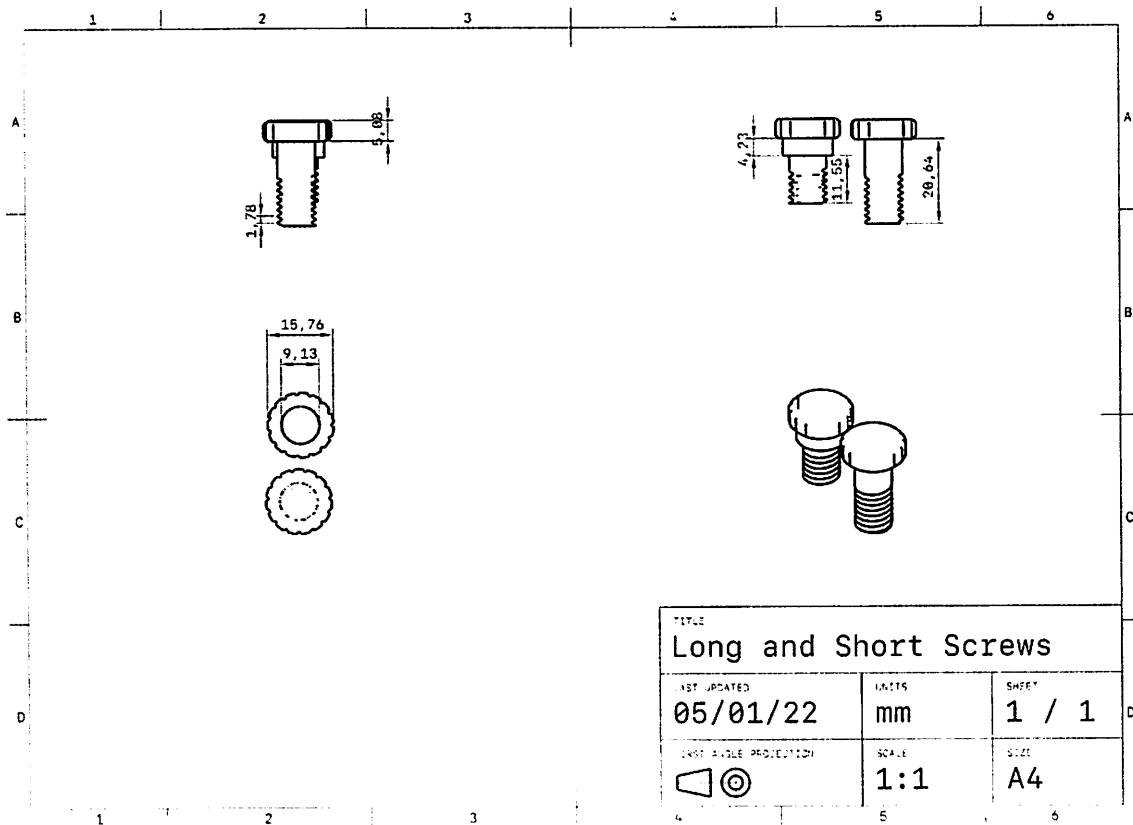


Figure 9.7: Long & Short Screws

The Raspberry Pi case top is shown below. It is the portion of the case that would face downwards if mounted to a ceiling despite being named the top. It has a cut out for the fan to sit on and has enough space to hold the Coral USB Accelerator. It also has a slit cut out for the camera ribbon cable and 3 large notches for the USB Ports and the Ethernet port. The hexagonal shape on the top and slits on the sides allow for adequate air flow as overheating used to be an issue.

5.3 Hardware Layout/Detailed Drawings/Equipment Selection and Specifications (cont.)

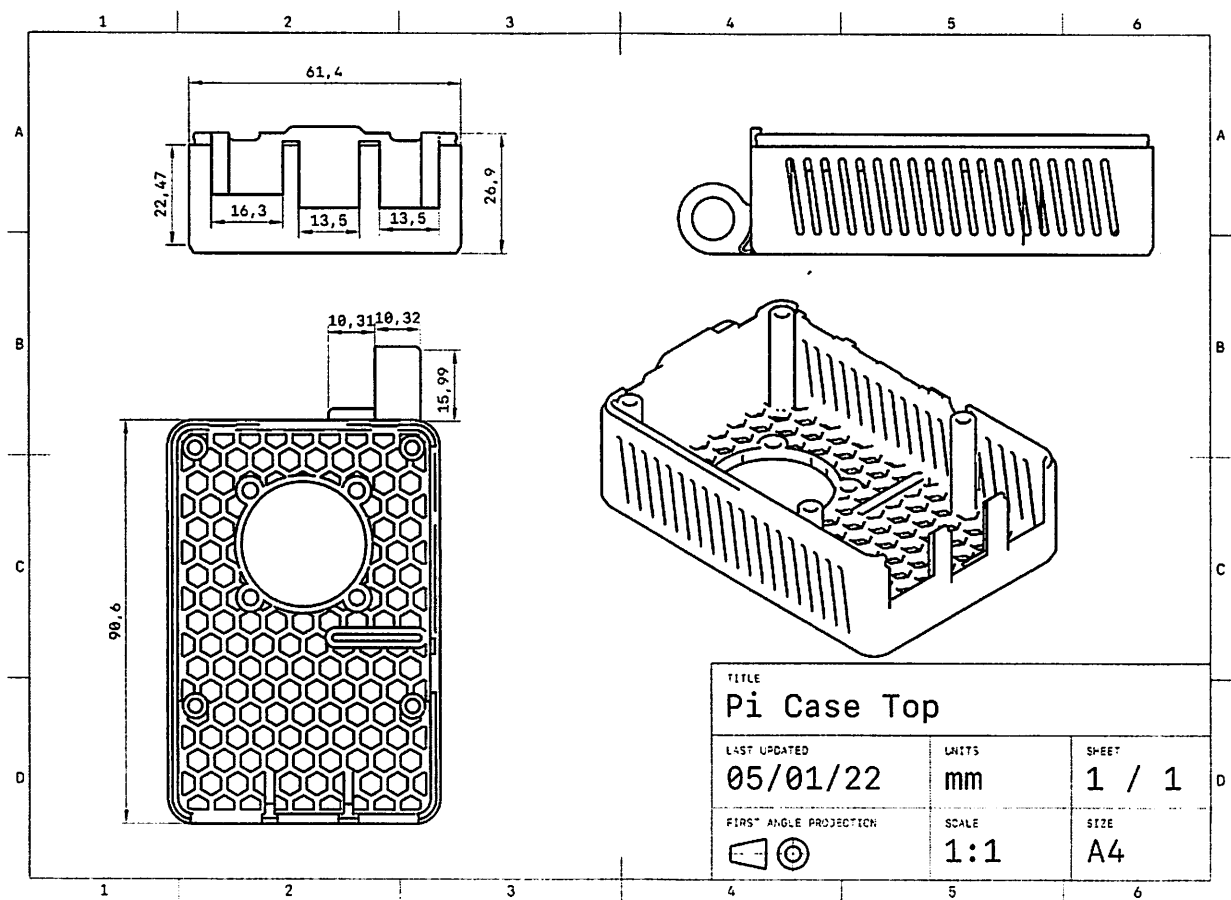


Figure 9.8: Pi Case Top

The Raspberry Pi case bottom can be seen below. It is the portion of the case that would be touching the surface in which the device is being mounted. The case has 4 cut outs for the power, HDMI ports, and the audio port. The two case pieces snap into one another with the Raspberry Pi, fan and Coral USB accelerator fixed in place on the inside.

5.3 Hardware Layout/Detailed Drawings/Equipment Selection and Specifications (cont.)

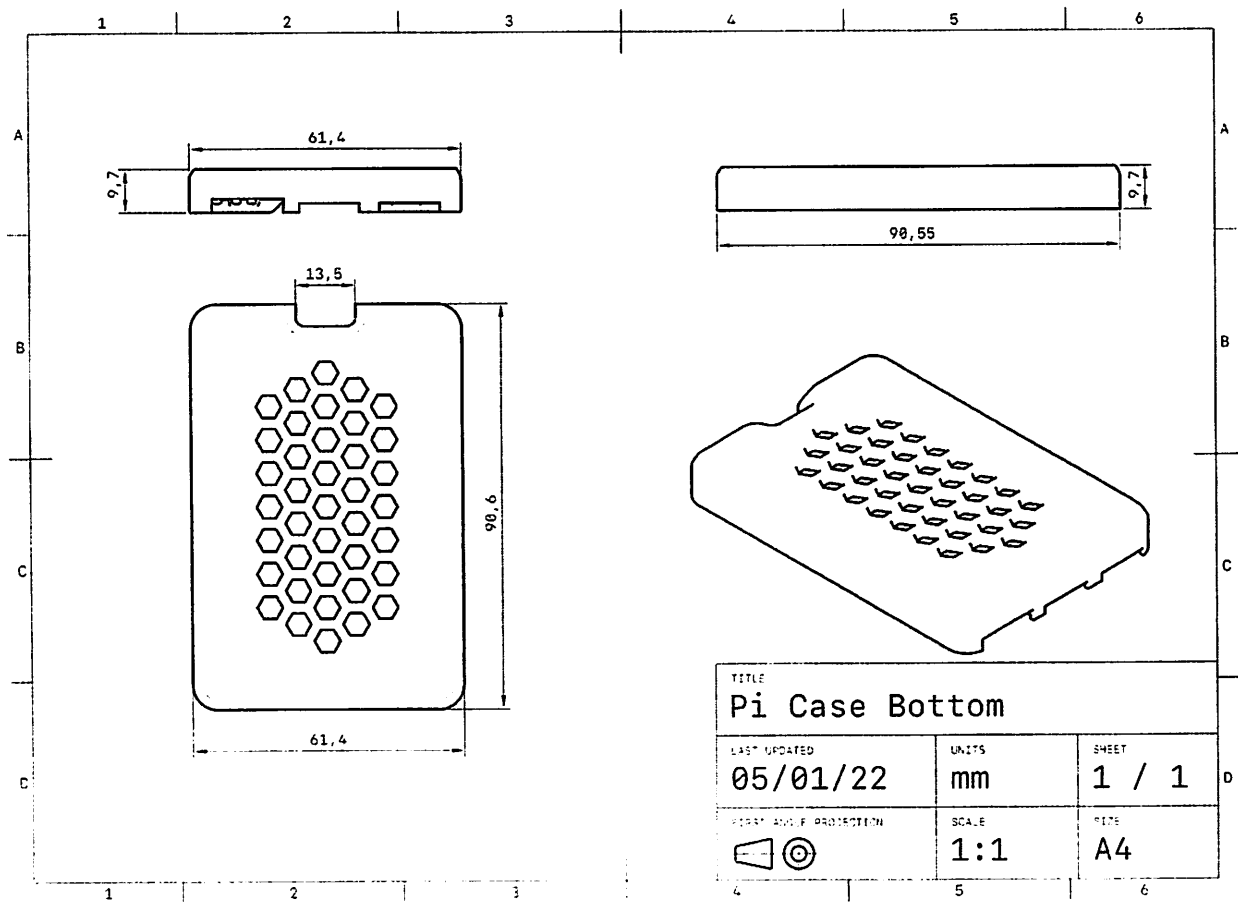


Figure 9.9: Pi Case Bottom

The parts are all 3D printed from PLA including the screws. This allows for the housing to be very lightweight.

5.3 Hardware Layout/Detailed Drawings/Equipment Selection and Specifications (cont.)

Equipment Selection and Specifications

The Raspberry Pi 4 Model B was selected due to it being the most up to date Raspberry Pi to be released. It has a much better performance than the other Raspberry Pi models. A LINPAC benchmark test showed that the Raspberry Pi 4 B almost tripled the score of the Raspberry Pi 3B+. The specifications of the Raspberry Pi 4 Model B are shown in the table below.

Raspberry Pi 4 Model B	
Processor	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Ram	8GB LPDDR4-3200 SDRAM (depending on model)
Wi-Fi/Bluetooth	2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
Ethernet Port	Gigabit Ethernet
USB Ports	2 USB 3.0 ports; 2 USB 2.0 ports.
GPIO	Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
Display Ports	2 × micro-HDMI ports (up to 4kp60 supported) 2-lane MIPI DSI display port
Camera Port	2-lane MIPI CSI camera port
Audio Port	4-pole stereo audio and composite video port
Video Encoder	H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
Graphics	OpenGL ES 3.1, Vulkan 1.0
Storage	Micro-SD card slot for loading operating system and data storage
Power	5V DC via USB-C connector (minimum 3A*) 5V DC via GPIO header (minimum 3A*) Power over Ethernet (PoE) enabled (requires separate PoE HAT)
Operating Temperature	0 – 50 degrees C ambient

Table 3.1: Raspberry Pi 4 Model B Specifications

5.3 Hardware Layout/Detailed Drawings/Equipment Selection and Specifications (cont.)

The Raspberry Pi Camera was selected due to its easy compatibility with the Raspberry Pi 4 Model B. The specifications of the Raspberry Pi Camera are shown below.

Raspberry Pi Camera Rev 1.3	
Still Resolution	5 Megapixels
Video Modes	1080p30, 720p60 and 640 × 480p60/90
Linux Integration	1080p30, 720p60 and 640 × 480p60/90
C Programming API	OpenMAX IL and others available
Sensor	OmniVision OV5647
Sensor Resolution	2592 × 1944 pixels
Sensor Image Area	3.76 × 2.74 mm
Pixel Size	1.4 μm × 1.4 μm
Optical Size	1/4"
Full-frame SLR Lens Equivalent	35 mm
S/N Ratio	36 dB
Dynamic Range	67 dB @ 8x gain
Sensitivity	680 mV/lux-sec
Dark Current	16 mV/sec @ 60 C
Well Capacity	4.3 Ke-
Depth of Field	approx. 1 m to infinity
Focal Length	3.60 mm +/- 0.01
Horizontal Field of View	53.50 +/- 0.13 degrees
Vertical Field of View	41.41 +/- 0.11 degrees
Focal Ratio (F-Stop)	2.9

Table 3.2: Raspberry Pi Camera Specifications

5.3 Hardware Layout/Detailed Drawings/Equipment Selection and Specifications (cont.)

The Google Coral USB Accelerator was selected due to its easy compatibility with the Raspberry Pi 4 Model B. This device aided in increasing the frame rate by approximately 20 frames per second. The specifications of the Google Coral USB Accelerator are shown below.

Google Coral USB Accelerator	
ML accelerator	Google Edge TPU coprocessor: 4 TOPS (int8); 2 TOPS per watt
Connector	USB 3.0 Type-C* (data/power)
Dimensions	65 mm x 30 mm

Table 3.3: Coral USB Accelerator Specifications

The Raspberry Pi 4 Official Power Supply Unit was selected due to it being built specifically to meet the requirements for the Raspberry pi 4 Model B. The specifications of the Raspberry Pi 4 Official Power Supply Unit are shown below.

Raspberry Pi 4 Official Power Supply Unit	
Power	15.3-Watt output
Voltage	5.1-Volt output
Current	3.0-Amp output
Length	1.5-m cable

Table 3.4: Power Supply Specifications

The QVS Micro-HDMI to HDMI Cable was selected because it fit the Raspberry Pi 4 B and the monitor we used. The specifications of the cable are shown below.

QVS Micro-HDMI to HDMI Cable	
Supports	15.3-Watt output
Length	2-m cable
Connectors	HDMI Male to Micro HDMI Male

Table 3.5: Micro-HDMI to HDMI Cable Specifications

5.3 Hardware Layout/Detailed Drawings/Equipment Selection and Specifications (cont.)

The Samsung 32GB EVO Plus UHS-I microSDHC Memory Card was selected due to it having enough storage capacity and a fast enough speed for the project. The specifications of the Raspberry Pi 4 Official Power Supply Unit are shown below.

Samsung 32GB EVO Plus UHS-I microSDHC Memory Card	
Card Type	microSDHC
Storage Capacity	32 GB
Bus Type	UHS-I
Speed Class	10
UHS Speed Class	U1
Read Speed	Maximum: 95 MB/s
Write Speed	Maximum: 20 MB/s Minimum: 10 MB/s
Durability	Magnet-Proof, Temperature Extremes, Waterproof, X-Ray Proof
Operating Temperature	-13 to 185 F / -25 to 85 °C
Storage Temperature	-40 to 185 F / -40 to 85 °C
Wireless Capability	None

Table 3.6: MicroSD Card Specifications

5.3 Hardware Layout/Detailed Drawings/Equipment Selection and Specifications (cont.)

The Pi-FAN was selected due to its easy compatibility with the Raspberry Pi. The specifications of the Pi-FAN are shown below.

Pi-FAN	
Operating Voltage	5 Volts
Current	0.2 Amps
Motor	Brushless DC fan
Dimensions	30mm x 30mm x 8mm
Wire Length	3.25" / 80mm
Weight	6.2g / 0.22oz

Table 3.7: Pi-FAN Specifications

The OVERTURE Polylactic (PLA) ACID Filament was selected due to its easy compatibility with the 3D printer that was used. It was also ideal for its biodegradability and was already readily available. The specifications of the OVERTURE PLA Filament are shown below.

OVERTURE Polylactic ACID Filament	
Diameter	1.75 mm diameter
Dimensional Accuracy	Accuracy +/- 0.03 mm
Material	Polylactic Acid (PLA)
Extrusion Temperature	190°C - 220°C (374°F~ 428 °F)
Heated Bed Temperature	25°C - 60°C (77°F~ 140°F)
Color	Black

Table 3.8: Polylactic Acid Specifications

5.4 Overview of Code

There are three main portions of the code. The first section handles training the model to detect weapons. The second section contains everything that deals with the process of running the camera and detecting objects. The third section handles the logic behind how and when to send the notification. The main portions of these three sections will be explained below; however, to see the full-length code, see appendix page A2

Training the Model

When training the model, we used over 5000 images that we labeled as a specific category of weapon. These images became our dataset. The model is then trained on this data set by running the training executable. This code, along with instructions on how to configure the training job can be found in the appendix on page A8. This section calls many different files to aid in the training process. To summarize what is happening during training, the algorithm takes an image that looks at where the bounding box for the weapon is and looks for certain patterns and similarities throughout the various images.

Running the Camera

The code that runs the camera and weapon detection starts with the camera constraints. The dimensions of the camera must be defined because the bounding box location for the weapons is calculated using the image height and width. Below it is shown that the image width for our camera is 1280 and the height is 720.

```
IM_WIDTH = 1280
IM_HEIGHT = 720
```

Figure 10.1: Camera Dimensions

5.4 Overview of Code (continued)

A label map is a list of all the objects, or classes, we want to detect. In our case, it is just weapons, so we have a label map with rifle, handgun, blade, and blunt object. We call this label map file and define the number of classes. The figure below shows the code that performs these actions.

```
# Path to label map file
PATH_TO_LABELS = os.path.join(CWD_PATH, 'training', 'labelmap.pbtxt')

# Number of classes the object detector can identify
NUM_CLASSES = 4

## Load the label map.
# Label maps map indices to category names, so that when our convolution
# network predicts '5', we know that this corresponds to 'king'.
# Here we use internal utility functions, but anything that returns a
# dictionary mapping integers to appropriate string labels would be fine
label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
categories = label_map_util.convert_label_map_to_categories(label_map,
max_num_classes=NUM_CLASSES, use_display_name=True)
category_index = label_map_util.create_category_index(categories)
```

Figure 10.2: Loading Label Map

The program then loads the TensorFlow model into memory. This is the model that was trained to detect weapons. This can be seen in the figure below.

```
# Load the TensorFlow model into memory.
detection_graph = tf.Graph()
with detection_graph.as_default():
    ops = tf.get_default_ops()
    with tf.Session(graph=detection_graph):
        # Load frozen computation graph
        serialized_graph = tf.gfile.GFile(PATH_TO_CKPT, 'frozen_inference_graph.pb',
        mode='rb').read()
        detection_graph_def = tf.GraphDef()
        detection_graph_def.ParseFromString(serialized_graph)
        tf.import_graph_def(detection_graph_def, name='')

# Create a session for the whole detection_graph
sess = tf.Session(graph=detection_graph)
```

Figure 10.3: Loading in the Model

5.4 Overview of Code (continued)

The output of the model includes the detection boxes, scores, classes, and number of objects detected. In the figure below, these are seen as `detection_boxes`, `detection_scores`, `detection_classes`, and `num_detections`. The detection boxes give the coordinates of the top left and bottom right points of the bounding box. The score is the confidence level this is displayed as a percentage. The classes are the names of the object being detected and the number of detections keeps track of how many of the detected objects are in the view of the camera.

```
# Output tensors are the detection boxes, scores, and classes
# Each box represents a part of the image where a particular object was detected
detection_boxes = detection_graph.get_tensor_by_name('detection_boxes:0')

# Each score represents level of confidence for each of the objects.
# The score is shown on the result image, together with the class label.
detection_scores = detection_graph.get_tensor_by_name('detection_scores:0')
detection_classes = detection_graph.get_tensor_by_name('detection_classes:0')

# Number of objects detected
num_detections = detection_graph.get_tensor_by_name('num_detections:0')
```

Figure 10.4: Boxes, Scores, Classes, and Number of Boxes

The live value of the frame rate is calculated by checking the tick frequency. This is shown in the figure below. It is important to have the exact frame rate of the camera as this directly affects the speed at which the notification is sent.

```
# initialize frame rate calculation
start = time.time()
while True:
    img = cv2.imread('img.jpg')
    end = time.time()
```

Figure 10.5: Frame Rate Calculation

5.4 Overview of Code (continued)

The model is run and uses each frame from the camera as its input. This code is shown below.

```
# Perform the actual detection by running the model with the image as input
boxes scores classes num sess
detection_boxes detection_scores detection_classes num_detections
feed_dict image_tensor frame_expanded
```

Figure 10.6: Running Detection Model

The results from the detection model are displayed over the actual live footage from the camera. This is done by drawing the boxes, showing the classes, and updating the scores for each frame. The figure below demonstrates this process.

```
# Draw the results of the detection (aka 'visulaize the results')
vis_util
    frame
    np boxes
    np classes np
    np scores
    category_index
    use_normalized_coordinates True
    line_thickness 4
    min_score_thresh
```

Figure 10.7: Visualization of Results

Notification System

For our notification system, we used Twilio. To code in the Twilio information we first import the Twilio client. Then the account SID number, authorization number, the phone number of who needs to receive the notification, and the number the message is being sent from are all defined. In the figure below, it is shown where these values are defined as the specific variables: account_sid, auth_token, my_number, and twilio_number, respectively. The numbers are covered for privacy purposes.

5.4 Overview of Code (continued)

```
# Set up Twilio
from twilio.rest import Client

# Twilio SID, authentication token, my phone number, and the Twilio phone number
account_sid = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
auth_token = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
my_number = 'XXXXXXXXXXXXXXXX'
twilio_number = '19362510045'

client = Client(account_sid, auth_token)
```

Figure 11.1: Twilio Setup

The way the program knows when to send the notification is a series of “if” statements. It determines if a weapon is on the screen. The weapons are defined by their ID classes; 1 for rifle, 2 for handgun, 3 for knife, 4 for blunt object. A predetermined area of detection is defined if it is not wanted to detect weapons in the camera’s full field of view. If one of the weapons is located within that area for 300 frames, it sends the message “A weapon has been detected on your camera” to the numbers previously defined. The program then pauses for a set amount of time to prevent messages being spammed. This can all be seen in the snapshot of the code below.

```
if int(classes[0][0]) == 1 or int(classes[0][0]) == 2 or
int(classes[0][0]) == 3 or int(classes[0][0]) == 4 and pause == 0:
    x = int(boxes[0][0])
    y = int(boxes[0][1])
    # Draw a circle at center of object
    cv2.circle(frame, (x, y), 5, (255, 255, 255), 1)
    # If object is in inside box, increment inside counter variable
    if x > TL_inside and x < BR_inside and y > TL_inside and
y < BR_inside:
        inside_counter = inside_counter + 1

if inside_counter == 300:
    detected_inside = True
    message = client.messages.create(
        body="A weapon has been detected on your camera",
        from_=twilio_number,
        to=my_number)

    inside_counter = 0
    outside_counter = 0
    # Pause weapon detection by setting "pause" flag
    pause = 1
```

Figure 11.2: SMS Notification Execution

5.5 Application of Engineering Codes and Standards used in Final Design ✓

The following IEEE Codes and Standards were used in the final design of our project:

IEEE 3652.1-2020: Federated Machine Learning - Our machine learning model can be constructed from data distributed across various repositories owned by various companies

IEEE P2671: Online detection with machine vision - This standard specifies through the general requirements of online detection based on machine vision

IEEE P2247.3: Recommended Practices for Evaluation of Adaptive Instructional Systems - best practice incorporates and promotes the principles of ethically aligned design for the use of artificial intelligence

IEEE 829-2008: IEEE Standard for Software and System Test Documentation - We used this standard when testing our software for detecting weapons

IEEE 2755.2-2020: Recommended Practice for Implementation and Management Methodology for software Based Intelligent Process Automation

IEEE 7.8: IEEE: Code of Ethics

- 3) To be honest and realistic in stating claims or estimates based on available data
- 5) To improve the understanding of technology, its appropriate application, and potential consequences

5.6 Economic Analysis

The economic analysis of our project can be separated into two phases. There were different components that were used in each phase which we will outline below. During phase one we were given the parts that we needed to start a preliminary design of our project. The parts that we used in our phase one design were given to us by the school and can be seen in table 4.1.

Item Name	Quantity	Total
Raspberry Pi 3b+	1	Pre-owned
Raspberry Pi Camera Rev 1.3	1	Pre-owned
	Total:	\$0

Table 4.1: Phase 1 Cost

After completing the testing of phase one with the given parts, we were able to identify which parts needed updating and investigate ways that we can improve our design. We needed to update our Raspberry Pi 3b+ to the Raspberry Pi 4b which is a more powerful and faster model, we also upgraded our MicroSD card so that our Raspberry Pi can hold more storage. Another upgrade to our project was the addition of the Google Coral Accelerator, which increases the frame rate of our camera and video processing. To connect the Raspberry Pi 4b to a monitor and observe an area we had to purchase a Micro HDMI to HDMI cord. The updated cost of our project can be seen in table 4.2.

Item Name	Quantity	Total
Raspberry Pi 4b	1	\$134
Raspberry Pi Camera Rev 1.3	1	Pre-owned
Google Coral USB Accelerator	1	\$174.58
Micro HDMI to HDMI cord	1	\$16.95
Micro SD card	1	\$20
	Total:	\$345.53

Table 4.2: Smart Threat Detection System Cost

5.7 Final Design - Show Pictures or Other Examples of Working Design

Figures 12.1 - 12.3 depict a rifle being detected by our camera. The yellow box and word “rifle” are specific to that class of weapon. In various orientations, including partially covered, the rifle is detected with remarkably high confidence.

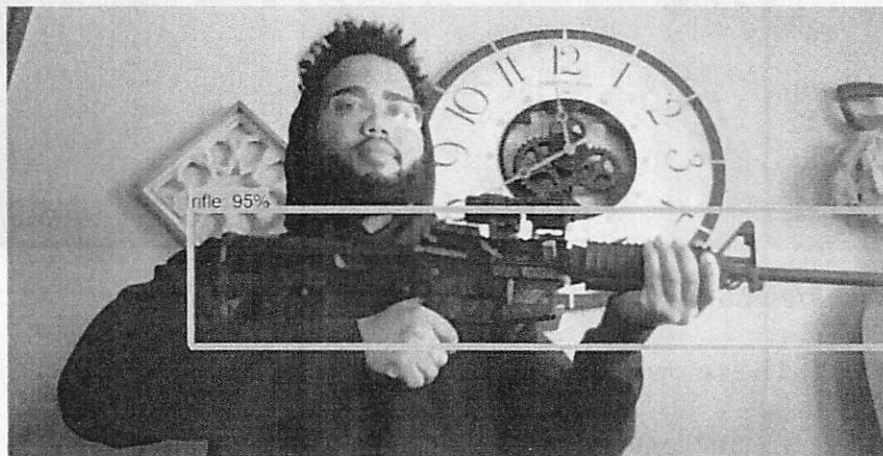


Figure 12.1: Rifle Detected with 95% Confidence



Figure 12.2: Rifle Detected with 100% Confidence

5.7 Final Design - Show Pictures or Other Examples of Working Design

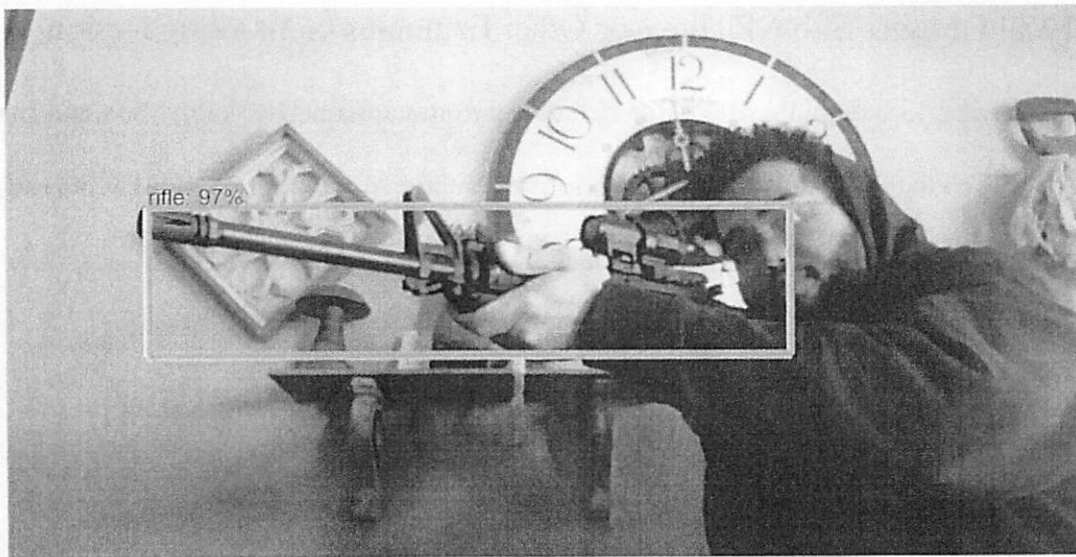


Figure 12.3: Covered Rifle Detected with 97% Confidence

Figures 13.1 - 13.5 below depict a handgun being detected by our camera. The green box and word "handgun" are specific to that class of weapon. In various orientations, including the barrel pointed towards the camera and a black gun on a black background, the handgun is detected with high confidence.

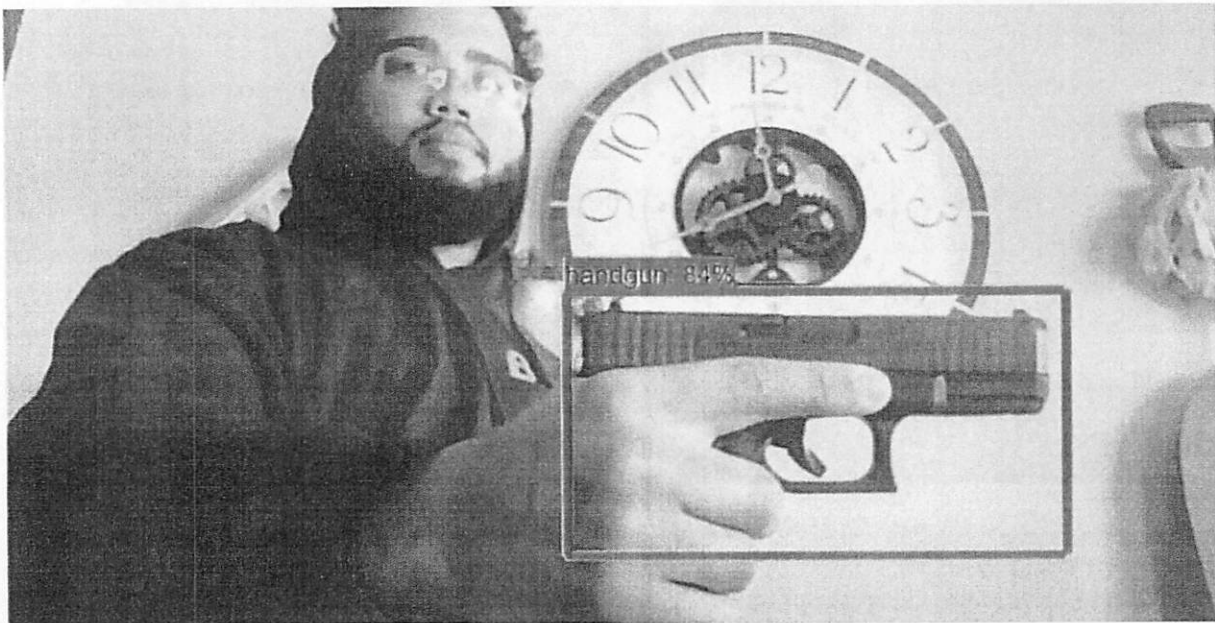


Figure 13.1: Handgun Detected with 84% Confidence

5.7 Final Design - Show Pictures or Other Examples of Working Design

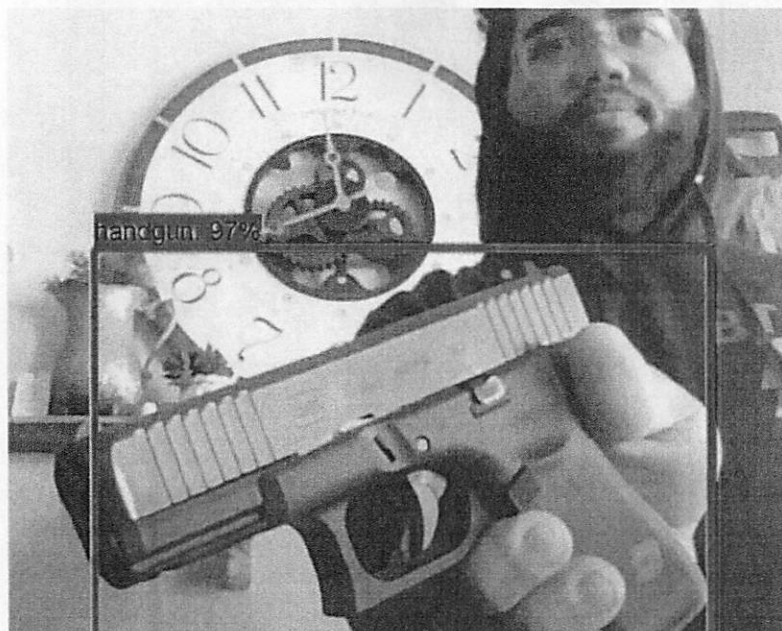


Figure 13.2: Handgun Detected with 97% Confidence

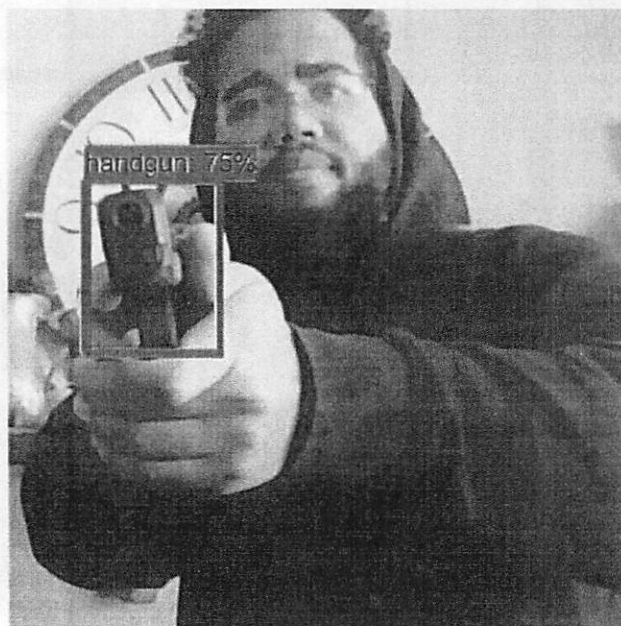


Figure 13.3: Barrel Pointed at Camera Detected with 75% Confidence

5.7 Final Design - Show Pictures or Other Examples of Working Design

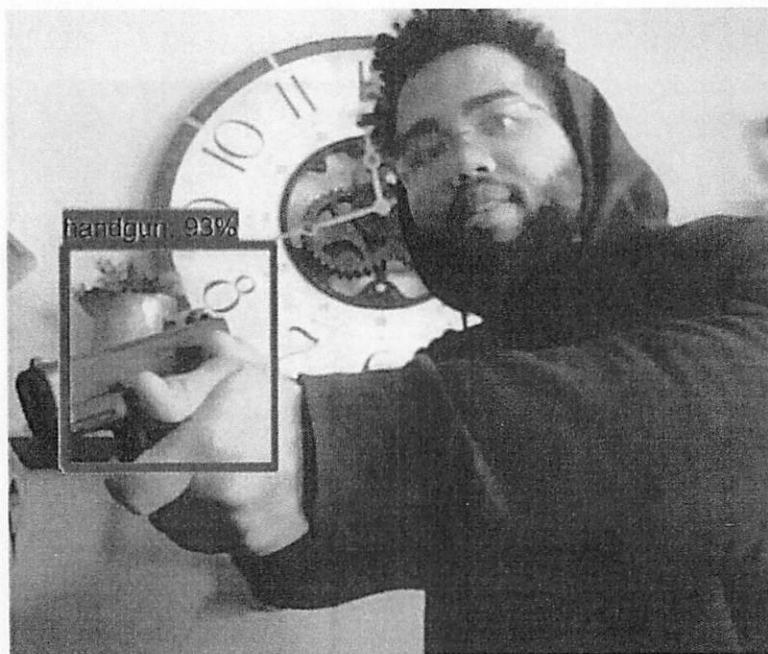


Figure 13.4: Partially Covered Handgun Detected with 93% Confidence

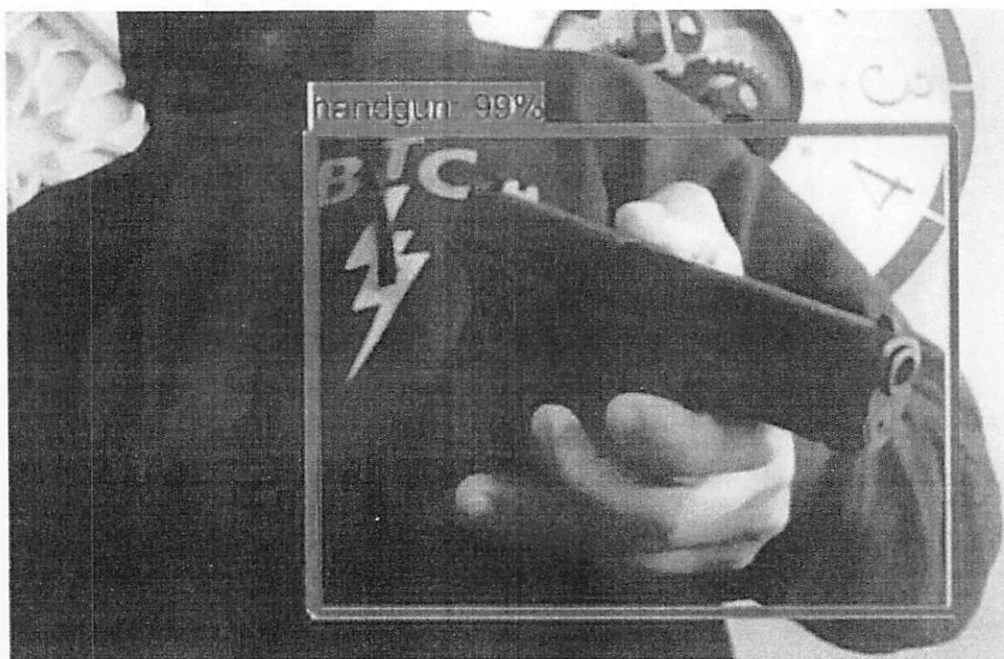


Figure 13.5: Black Handgun Detected on Black Jacket with 99% Confidence

5.7 Final Design - Show Pictures or Other Examples of Working Design

Figures 14.1 - 14.3 below depict a blade being detected by our camera. The greenish yellow box and word “blade” are specific to that class of weapon. The blade is detected with high confidence. However, the algorithm is less than 60% confident in categorizing the blade when its narrow profile is facing the camera. Objects with a confidence level less than 60% are not displayed on the screen.

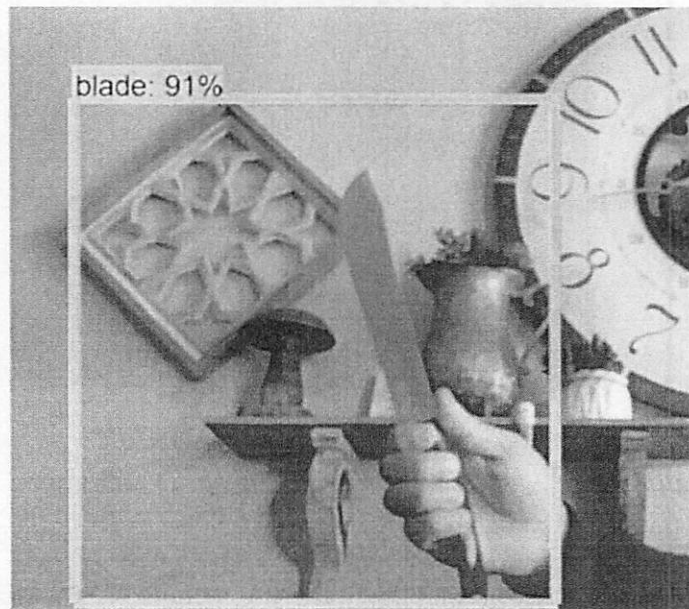


Figure 14.1: Blade Detected with 91% Confidence

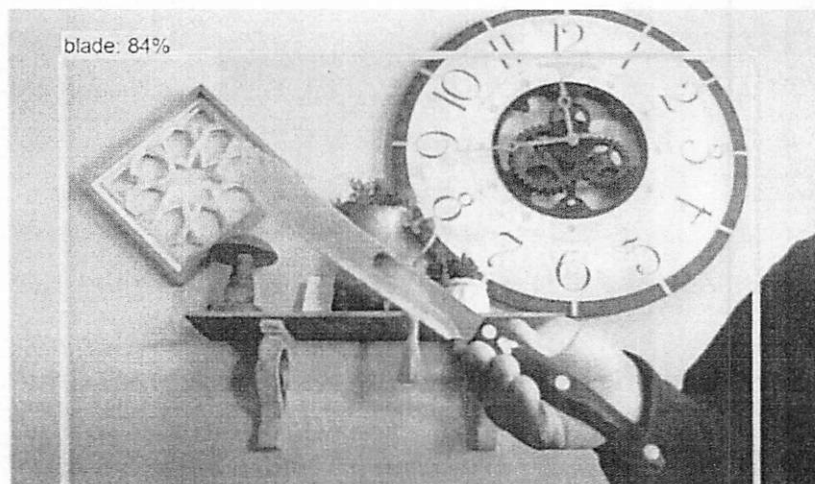


Figure 14.2: Blade Detected with 84% Confidence

5.7 Final Design - Show Pictures or Other Examples of Working Design



Figure 14.3: Narrow Profile of Blade not Detected

Figure 15 shows an example of what the notification could read when a weapon is detected. The “Sent from your Twilio trial account” portion does not appear when the small messaging fee is paid.

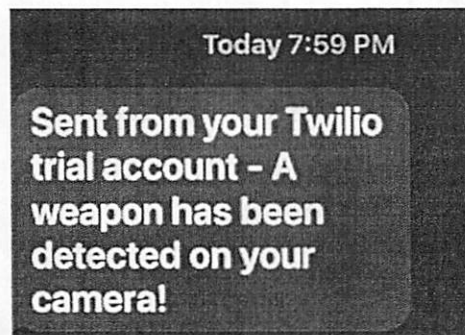


Figure 15: Example Notification

5.7 Final Design - Show Pictures or Other Examples of Working Design

Figures 16.1 & 16.2 demonstrate how the algorithm does not accidentally depict a similar looking object as a weapon. In this case, the phone is being held in a way a handgun would be held and no box is ever displayed. This shows that there are very few false positives.

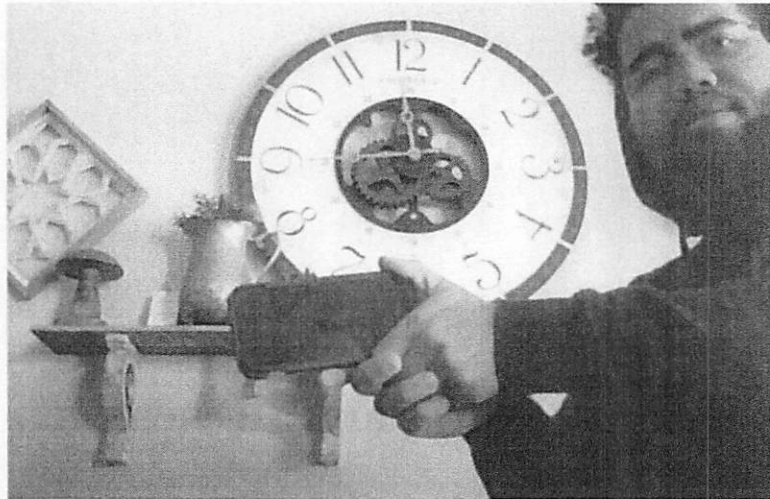


Figure 16.1: Cell Phone False-Positive Test (part 1)



Figure 16.2: Cell Phone False-Positive Test (part 2)

Section 6 - Results and Discussion

6.1 Summary of Goals and Objectives

The main objective was to implement object detection onto Raspberry Pi. We ran a mobile deep learning model, SSD-MobileNet, on Raspberry Pi to detect various objects. Our experimental results demonstrated the effectiveness of this implementation. Moreover, the ongoing work was to extend this implementation to weapon detection, which was used to detect different weapons in a real-time manner. Upon completing and validating this extension in different testing scenarios our results were successful. In order to complete the smart security system a notification system was integrated into the object detector once the new categories of objects were trained to be detected. This ensured that once a weapon was detected in the frame a text notification was sent to the appropriate authorities warning them of the potential threat.

6.2 Summary of Constraints and Codes met by Design

The main constraints we have observed with this project are creating a durable housing kit for the raspberry pi with a mountable camera that is capable of staying in place for an extended duration of time. The next constraint was the camera resolution impacting the accuracy of the weapon detection, as well as the distance that the camera is able to view, to fix this having a camera that has a minimum resolution of 1080p will ensure a clear quality to have the highest accuracy. The final notable constraint with our project was the frame rate of the real time tracking with the raspberry pi. Due to the severity of this issue we needed to increase the frame rate of the hardware, and after thorough research on how to solve the issue, the best solution to the situation was changing our software to TFlite and quantizing the data as well as utilizing a USB accelerator device in order to aid in speeding up the frame rates of the overall system.

The following codes were used in our final design:

IEEE 3652.1-2020: Federated Machine Learning - Our machine learning model can be constructed from data distributed across various repositories owned by various companies

IEEE P2671: Online detection with machine vision - This standard specifies through the general requirements of online detection based on machine vision

IEEE P2247.3: Recommended Practices for Evaluation of Adaptive Instructional Systems - best practice incorporates and promotes the principles of ethically aligned design for the use of artificial intelligence

IEEE 829-2008: IEEE Standard for Software and System Test Documentation - We used this standard when testing our software for detecting weapons

IEEE 2755.2-2020: Recommended Practice for Implementation and Management Methodology

6.2 Summary of Constraints and Codes met by Design

for software Based Intelligent Process Automation



IEEE 7.8: IEEE: Code of Ethics

- 3) To be honest and realistic in stating claims or estimates based on available data
- 5) To improve the understanding of technology, its appropriate application, and potential consequences

6.3 Discussion of Results and Other Use Cases

This project allowed us to design a completely unique security system by utilizing a machine learning algorithm capable of detecting weapons and immediately notifying the owner of the potential danger that is within the area. Designing this system granted us the opportunity to offer a more advanced security system to consumers at a lower cost compared to competitors. Machine learning has not been introduced to security systems easily and with the introduction of our device, we would be offering a continuously learning security system that will offer greater results as time goes on while also being capable to notify the owners of the home or establishment of a potential threat.

The use of this system can be expanded into being capable of finding concealed weapons with the implementation of an inferred camera, with this feature the number of areas this system would be able to be set up will drastically increase due to the uncertainty of if there is a weapon on a person. Taking into consideration that weapon detection is a specialization of object detection, the other uses of object detection can be implemented into several types of trackers, such as a people tracker, a vehicle tracker, or even an animal tracker. These would also easily be able to be used to notify when any object, not just a weapon, is detected in a certain area.

Section 7 - Conclusions and Recommendations

Section 7 - Conclusions and Recommendations (continued)

It is observed that a large population reconciles violence related to various weapons such as guns and knives all over the world. Therefore, it is a good proactive solution to detect these weapons in public areas to defend against this violence. Our Weapon Detection on Raspberry Pi design identifies active threats by revealing displayed weapons. These weapons can be either a rifle, a handgun, or a blade and the weapon must be in view of the camera. Our alert notification system provides additional time for actions to be taken to minimize casualties.

We recommend having a mountable dry surface for the Raspberry Pi platform. Wi-Fi and or data service are needed for text notifications. Electricity to run power to the Raspberry Pi system is also required. In the event of power failure, the possibility of having a backup generator is convenient to prevent system interruption. This artificially intelligent threat detection system will impact many from universities and church organizations to businesses and homeowners.

List of References

List of References (continued)

- Best Home Security System Company & Security Cameras,” Deep Sentinel, 24-Sep-2021. [Online]. Available: <https://www.deepsentinel.com/>. [Accessed: 04-Oct-2021].
- “Burglary Statistics: The Hard Numbers,” Alarms.org, 19-Dec-2019. [Online]. Available: <https://www.alarms.org/burglary-statistics/>. [Accessed: 04-Oct-2021].
- “Frequently Asked Questions About Small Businesses,” U.S. Small Business Administration Office of Advocacy, Oct-2020. [Online]. Available: <https://cdn.advocacy.sba.gov/wp-content/uploads/2020/11/05122043/Small-Business-FAQ-2020.pdf>. [Accessed: 04-Oct-2021].
- “Home Security Systems,” SimpliSafe, 2021. [Online]. Available: <https://simplisafe.com/>. [Accessed: 04-Oct-2021].
- “Introducing VEDX: GPT-3 For Vision,” Visual one. [Online]. Available: <https://www.visualone.tech/index.html#python>. [Accessed: 04-Oct-2021].
- J. Arora, “Theft Detection Using Machine Learning,” Medium, 25-Sep-2020. [Online]. Available: <https://medium.com/analytics-vidhya/theft-detection-using-machine-learning-a4232ea51f1c>. [Accessed: 04-Oct-2021].
- L. Blain, “Vaak’s AI theft-detection system is already 81% accurate,” New Atlas, 11-Apr-2019. [Online]. Available: <https://newatlas.com/vaak-vaakeye-ai-theft-detection/59263/>. [Accessed: 04-Oct-2021].
- “More security with security cameras systems,” GW Security. [Online]. Available: <https://www.gwsecurityusa.com/>. [Accessed: 01-Oct-2021].
- “Product Help,” SimpliSafe. [Online]. Available: <https://support.simplisafe.com/hc/en-us>. [Accessed: 04-Oct-2021].
- “Weapons Offenses and Offenders”, U.S. Department of Justice. [Online]. Available: <https://bjs.ojp.gov/content/pub/pdf/woofccj.pdf> [Accessed: 29-Mar-2022]
- S. Catalano, “National Crime Victimization Survey: Victimization During Household Burglary,” Office of Justice Programs, 28-Sep-2010. [Online]. Available: <https://bjs.ojp.gov/content/pub/ascii/vdhh.txt>. [Accessed: 04-Oct-2021].
- S. Cho and N. Tin, “Using infrared imaging technology for concealed weapons detection and visualization,” TENCON 2010 - 2010 IEEE Region 10 Conference, 2010, pp. 228-233, doi: 10.1109/TENCON.2010.5685995.
- S. K. Bandyopadhyay, B. Datta, and S. Roy, “Identifications of Concealed Weapon in a Human Body,” Arxiv. [Online]. Available: <https://arxiv.org/ftp/arxiv/papers/1210/1210.5653.pdf>. [Accessed: 04-Oct-2021].
- S. Perkins, “Heating up the search for Hidden Weapons,” Science News for Students, 03-Dec-2019. [Online]. Available: <https://www.sciencenewsforstudents.org/blog/eureka->

List of References (continued)

lab/heating-search-hidden-weapons.

S. Stasha, "18+ Intriguing Home Ownership Statistics for 2021," Policy Advice, 13-Sep-2021. [Online]. Available: <https://policyadvice.net /insurance/insights/home-ownership-statistics/>. [Accessed: 04-Oct-2021].

"Self-installed DIY Home Security Systems," Cove Security, 04-Oct-2021. [Online]. Available: <https://www.covesmart.com/>. [Accessed: 04-Oct-2021].

Statista Research Department, "Number of robberies in the United States in 2019, by weapon used," Statista, 01-Oct-2020. [Online]. Available: <https://www.statista.com/statistics/251914/number-of-robberies-in-the-us-by-weapon/>. [Accessed: 04-Oct-2021].

T. Covington, "Burglary Statistics, Research, and Facts," The Zebra, 08-Jul-2021. [Online]. Available: <https://www.thezebra.com/resources/research/burglary-statistics/>. [Accessed: 04-Oct-2021].

Uniform Crime Reporting Program, "Crime in the U.S. 2019 Table 23," FBI: UCR, 29-Aug-2019.[Online]. Available: <https://ucr.fbi.gov/crime-in-the-u.s/2019/crime-in-the-u.s.-2019/tables/table-23>. [Accessed: 27-Sep-2021].

Appendix

Code to Run Camera and Send Notification:

```
##### Webcam Object Detection Using TensorFlow-trained Classifier #####
#
# Description:
# This program uses a TensorFlow-trained classifier to perform object detection.
# It loads the classifier and uses it to perform object detection on a webcam
# feed.
# It draws boxes, scores, and labels around the objects of interest in each frame
# from the webcam.

# Import packages
import ctypes
import os
import cv2
import numpy as np
import tensorflow as tf
import sys

# This is needed since the notebook is stored in the object_detection folder.
sys.path.append("..")

# Import utilites
from utils import label_map_util
from utils import visualization_utils as vis_util

# Set up Twilio
from twilio.rest import Client
```

Appendix (cont.)

```
# Twilio SID, authentication token, my phone number, and the Twilio phone number
account_sid = 'AC7d4e08979f94ee3b3b40fb77a89e1434'
auth_token = '49528bd7208314b99bfb856666ec18e'
my_number = '2542851102'
twilio_number = '19362510045'

client = Client(account_sid, auth_token)

## get Screen Size
user32 = ctypes.windll.user32
screen_width, screen_height = user32.GetSystemMetrics(0), user32.GetSystemMetrics(1)
frame_height, frame_width = 480, 640

scaleWidth = float(screen_width) / float(frame_width)
scaleHeight = float(screen_height) / float(frame_height)

if scaleHeight < scaleWidth:
    imgScale = scaleWidth
else:
    imgScale = scaleHeight

newX, newY = 0, 0
frame = cv2.imread(imgPath, imgScale)
WINDOW_NAME = 'Full Integration'

# Set up camera constants
IM_WIDTH = 640
IM_HEIGHT = 480

# Name of the directory containing the object detection module we're using
MODEL_NAME = 'inference_graph'

# Grab path to current working directory
CWD_PATH = os.getcwd()

# Path to frozen detection graph .pb file, which contains the model that is used
```

Appendix (cont.)

```
# for object detection.
PATH_TO_CKPT = os.path.join(CWD_PATH, MODEL_NAME, 'frozen_inference_graph.pb')

# Path to label map file
PATH_TO_LABELS = os.path.join(CWD_PATH, 'training', 'labelmap.pbtxt')

# Number of classes the object detector can identify
NUM_CLASSES = 4

## Load the label map.
# Label maps map indices to category names, so that when our convolution
# network predicts `5`, we know that this corresponds to `king`.
# Here we use internal utility functions, but anything that returns a
# dictionary mapping integers to appropriate string labels would be fine
label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
categories = label_map_util.convert_label_map_to_categories(label_map,
max_num_classes=NUM_CLASSES, use_display_name=True)
category_index = label_map_util.create_category_index(categories)

# Load the TensorFlow model into memory.
detection_graph = tf.Graph()
with detection_graph.as_default():
    od_graph_def = tf.GraphDef()
    with tf.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
        serialized_graph = fid.read()
        od_graph_def.ParseFromString(serialized_graph)
        tf.import_graph_def(od_graph_def, name='')

    sess = tf.Session(graph=detection_graph)

# Define input and output tensors (i.e. data) for the object detection classifier

# Input tensor is the image
image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')

# Output tensors are the detection boxes, scores, and classes
```

Appendix (cont.)

```
# Each box represents a part of the image where a particular object was detected
detection_boxes = detection_graph.get_detections()[0].get('detection_boxes:0')
```

```
# Each score represents level of confidence for each of the objects.
# The score is shown on the result image, together with the class label.
detection_scores = detection_graph.get_detections()[0].get('detection_scores:0')
detection_classes = detection_graph.get_detections()[0].get('detection_classes:0')
```

```
# Number of objects detected
num_detections = detection_graph.get_detections()[0].get('num_detections:0')
```

```
#### Initialize other parameters ####
```

```
# Initialize frame rate calculation
frame_rate_calc = 1
freq = cv2.FONT_HERSHEY_SIMPLEX
font = cv2.FONT_HERSHEY_SIMPLEX
```

```
# Define inside box coordinates (top left and bottom right)
TL_inside = (int(IM_WIDTH*0.46), int(IM_HEIGHT*0.25))
BR_inside = (int(IM_WIDTH*0.8), int(IM_HEIGHT*0.85))
```

```
# Define outside box coordinates (top left and bottom right)
#TL_outside = (int(IM_WIDTH*0.46),int(IM_HEIGHT*0.25))
#BR_outside = (int(IM_WIDTH*0.8),int(IM_HEIGHT*0.85))
```

```
# Initialize control variables used for weapon detector
detected_inside = False
#detected_outside = False
```

```
inside_counter = 0
#outside_counter = 0
```

```
pause = 0
pause_counter = 0
```

```
= Initialize frame rate calculation
```

Appendix (cont.)

```
frame_rate_calc = 1
freq = cv2.getTickFrequency()
font = cv2.FONT_HERSHEY_SIMPLEX

# Initialize webcam feed
video = cv2.VideoCapture(0)
ret = video.set(3, 1280)
ret = video.set(4, 720)

def detect_objects(frame):

    # Use globals for the control variables so they retain their value after
    function exits
    global detected_inside
    global inside_counter
    global pause, pause_counter

    frame_expanded = np.expand_dims(frame, axis=0)

while True:

    # Acquire frame and expand frame dimensions to have shape: [1, None, None, 3]
    # i.e. a single-column array, where each item in the column has the pixel RGB
    value
    ret, frame = video.read()
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    frame_expanded = np.expand_dims(frame_rgb, axis=0)

    # Perform the actual detection by running the model with the image as input
    boxes, scores, classes, num = sess.run(
        detection_boxes, detection_scores, detection_classes, num_detections,
        feed_dict={image_tensor: frame_expanded})

    # Draw the results of the detection (aka 'visualize the results')
    vis_util
    frame
```

```

np.squeeze(boxes)
np.squeeze(classes) # np.int32
np.squeeze(scores)
category_index
use_normalized_coordinates=True
line_thickness 8
min_score_thresh=0.50)

# Draw boxes defining "outside" and "inside" locations.
#cv2.rectangle(frame,TL_outside,BR_outside,(255,20,20),3)
#cv2.putText(frame,"Outside #box",(TL_outside[0]+10,TL_outside[1]-
10),font,1,(255,20,255),3,cv2.LINE_AA)
cv2.rectangle(frame,TL_inside,BR_inside,(20,20,255),3)
cv2.putText(frame,"Detection Area",(BR_inside[0]-
240,TL_inside[1]+35),font,1,(20,255,255),3,cv2.LINE_AA)
# Check the class of the top detected object by looking at classes[0][0].
# If the top detected object is a weapon (1-4)
# find its center coordinates by looking at the boxes[0][0] variable.
# boxes[0][0] variable holds coordinates of detected objects as (ymin, xmin,
ymax, xmax)
if int(classes[0][0]) or int(classes[0][0]) or
int(classes[0][0]) or int(classes[0][0]) and pause == 0
x = int(boxes[0][0] + boxes[0][2] / 2) / IM_WIDTH
y = int(boxes[0][1] + boxes[0][3] / 2) / IM_HEIGHT
# Draw a circle at center of object
cv2.circle(frame,x,y,1,1,1,1)
# If object is in inside box, increment inside counter variable
if x > TL_inside[0] and x < BR_inside[0] and y > TL_inside[1] and
y < BR_inside[1]
    inside_counter = inside_counter + 1

if inside_counter == 4:
    detected_inside = True
    message = "Weapon has been detected on your camera!"
    from twilio_number
    to my_number

inside_counter = 0
outside_counter = 0
# Pause weapon detection by setting 'pause' flag
pause

```

```

# All the results have been drawn on the frame, so it's time to display it.
cv2.imshow('Object detector', frame)

# Press 'q' to quit
if cv2.waitKey(1) &lt;= ord('q'):
    break

# Clean up
video.release()
cv2.destroyAllWindows()

```

Code for the Training of the Model

```

r"""Training executable for detection models.

```

This executable is used to train DetectionModels. There are two ways of configuring the training job:

1) A single pipeline_pb2.TrainEvalPipelineConfig configuration file can be specified by --pipeline_config_path.

Example usage:

```

./train \
  --logtostderr \
  --train_dir=path/to/train_dir \
  --pipeline_config_path=pipeline_config.proto

```

2) Three configuration files can be provided: a model_pb2.DetectionModel configuration file to define what type of DetectionModel is being trained, an input_reader_pb2.InputReader file to specify what training data will be used and a train_pb2.TrainConfig file to configure training parameters.

Example usage:

```

./train \

```

```

--logtostderr \
--train_dir=path/to/train_dir \
--model_config_path=model_config.pbtxt \
--train_config_path=train_config.pbtxt \
--input_config_path=train_input_config.pbtxt
"""

import functools
import json
import os
import tensorflow.compat.v1 as tf
from tensorflow.python.util import nest

from object_detection.builders import dataset_builder
from object_detection.builders import model_builder
from object_detection.builders import loss_builder
from object_detection.builders import input_builder
from object_detection.builders import output_builder

tf.flags.DEFINE_string('master', '', 'Name of the TensorFlow master to use.')
tf.flags.DEFINE_integer('task', 0, 'task id')
tf.flags.DEFINE_integer('num_clones', 1, 'Number of clones to deploy per worker.')
tf.flags.DEFINE_boolean('clone_on_cpu', False,
    'Force clones to be deployed on CPU. Note that even if '
    'set to False (allowing ops to run on gpu), some ops may '
    'still be run on the CPU if they have no GPU kernel.')
tf.flags.DEFINE_integer('worker_replicas', 1, 'Number of worker+trainer '
    'replicas.')
tf.flags.DEFINE_integer('ps_tasks', 0,
    'Number of parameter server tasks. If none, does not use '
    'a parameter server.')
tf.flags.DEFINE_string('train_dir', '',
    'Directory to save the checkpoints and training summaries.')

tf.flags.DEFINE_string('pipeline_config_path',
    'path to pipeline_config.pbtxt', 'PipelineConfig config')

```

```

        'file. If provided, other configs are ignored')

flags.DEFINE_string('train_config_path', '',
                   'Path to a train_pb2.TrainConfig config file.')
flags.DEFINE_string('input_config_path', '',
                   'Path to an input_reader_pb2.InputReader config file.')
flags.DEFINE_string('model_config_path', '',
                   'Path to a model_pb2.DetectionModel config file.')

```

```

FLAGS = flags.FLAGS

```

```

def main(argv=None, 'Use object_detection/model_main.py.'):
    def _load_configs():
        assert FLAGS.train_dir, 'train_dir is missing.'
        if FLAGS.task == 0: tf.gfile.Glob(FLAGS.train_dir)
        if FLAGS.pipeline_config_path:
            configs = tf.gfile.Glob(FLAGS.pipeline_config_path)
            FLAGS.configs.extend(configs)
        if FLAGS.task == 0:
            tf.gfile.Glob(FLAGS.os_path + FLAGS.pipeline_config)
            overwrite = True
        else:
            configs = tf.gfile.Glob(FLAGS.os_path + FLAGS.pipeline_config)
            overwrite = True
        if FLAGS.task == 0:
            for name, config in ('model.config', FLAGS.model_config_path),
                               ('train.config', FLAGS.train_config_path),
                               ('input.config', FLAGS.input_config_path):
                tf.gfile.Glob(FLAGS.os_path + config)
                overwrite = True
    _load_configs()

```

```

model_config = configs['model.config']
train_config = configs['train.config']
input_config = configs['input.config']

```

```

model_fn = functools.partial

```

```

model_builder.build(
    model_config=model_config,
    is_training=True,

def _create_model(config):
    return dataset_builder.make_data_loader(
        dataset_builder.build(config)).get_model()

create_input_dict_fn = functools.partial(_create_model, input_config)

env = json.loads(os.environ.get('TF_CONFIG', '{}'))
cluster_data = env.get('cluster', None)
cluster = tf.train.ClusterSpec(cluster_data) if cluster_data else None
task_data = env.get('task', None) or {'type': 'master', 'index': 0}
task_info = type('TaskSpec', object, {})(task_data)

# Parameters for a single worker.
ps_tasks = 0
worker_replicas = 1
worker_job_name = 'lonely_worker'
task = 0
is_chief = True
master = ''

if cluster_data and 'worker' in cluster_data:
    # Number of total worker replicas include "worker"s and the "master".
    worker_replicas = cluster_data['worker'] + 1
if cluster_data and 'ps' in cluster_data:
    ps_tasks = cluster_data['ps']

if worker_replicas == 1 and ps_tasks == 0:
    raise ValueError('At least 1 ps task is needed for distributed training.')

if worker_replicas == 1 and ps_tasks == 1:
    # Set up distributed training.
    server = tf.train.Server(cluster=cluster, protocol='grpc',
                             job_name=task_info,
                             task_index=task_info)

```

```

if task_info.type == 'ps':
    server.join()
    return

worker_job_name = '%s/task:%d' % (task_info.type, task_info.index)
task = task_info.index
is_chief = (task_info.type == 'master')
master = server.target

graph_rewriter_fn = None
if 'graph_rewriter_config' in configs:
    graph_rewriter_fn = graph_rewrite_builder.build(
        configs['graph_rewriter_config'], is_training=True)

trainer.train(
    create_input_dict_fn,
    model_fn,
    train_config,
    master,
    task,
    FLAGS,
    worker_replicas,
    FLAGS,
    ps_tasks,
    worker_job_name,
    is_chief,
    FLAGS,
    graph_hook_fn, graph_rewriter_fn)

if __name__ == '__main__':
    tf

```

AVERAGE

Outcome 1

Average

Complex Engineering Problem Evaluation Form
ECE Senior Design & Professionalism
Outcome 1

Title of Presentation:

Team 7 – Home Security Anti Burglar System
 Oluwadamilola Osuntuyi, Nolan Coston, Elizabeth Reyna, Teminioluwa Folorunso, Ben Ombusa

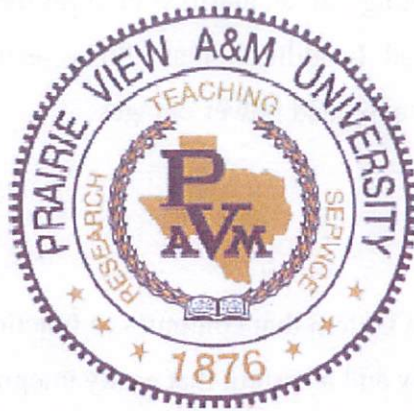
Date of Presentation:

Name of Evaluator and Affiliation (University or Company name)

	Excellent 99	Very Good 89	Average 79	Below Avg. 69	Poor 54	Comments
Identify and Formulate						Chapter 1
Able to understand the given problem and identify the subject area and concepts involved		✓				Frq. 8 - Good
Able to identify what information is known and what needs to be determined		✓				
Able to convert the problem into a well labeled sketch or appropriate representation (free body diagram, flow chart, functional block diagram, schematic diagram, etc.)			✓			
Able to formulate the complex engineering problem into an engineering model or design – showing any math or science or technology needed to validate the design		✓	✗			
Solve Complex Problem						Chapter 6
Able to design a practical solution to address the problem.		✓				
Able to identify parts and other needed capability (like coding) to solve the problem or proposal		✓				
Able to deliver a final solution that integrates all aspects of the design from hardware to software		✓				
The final solution should be consistent with project deliverables and able to demonstrate the objective has been achieved		✓				

87

**PRAIRIE VIEW A&M UNIVERSITY
ROY G. PERRY COLLEGE OF ENGINEERING**



**HOME SECURITY ANTI-BURGLAR SYSTEM
SENIOR DESIGN AND PROFESSIONALISM II
ELEG 4248 – SPRING 2022**

FINAL WRITTEN REPORT

Date Due: 5/1/2022

Team 7:

Ben Ombura
Elizabeth Reyna
Oluwadamilola Osuntuyi
Teminioluwa Folorunso
Nolan Coston

Submitted to:

Professor Turner, Vewiser

Sponsor:

Dr. Akujuobi, Cajetan

EXECUTIVE SUMMARY

Problem Statement

In the United States, an average of 4.1 property crimes occurs per second. Customers tend to pay high costs when investing into an alarm system for their protection. Often, older alarm components cannot be reused. In addition, many home security systems often fail to monitor and communicate during extended power outages.

Objective

To design a security alarm system that continues to function during power outages with the use of a versatile power supply and a system that easily integrates with a variety of monitoring and Internet of Things (IoT) devices. Our alarm system will be a low-cost design that provides security for a home and office. Home Assistant and Zigbee Alliance will be utilized to seamlessly unify brands customers love.

Project Deliverables

- Remote Monitoring
- Multi-detection sensors
- Alternative energy source

Target Audience

Homeowners and Businesses

Budget and Spending

- Project Budget: \$600
- Amount Spent: \$560

EXECUTIVE SUMMARY (continued)

Customer Requirements and Engineering Solutions

- Customers should be able to interact with security system remotely.
- Mobile application should be user-friendly.
- Camera should detect moving objects within range.
- Security alarm should notify client when alarm system is triggered due to sensor detection of potential intruder.

Major Constraints

- Storing the power with the solar panel.
- Lack of experience in code development for mobile applications.
- Electromagnetic jammers could disrupt the system.

Benefits

- **Safety**: The overall goal of this project is to create a product that increases the safety of the user and the customer privacy of the user using Home Assistant. This will keep more homes and businesses safer. Users will feel more comfortable away from their properties due to our unique design utilizing Home Assistant open-source software. Our design uses multiple detection sensors such as a camera, magnetic reed switch (door sensor), and a temperature and humidity sensor.
- **Economic**: Homeowners have a high probability of receiving discounts from Homeowners due to the fact that they have a security alarm system in their home or in this case our alarm system in their home. Security alarm systems tend to increase the value of the property. Therefore, owners can make more profit when selling or renting out their property to others. Our alarm system uses less expensive components than most systems while still being as effective or more effective at maintain the security of a property.

EXECUTIVE SUMMARY (continued)

- **Technical:** Our project utilizes LoRa module and ESP32 Cam microcontroller to provide security through long range communication and Wi-Fi/Bluetooth capabilities. For this project we mainly focused on utilizing the Wi-Fi and Zigbee capabilities of our design. Our alarm system will have a remote user interface that enables users to monitor and control the system from anywhere around the world.

Recommendation

- Compact and remotely accessible design
- Use solar power, as a renewable resource, for self-sustaining energy
- Device must be waterproof and weather-resistant
- Long lasting battery power

Conclusion

The design of our security system will impact the standard of safety in societies by securing people and their possessions even during extended power outages.

TABLE OF CONTENTS

EXECUTIVE SUMMARY	2-4
TABLE OF CONTENTS	5
LIST OF FIGURES	6
LIST OF TABLES	6
LIST OF TEAM MEMBERS AND ROLES	6
1. PROJECT SCOPE	7
Project Statement, Business Objective and Target Audience.....	8
Client identification and recognition of need.....	9-10
Project goals and objectives.....	11-12
Contemporary issues relevant to project.....	13
Potential impact of project on society.....	14-18
Initial Project constraints.....	19
2. PROJECT PLANNING AND TASK DEFINITION	20
2.1 Task Identification	21
2.2 Gantt Chart	22
3. LITERATURE REVIEW	23
3.1 List of all sources of information	24
3.2 Statement of Learning from each source	25-26
4. PRELIMINARY DESIGN	27
4.1 Concept Generation, Evaluation and Selection.....	28-30
4.3 Design Constraints Used in the Preliminary Design.....	31
4.3 Applicable Codes and Standards Used in the Preliminary Design.....	32
4.4 Preliminary Analysis	33-34
4.5 Preliminary Design.....	35
5. DETAILED SYSTEM DESIGN	36
5.1 Project Parts Summary	37
5.3 Overview of Code Developed to Support Engineering Analysis.....	38-40
5.4 Application of Engineering Codes and Standards used in Final Design.....	41
5.5 Economic Analysis	42
5.6 Final Design	43-47
6. RESULTS AND DISCUSSION	48
6.1 Summary of Goals and Objectives.....	49
6.2 Summary of Constraints and Codes Met by Design	54
6.3 Discussion of Results and Other Use Cases	55
7. CONCLUSIONS AND RECOMMNDATIONS	56-57
LIST OF REFERENCES	58-61
APPENDIX	62

LIST OF FIGURES

Figure 1 – Statista Digital Market Statistics.....	14
Figure 2 – Home Burglary Statistics.....	16
Figure 3 – Security System Application and Home.....	17
Figure 4 – Electronics Waste Diagram.....	18
Figure 5 – Project Schedule.....	21
Figure 6 – Gantt Chart.....	22
Figure 7 – Project Comparison.....	29
Figure 8 – Engineering Design Methodology.....	30

LIST OF TABLES

Table 1 -	##
Table 2 - Comparison of Security Alarm Systems.....	##

LIST OF TEAM MEMBERS AND ROLES

Elizabeth Reyna.....	Team and Test Lead
Oluwadamilola Osuntuyi.....	Scribe and Test Lead Assistant
Nolan Coston.....	Software Programming and Document Managing Assistant
Teminioluwa Folorunso.....	Software Programming Assistant and Document Manager
Ben Ombura.....	Software Programmer and Test Lead Assistant

PROJECT SCOPE

Project Statement, Business Objective, and Target Audience

Problem Statement: Many customers often pay a high cost when investing in alarm systems as a form of protection. More than often, older alarm components cannot be reused. In addition, many alarm systems cannot function during power outages.

Businesses Objective: To design a low-cost alarm system that provides the comfort of a secure home or office. Home Assistant and The Zigbee Alliance will allow our alarm system to seamlessly unify brands we all love. The alarm system will also monitor and communicate during extended power outages. We plan to advertise all the perks and benefits of our alarm system to increase interest from our audience during our presentation of our completed product such as the fact that our system will use solar power (renewable energy source) and many other perks.

Target Audience: Homeowners and businesses. Because both are the most targeted by burglars.

Client Identification and Recognition of Need

The security alarm system must be simple enough for homeowners and businesses to set up. Our design will be easy to manage for our customer due to the platform we have decided to use, Home Assistant, as an application for the user. Home Assistant is user friendly and will allow the user to easily customize their system, monitor, and control their system. The user can also develop their own code to utilize a sensor or any component within the system to their own liking. The microcontroller (ESP32 CAM) will be used in our design and integrated onto Home Assistant. This microcontroller camera must be capable of sending and receiving messages and commands. Our use of a LoRa module in this design will enable long range communication within and outside of the property. Homeowners and businesses need multiple detection sensor for overall safety. Therefore, this design will utilize a camera, magnetic reed switch (door sensor), and a temperature and humidity sensor. Home Assistant allows the user to many replace similar components being utilized int the system with newer or older components. This highlights Home Assistants high level of compatibility regarding seamless integration of various brands or types of components. Our design will utilize a breadboard containing and resistors, wires, a buzzer, and an LED. The LED will flash when alarm system is triggered which will inform the user that the system has been triggered. Additionally, the buzzer will sound off at a high, also adjustable, volume so the user will be able to have audible confirmation of the alarm system being triggered.

Need Stronger
reference to
this page &
Table 1 on
next page
?

Client Identification and Recognition of Need (continued)

Table 1: Marketing and Engineering Requirements to highlight client identification and recognition of needs:

MARKETING REQUIREMENTS	ENGINEERING REQUIREMENTS	JUSTIFICATIONS
Integrated communication solution	• ESP32 microcontroller • REYAX Lora module • Texas Instrument Sonoff Zigbee analyzer	• Low energy system • Sends data and live feed for up to 12 kilometers. • Embedded Wi-Fi, Bluetooth, and Zigbee capabilities.
Remote monitoring	• Self-monitoring through application • Optional 24/7 emergency and monitoring services	• 24/7 monitoring of property will reassure a customer of investment.
Multi detection sensors	• Reed switch sensor (door / window sensor) • Camera • Temperature and Humidity sensor	• Multiple sensors collect readings to alert the customer of a hazard or a security breach.
Off-grid power source	• Solar power • Rechargeable battery source (12V 10Ah Lithium Iron Phosphate Battery)	• Additional marketing tool for sales. • Provides protection during extended power outages. • Customer safety increased.

Project Goals and Objectives

- To create a low-cost security alarm system

How?

- Research simple yet effective alarm system designs
- Identify needed components and research costs to purchase components and the reliability or reputation of each brand. Additionally, weigh pros and cons.
- Develop design and test design for energy consumption and consistency in proper operation.

- User friendly system

How?

- Utilize a home automation platform that provides easy customization and integration capabilities. This home automation platform must be compatible with various brands and types of sensors a user may want to choose from to add to their system. Dashboard on platform must be easy to control and data should be readily available on the dashboard of the home automation application.

- Broad impact in multiple areas:

How?

- We have designed our system to touch multiple areas of impact such as safety/welfare, economic, technical, and environmental.

Project Goals and Objectives (continued)

- Renewable energy use, power supply that enables monitoring and communication during power outage.

How?

- Utilize solar power by incorporating a solar panel within the design of our power supply. Many people are looking for products that utilize clean energy. In addition, solar is the most abundant and renewable energy source. The design of our supply will be useful in the case of a power outage, enabling the alarm system to still monitor and communicate in that case. The power supply will also utilize a 12V 10Ah Lithium Iron Phosphate Battery.
- 12V 10Ah Lithium Iron Phosphate Battery Features:
 - 3000+ cycles and 10+ years life span
 - High energy density and lighter weight
 - Built-in battery management system (BMS)

- Notification of threat and alarm triggered

How?

- Program system to send a notification to the user's phone and lcd or tablet screen when sensors detect a threat and when alarm system is triggered. Programming for sensors will be done on ESPHome within Home Assistant. Programming for camera will be done on Arduino to access live camera feed, feed will be imported onto Home Assistant.

Contemporary Issues Relevant to Project

Customers tend to overlook some of the benefits of an alarm system besides safety. When looking to purchase an alarm system, customers should look for systems that are customizable and reduce energy bills. Features such as automated temperature control and programmable lighting features can significantly decrease energy bills. According to the U.S. Environmental Protection Agency, a programmable thermostat can save homeowners an average of \$180 every year in energy costs. Automated and “smart” features let you create customized schedules for lighting and thermostat activities to meet your routine and save energy. Additionally, you can manage them remotely from your smartphone.

Note: Since our system utilizes “smart” tech this will allow the user to program any sensor of their choosing to their own benefit, one of those benefits being reduced energy bills.

Potential Impact of Project on Society

Our project impacts four major areas: economic, safety and welfare, technological, and environmental.

Economic Impact

As recorded by Statista Database company, as shown in **Figure 1** below, the installation of home security systems has gained a revenue of 5 billion dollars as of 2021 and is expected to increase to over 8 billion dollars by the year 2025. Therefore, signifying the demand for alarm systems around the world. Below are further benefits of our product to the economy:

- Lower product cost: The security alarm design utilizes less expensive parts than most security alarm systems; therefore, makes our design a more affordable product to purchase on the market.
- Lowers insurance: Having our design framework in your home can make home providers discount your mortgage holder's protection up to 20%.
- Increases property value: Installation of our security alarm design in homes and business will increase the value of many properties.

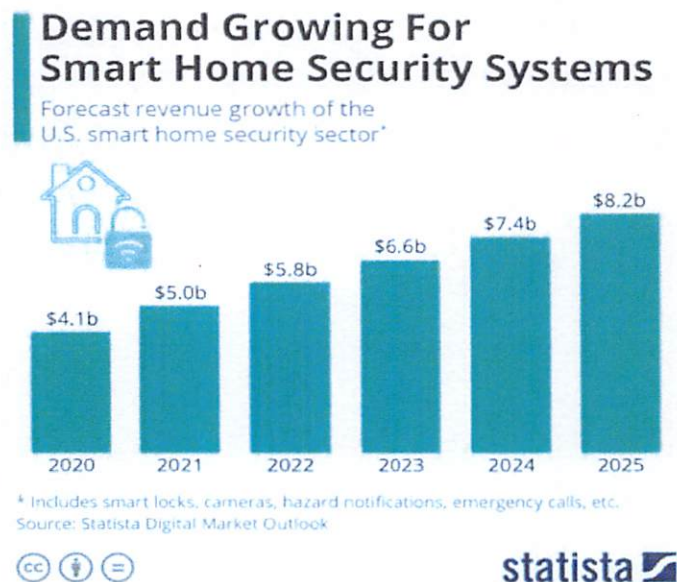


Figure 1 - Statista Digital Market Statistics

Safety and Welfare Impact

Protects valuables: This is an obvious advantage a many people consider. We probably all know somebody who has lost gadgets, adornments, or other valuable possessions because of a home or business robbery. The misfortune is intensified when the possession is an indispensable family treasure. A security system can assist in protecting these possessions that could be potentially stolen in a robbery. No one enjoys losing a possession especially the ones we hold dear to us.

Notifies you of fire or gas issue: With this alarm design the user can receive notifications of the status of the temperature and smoke or gas levels within the property. The alarm could go off when levels reach the point of a potential threat to safety.

Helps keep tabs on kids: Many homeowners have children. As the saying goes, “children are our future”, this alarm system will be a device that parents can rely on to protect the future of their children by keeping them safe. If you're away from home, and you want to monitor your home or kids back home, you can utilize our portable app to watch what's going on in your home by means of video via a surveillance camera(s). If you have an electronic entryway lock, you can open them to give the children access after school, so they don't need to stress over losing their own key to the house and unfortunately waiting for an in-person assistance to open a door to the house.

Improves peace of mind: The conviction that all is good and solace you gain with an alert framework is maybe the best advantage of all. Close to being protected, the certainty of having a sense of security will assist you with being a more useful, sound, and centered individual.

Adding a security framework to your home provides you with an additional a layer of protection against any expected interlopers. Regardless of whether you lease or own, you can partake in the advantages that accompany having a security framework.

Safety and Welfare Impact (continued)

Deters crime: A study by Rutgers University tracked down that the large quantity of home security alarm diminishes the quantity of burglaries. Having a security alarm ensures customers' safety, frightens away burglars, and increases neighborhood peace.

Figure 2 below shows statistics from Engineering Protection System (EPS) on home burglary.

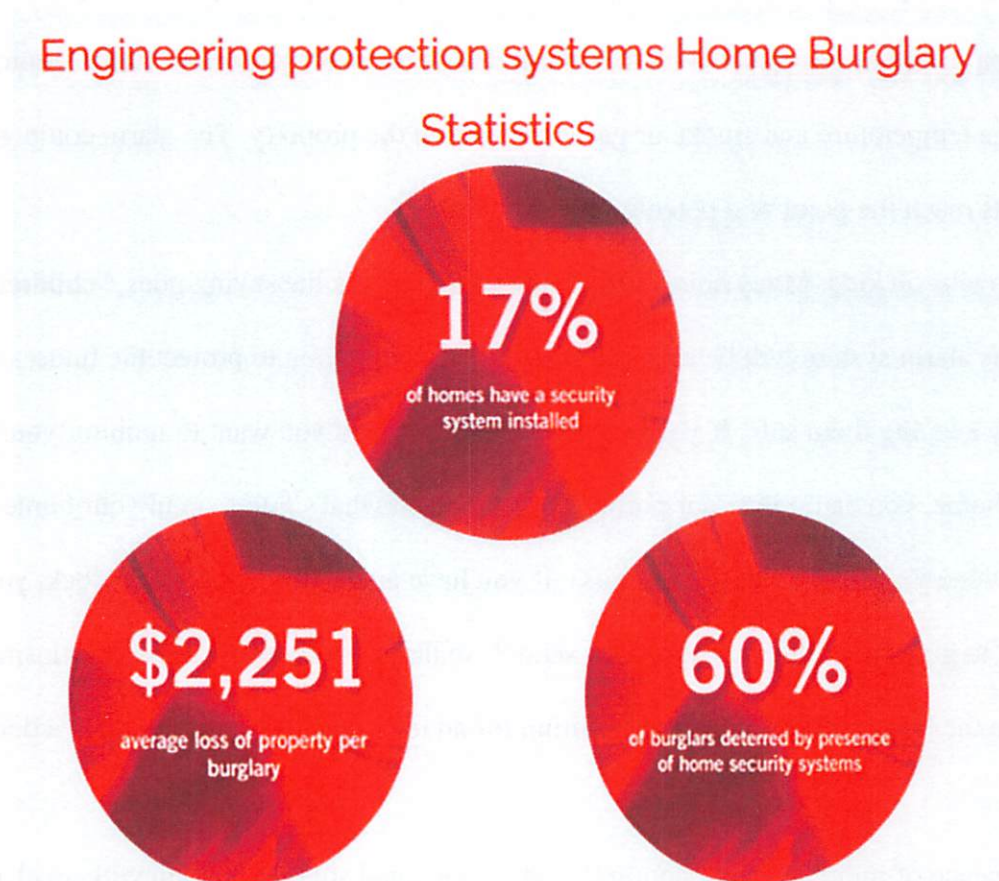


Figure 2 - Home Burglary Statistics

Start
Explain
the
figure
Don't
leave
to
reader

Technological Impact

Remote access: When the customer is away from their home, they can utilize the mobile application to survey their home by means of video footage from other surveillance camera(s) and data from sensors which will be provided on the Home Assistant application.

Tight security: With the implementation of YAML language and Home Assistant, an end-to-end encryption will be used to protect the system from possible hackers and enhance the level of security.

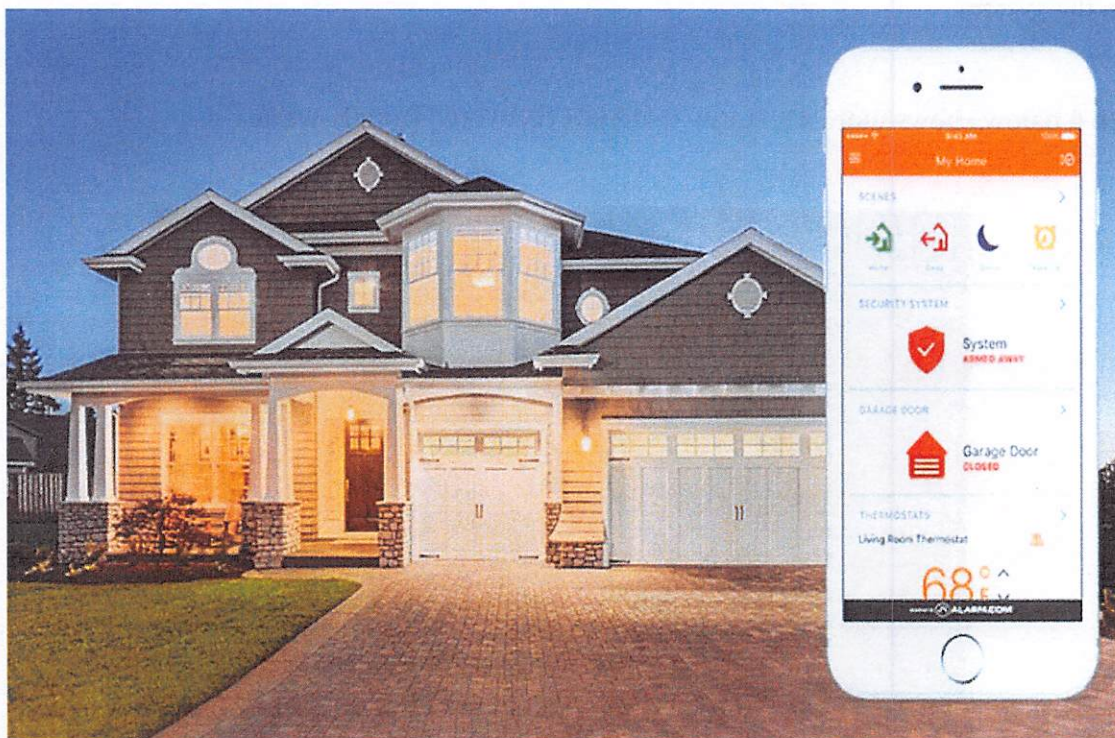


Figure 3 - Security System App and Home

Environmental Impact

When electronics are improperly disposed, toxic chemicals are released to the environment; therefore, impacting the earth's soil, the air, water, and the public's health. According to Global E-Waste Monitor, about \$54.2 million worth of raw materials was lost in 2021, while only \$8.5 million worth of raw materials were recovered.

Helps electronic waste: Our design will allow users to use older and newer components that they may have and integrate them into their alarm system. This is possible due to the fact that our alarm system is a smart alarm system that allows the user to program any sensor of their choosing on Home Assistant to tailor the system to their needs in regard to a particular sensor within the system.

Figure 4 below shows more about raw materials recovered from E-waste.



Figure 4 – Electronic waste diagram

Initial Project Constraints

We needed a solar panel as a source of supplying alternative power to security system, but the solar panel was one of the most expensive parts. Our budget was only \$600 for this project. Therefore, we had to find more affordable prices for the other parts. Our minimal cybersecurity experience was hurdle we must overcome. In order to do this, we researched and educated ourselves on cybersecurity, specifically in relation to a security alarm system. Without cybersecurity, our security system is vulnerable to hackers, which endangers the client or their property. Another vulnerability is that our alarm system may be vulnerable to electromagnetic jammers, so we made sure to research on that topic.

PROJECT PLANNING AND TASK DEFINITION

Task Identification

Our team started determining phases to our project plan. We completed our hardware research and made changes to the parts list. We worked on the hardware and software development of our security alarm system. **Figure 5** shows our project schedule.

Project Schedule

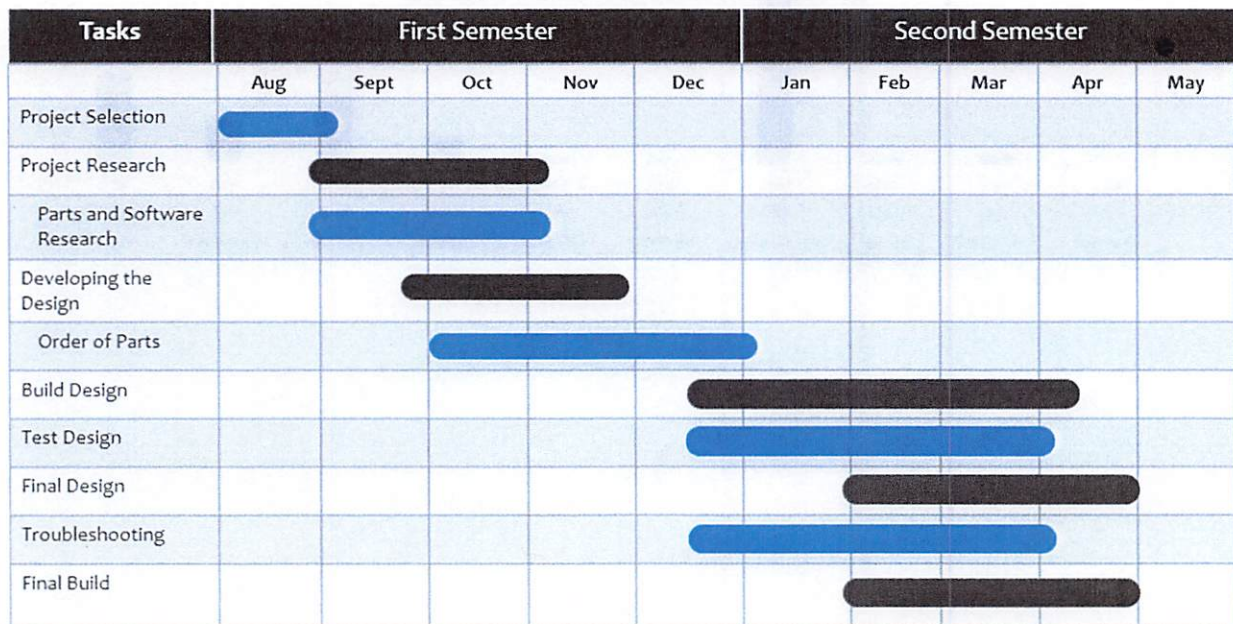


Figure 5 - Project Schedule

Gantt Chart

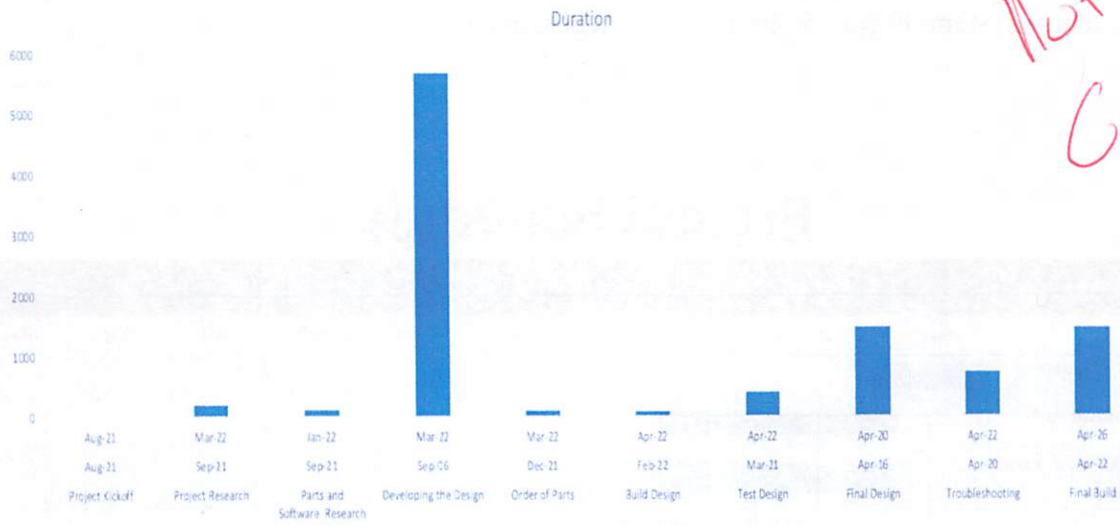


Figure 6 - Gantt Chart

Not a Gantt Chart

LITERATURE REVIEW

List of all sources of information

The research of our security alarm system was compared with other home security systems. We compared our design to ADT security system and Simpli Safe Alarm System. The ADT wired

security system is one of highly rated home securities that operates the largest national network of professional monitoring centers with money-back guarantee. Unfortunately, ADT only pro- installs full system; therefore, they do not install standalone devices. The pricing, depending on the package, can reach up to about \$1,450 upfront or up to \$82 monthly for 60 months. The Award Winning Wireless Simpli Safe Alarm system is an easy-to-setup system that is ran by professionals with no contracts. Simpli charges up to a monthly cost of \$25 or minimum upfront cost of \$230.

Statement of Learning from each source

Before we started our research and literature review for our project. We looked over the requirements and objectives of our security alarm system project by reviewing the project description we were provided with. After understanding the requirements and objectives of our selected project, we decided to focus on researching and gathering information on several topics: benefits of security alarm systems, existing alarm systems, alarm system project ideas, project impact areas, potential alarm system components, functions of alarm system components, compatibility, and app development.

As we did our research, we made sure to gather our information from credible sites and databases. As we researched about the benefits, we wanted to focus on gathering information on benefits that most people may not know about alarm systems such as the fact that most insurance carriers offer discounts on insurance premiums if a home has an installed security system. When gathering research about existing alarm systems we focused on understanding more about how an alarm system works and the communication aspect of the system. To get an idea of how we wanted to design our system, we gathered ideas from sites online on alarm system projects that have been done. Furthermore, we did research on how we can make our alarm system unique in comparison to the alarm system projects that we came across through our research.

Regarding project impacts, we concluded that our project would have an impact in five areas and gathered research on how alarm systems make an impact in those areas. The five areas of impact we decided to do research on are: economic impact, technological impact, social impact, global impact, and safety/welfare impact. The most difficult part of research and literature review was deciding what specific components we would like to use in our design. For our design we needed to have a transmitting side and receiving side consisting of two microcontrollers, two LoRa modules, sensors, reed switch, power supply, and lcd screen/buzzer (alert notification devices)

Statement of Learning from each source (continued)

In order to settle on specific components to include in our design, we gathered research/literature on various types of microcontrollers, LoRa modules, sensors, reed switches, power sources, and LCD screens/buzzers. Furthermore, we did research to understand the functions of each component. After, we then chose our components based on whether the component would help our team achieve our desired design and purpose of our alarm system based on the features provided by the component. The last area we decided to do research was app development, specifically for security alarm systems. We were able to find sites such as GitHub.com, that provided information on how to develop code for a security alarm system.

PRELIMINARY DESIGN

Concept Generation, Evaluation and Selection

Considered designs

- Smart security design
 - Utilizes smart features and programmable capabilities for components within the system.
 - Remote monitoring via cloud and or local control.
 - Adjustable settings.
 - Saving in energy bills due to automated control and programable features of system.
- Wired ADT security design
 - Wired system and reliable. The system does not come with user programmable capabilities.
 - System is monitored by ADT; the system comes with a minimum 36-month contract. Equipment is highly expensive on top of expense for the contract. ADT is a company that provides alarm monitoring services throughout the United States.
- Wireless Simpli Safe security design
 - This system function wirelessly like the smart security alarm system. The design utilizes parts only provided from Simpli Safe. These parts are relatively expensive in comparison to a smart alarm system that utilizes Home Assistant like our design. Our design allows you to use multiple brands of components that can be programmed and integrated to their smart security system. This allows the use to have more control over the cost of their system. Simpli Safe does not provide this ability.
 - This system includes monthly continuous fees for monitoring services from Simpli Safe.

Concept Generation, Evaluation and Selection (continued)

Below in **Figure 6** is a product comparison table of the designs we considered in comparison to the design we chose, which was the smart security alarm system. The designs compared were the smart security system, wired ADT security system, and the wireless Simpli Safe alarm system. In order to conclude which design to choose, we weighed the pros and cons between all three designs in the table below (Figure 6). Figure 6 also highlights the criteria we used to select our design as you'll see below. We concluded that the pros of the smart security alarm system superseded any of the pros of the others and the smart security system had less cons. In addition, the smart security cost less than the others in regard to upfront cost, contracts, and maintenance.

Product Comparison

PRODUCT	IMAGE	PROS	CONS	COST	REFERENCE	DECISION
Smart Security System		• Works off-grid • Integrate multiple brands • Serves as a smart home control panel	• Vulnerable to burglars due to jamming devices	\$600	N/A	✓
Wired ADT Security system		• Extremely reliable • Access to wide range of smart home devices	• Expensive installation • Minimum 36-month contract required	\$600 to \$2500	https://www.thisoldhouse.com/home-safety/21355319/adt-home-security	✗
Wireless Simpli Safe alarm system		• Easy to setup and install • Systems available in DIY kits	• Possible connection errors • High up-front cost	\$200 to \$1000	2021 Home Security System Cost Monthly Alarm System Prices (homeguide.com)	✗

Figure 7 - Project Comparison

Concept Generation and Evaluation and Selection (continued)

Our team first came up with the methodology to complete our design. Our team collaborated to develop our design methodology in steps shown in **Figure 7** below.

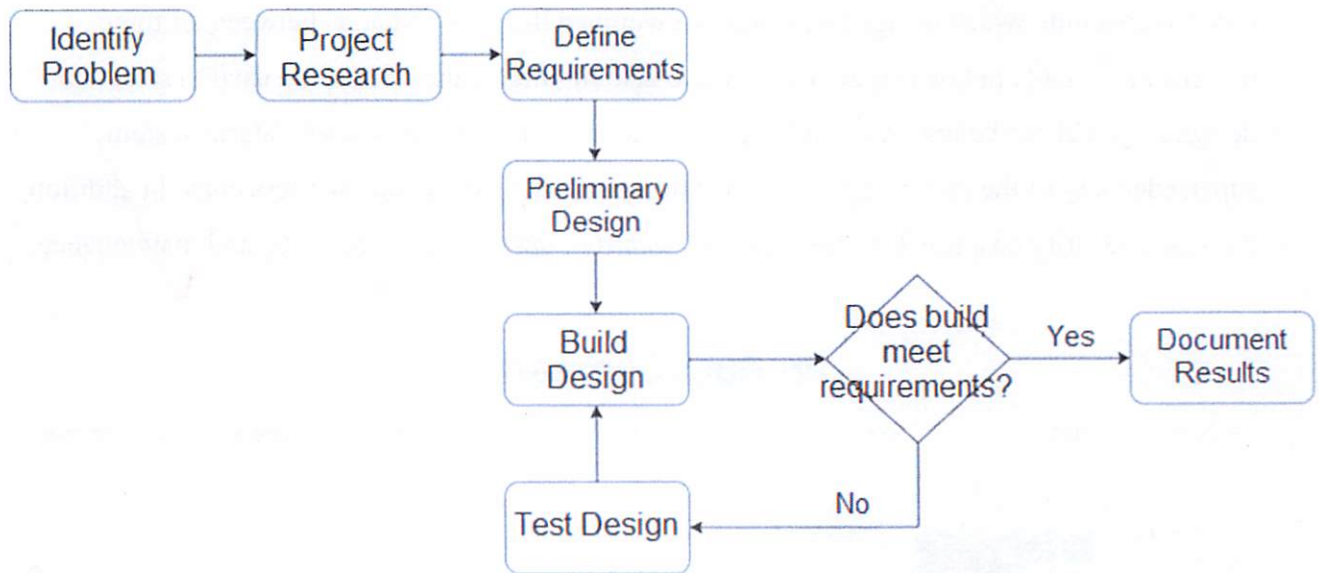


Figure 8 - Engineering Design Methodology

Using our design methodology, our team started with identifying the problem. Next, we researched sources to gain knowledge and support ideas behind our project. With a sufficient information, our team defined and understood the requirements of our project. After that, our team created concepts to form a preliminary design. Once the preliminary design agreed, we began building the design. There were a few times where our design did not meet the requirements and we needed to test design and fix any issues until the build requirements were met. After a few iterations and a successful build, we documented our results.

Design Constraints Used in the Preliminary Design

Design Constraints

Our security alarm design had constraints and limitations that were difficult to overcome. As a team, we needed to shift focus to ensuring the major parts of an average security alarm worked. One of our constraints was that our team wanted to use cybersecurity to use protocols that protect against outside harm, such as hackers, DoS attacks, viruses, etc., but we had minimal knowledge in cybersecurity. To resolve our minimal knowledge in this area, we gathered information from YouTube and sought professional help for our design. We also used Zigbee Alliance to smoothly connect many devices which makes it harder for predators to interrupt the system. Another constraint was our \$600 price budget and we needed to find affordable parts for our design. A few parts were expensive, so we compared expensive parts with cheaper parts and made informed decisions to deliver a highly reliable design while staying under budget. Another constraint of our design was its ability to counter electromagnetic pulses (EMP). While EMPs are not commonly used and usually affect smaller devices, some metal wiring, a capacitor, and a large power source can be enough to disable or even damage security alarm design. In case of emergency, most of the crucial components can be stored in a metal box that is EMP resistant.

Applicable Codes and Standards Used in the Preliminary Design

Applicable Codes and Standards

While working on certain project designs, it is important to take note and stay in compliance with the safety rules, codes, and standards. It is very important to acknowledge, recognize, and abide by them. As a group we concluded that our primary focus should be on the IEEE 7.8 – Code of Ethics, highlighted in red. The list of applicable codes and standards used in the preliminary design are as followed:

- IEEE 7.1 - United States Engineering Registration: The IEEE recognizes that licensure and registration contributes to the professions' efforts to protect the health, safety, and welfare of the public by ensuring that practitioners meet minimum recognized levels of education, experience, and competence.
- IEEE 7.6 – Protection of the Public: The IEEE recognizes the obligation of the profession to protect the health, welfare and safety of the public.
- IEEE 7.8 – Code of Ethics:
 - 1) To hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices.
 - 5) To seek, accept, and offer honest criticism of technical work.
- IEEE Std 1547-2020 - IEEE Conformance Test Procedures for Equipment Interconnecting Energy Resources with Electric Power Systems and Associated Interfaces: This standard is recognized that a distributed energy resource can be a single device providing all required functions or an assembly of components, each having limited functions and capabilities.
- IEEE/ASHRAE Guide for the Ventilation and Thermal Management of Batteries for Stationary Applications: Ventilation of stationary lithium-ion battery installations is critical to improving battery life.
- NESC - National Electrical Safety Code: Electrical safety precautions are specific control measures implemented to remove electrical hazards and mitigate the risks of electrical accidents and injuries.

Preliminary Analysis

When developing our initial code on the Arduino IDE platform for the camera aspect of our alarm system, we ran into multiple errors such as a “timed out waiting for packet header” error during our preliminary design phase. In order to fix this, we researched this error on google and tried various suggestions from our internet “friends.” We found out that for our code to upload successfully onto our ESP32 Camera microcontroller we had to change our upload speed on Arduino from 921,600 bits per second to 115,200 bits per second. In addition, we had to change our flash frequency from 40 MHz to 80 MHz. After doing this, our code uploaded successfully on to the ESP32 Camera microcontroller. Our chosen ESP32 cam microcontroller was perfect for our project due to features we needed like its 540 KB SRAM, 448 KB of ROM, and 16 KB of RTC SRAM. This microcontroller served well for data storage purposes of our project. The ESP32 Cam microcontroller satisfied what we were looking for in regard to high speed Wi-Fi connectivity. The device supports 802.11 b/gn Wi-Fi connectivity with speeds up to 150 Mbps! As far as battery life, we tested our design to establish how long the battery could power the system without being plugged to a charger. Our battery was able to work continuously for 4 days without charge. Therefore, we established that every 4 days the battery should be hooked up to a charger. The battery was able to be fully recharged in less than a few hours of charging.

For our temperature humidity and magnetic reed switch (door and window sensor), we decided to use the Aqara brand. The Aqara temperature and humidity sensor has a humidity detection accuracy of $\pm 3\%$ and temperature detection accuracy of $\pm 0.3^{\circ}\text{C}$. The Aqara door and window sensor detects the status of doors or windows via the distance of the magnets that are inside the sensors between the sensor and the magnet. The sensor is also energy efficient, and one battery can last over 2 years under normal usage.

Preliminary Analysis (continued)

Specifications of the Aqara Temperature and Humidity Sensor

Wireless Protocol:

Zigbee

Dimensions:

36 × 36 × 9 mm (1.42 × 1.42 × 0.35 in.)

Temperature Range and Precision:

-20°C – +50°C, ±0.3°C (-4°F – +122°F, ±0.5°F)

Humidity Range and Precision:

0 – 100% RH (non-condensing), ±3%

Atmospheric Pressure Range and Precision:

30 kPa – 110 kPa, ±0.12 kPa

Specifications of the Aqara door and window sensor

Wireless Protocol:

Zigbee

Dimensions:

41 × 22 × 11 mm (1.61 × 0.87 × 0.43 in.)

Maximum Detection Distance:

22 mm

Operating Temperature:

-10°C – +45°C (14°F – +113°F)

Operating Humidity:

0 - 95%RH, non-condensing

Preliminary Design

After research, concept evaluation, and selection of design, we determined how we wanted to construct our initial design. Below in **Figure 8** was our initial design or preliminary design that we developed.

Preliminary Engineering Design

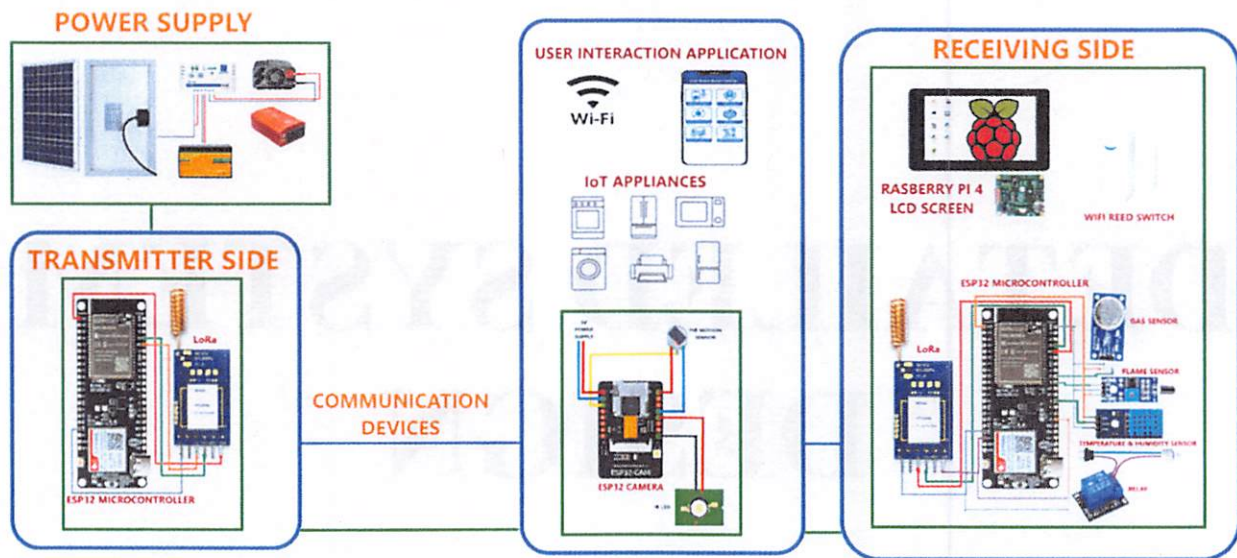


Figure 8 - Preliminary Design

This design consisted of several aspects such as a power supply, transmitting side, receiving side, and Wi-Fi / Bluetooth communication with sensor components. Transmitting side consisted of an ESP32 microcontroller and LoRa (Long Range) module for long range communication with the system. The receiving side consisted of and at home or business LCD screen, Wi-Fi reed switch, gas sensor, temperature and humidity sensor, flame sensor, and relay. In addition, a LoRa module, Raspberry Pi 4, and a ESP32 Camera were included on the receiving side. We wanted our system to be able to communicate with any IoT appliances on the users property such as a refrigerator, washing machine, or a microwave as you can see in the middle portion of Figure 8 above. As you continue in this report you will see that our group made changes within this design to establish our final design for our security alarm system.

DETAILED SYSTEM DESIGN

Project Parts Summary

In **Figure 9** below is our project parts summary for our security alarm system project.

Project Parts Summary

Power Supply



- 10W Solar Panel
- 10A Charge Controller
- Power Inverter
- Power Cords
- Batteries (120V 10Ah lithium iron phosphate rechargeable battery)

Sensors



SENSOR

- ESP32 Camera
- Aqara Temperature & Humidity Sensor
- Aqara Door and Window Sensor

Communication Systems



- ESP32 Microcontroller (Wi-Fi/Bluetooth Connection)
- REYAX RYLR8g6 Lora Module
- Samsung Tablet
- Texas Instrument Sonoff Zigbee analyzer
- Raspberry Pi 4
- Passive Buzzer
- LED

Cameras



- ESP32 Wi-Fi Bluetooth Camera Module
- Wide range detection

Figure 9 – Project Parts Summary



As shown in Figure 9, the security alarm system uses an ESP32 camera, REYAX LoRa module, and a Sonoff Zigbee protocol analyzer for communications between the user and the system. The ESP32 camera has a Wi-Fi and Bluetooth integrated microcontroller, the LoRa module helps send and receive data over long distances, and the protocol analyzer uses Zigbee to connect end systems to other end systems. The camera's live feed can be seen on the Samsung tablet display or on the user's smart phone. Our design utilizes renewable energy via a solar panel.

Overview of Code Developed to Support Engineering Analysis

```
camera_config_t config;
config.ledc_channel = LEDC_CHANNEL_0;
config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = Y2_GPIO_NUM;
config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;

// if PSRAM IC present, init with UXGA resolution and higher JPEG quality
// for larger pre-allocated frame buffer.
if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
} else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
}

#include "esp_camera.h"
#include <WiFi.h>

//
// WARNING!!! PSRAM IC required for UXGA resolution and high JPEG quality
// Ensure ESP32 Wrover Module or other board with PSRAM is selected
// Partial images will be transmitted if image exceeds buffer size
//

// Select camera model
// #define CAMERA_MODEL_WROVER_KIT // Has PSRAM
// #define CAMERA_MODEL_ESP_EYE // Has PSRAM
// #define CAMERA_MODEL_M5STACK_PSRAM // Has PSRAM
// #define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version B Has PSRAM
// #define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM
// #define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM
// #define CAMERA_MODEL_M5STACK_UNITCAM // No PSRAM
// #define CAMERA_MODEL_T_TINKER // Has PSRAM
// #define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM

#include "camera_pins.h"

const char* ssid = "PAPI";
const char* password = "omojesus";

void startCameraServer();

void setup() {
    Serial.begin(115200);
    Serial.setDebugOutput(true);
    Serial.println();

    config.fb_count = 1;
}

#if defined(CAMERA_MODEL_ESP_EYE)
pinMode(13, INPUT_PULLUP);
pinMode(14, INPUT_PULLUP);
#endif

// camera init
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
}

sensor_t * s = esp_camera_sensor_get();
// initial sensors are flipped vertically and colors are a bit saturated
if (s->id.PID == OV3660_PID) {
    s->set_vflip(s, 1); // flip it back
    s->set_brightness(s, 1); // up the brightness just a bit
    s->set_saturation(s, -2); // lower the saturation
}
// drop down frame size for higher initial frame rate
s->set_framesize(s, FRAMESIZE_QVGA);

#if defined(CAMERA_MODEL_M5STACK_WIDE) || defined(CAMERA_MODEL_M5STACK_ESP32CAM)
s->set_vflip(s, 1);
s->set_hmirror(s, 1);
#endif

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
```

```

    if (s->id.PID == OV3660_PID) {
        s->set_vflip(s, 1); // flip it back
        s->set_brightness(s, 1); // up the brightness just a bit
        s->set_saturation(s, -2); // lower the saturation
    }
    // drop down frame size for higher initial frame rate
    s->set_framesize(s, FRAMESIZE_QVGA);

    #if defined(CAMERA_MODEL_M5STACK_WIDE) || defined(CAMERA_MODEL_M5STACK_ESP32CAM)
        s->set_vflip(s, 1);
        s->set_hmirror(s, 1);
    #endif

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");

    startCameraServer();

    Serial.print("Camera Ready! Use 'http://");
    Serial.print(WiFi.localIP());
    Serial.println("' to connect");
}

void loop() {
    // put your main code here, to run repeatedly:
    delay(10000);
}

```

Figure 10 – Camera Codes



#Include allows us to access the needed built in Wi-Fi library on Arduino for our ESP32 device. In addition, it exposes the functions to initialize the ESP32 camera and get pictures from it. The #define statement allow us to define the specific board we are using for our alarm system. As you can see the ssid and password section allowed us to connect to a Wi-Fi network name and to have access to that network name via a password. This is done by defining variables to hold both Wi-Fi credentials. Serial.begin allows us to open up a serial connection to output results from our developed code. This section allows us to establish all our configuration pins and additional features of the camera such frame size and jpeg quality.

Overview of Code Developed to Support Engineering Analysis (continued)

```
26 captive_portal:
27
28 # LED light OUTPUT
29 ▾ output:
30 ▾ - platform: gpio
31   id: redLED
32 ▾ pin:
33   number: 2
34 # LED Settings
35 ▾ light:
36   platform: binary
37   name: "Red LED"
38   output: redLED
39 ▾ effects:
40 ▾ - strobe:
41   name: blink_fast
42 ▾   colors:
43 ▾     - state: true
44     duration: 0.25s
45 ▾     - state: false
46     duration: 0.25s
47
48 #####
49
50 # BUZZER OUTPUT
51 ▾ switch:
52 ▾ - platform: gpio
53   id: beep
54 ▾ pin:
55   number: 12
56 ▾ # Buzzer Settings, with a piezo buzzer, beep 6 times
57 ▾ - platform: template
58   name: alert
59 ▾ turn_on_action:
60 ▾   - switch.turn_on: beep
61   - delay: 500ms
62 ▾   - switch.turn_off: beep
63   - delay: 60ms
64 ▾   - switch.turn_on: beep
65   - delay: 500ms
66 ▾   - switch.turn_off: beep
67   - delay: 60ms
68 ▾   - switch.turn_on: beep
69   - delay: 500ms
70 ▾   - switch.turn_off: beep
```

Figure 11- LED and Buzzer Codes

Above in **Figure 11**, is the code we developed for the LED light and buzzer. We have were able to add an action to the LED light called blink_fast so that the LED light would blink rapidly when the alarm system is triggered. In addition, we were able to use the turn_on_action option to add a loop action for the buzzer to sound and beep.

Application of Engineering Codes and Standards used in Final Design

IEEE Std 1547-2020: IEEE Conformance Test Procedures for Equipment Interconnecting Energy Resources with Electric Power Systems and Associated Interfaces – This standard is recognized that a distributed energy resource can be a single device providing all required functions or an assembly of components, each having limited functions and capabilities.

IEEE/ASHRAE Guide for the Ventilation and Thermal Management of Batteries for Stationary Applications - Ventilation of stationary lithium-ion battery installations are\ critical to improving battery life.

NESC:

National Electrical and Safety Code is a United States standard of the safe installation, operation, and maintenance of electric power and communication utility systems including power substations, power and communication overhead lines, and power and communication underground lines. This safety measure was implemented to make sure that the electrical components of the are not exposed and create any harm.

Economic Analysis

Because our team carefully analyzed various brands of components for cost and reputation, we were able to complete our build without going over budget. We were able to get great deals on several parts. As you can see below in the parts list, we only spent \$392.44 which is way below our \$600 budget. The cost for each part is shown in **Table 3** below.

Brand	Item Name	Quantity	Total
ECO -WORTHY	Solar Panel Kit and Battery	1	\$108.00
YuHan	Power Supply and Control	1	\$29.99
ESP32	Camera kit	2	\$22.99
Aqara	Temperature Humidity Monitor and Door / Window Sensor	1	\$37.98
REYAX	Lora Module	1	\$19.50
Raspberry	Raspberry Pi 4	1	\$145.00
Samsung	Samsung Tablet	1	\$119.00
ELEGOO	Jumper wires	1	\$7.99
Adafruit	Passive Buzzer	1	\$1.50
Juried Engineering	LEDs	1	\$8.49
Total:			\$ 392.44

Table 3 – Parts Order list

Final Design

Security Alarm System

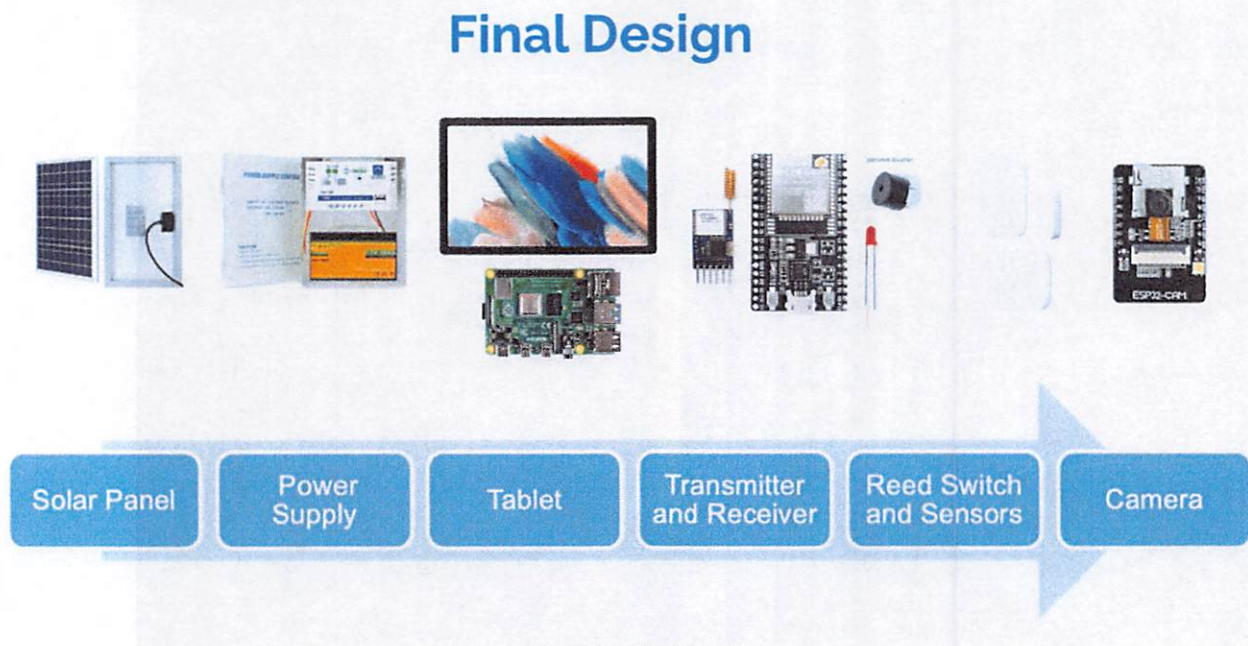


Figure 12 – Final Design

Above in **Figure 12** above is our final design, we made a couple changes from our initial preliminary design to our final design. As you can see, we decided to replace our LCD screen with a Samsung tablet. In addition, we removed that PIR sensor that was combined with our ESP32 Camera. Overall, our power supply stayed the same from our preliminary design. In this design we decided to use one ESP32 camera as a transmitter and receiver instead of two microcontrollers for both functions. The LED light will serve as a visual alert that the system has triggered via sensor detection. The buzzer will serve as an audible alert that the system has been triggered. The ESP32 camera microcontroller comes with HD quality capability for live feed footage. Our sensors utilize a wireless protocol called Zigbee. These sensors and reed switch (door/window sensor) will be integrated onto the Home Assistant platform for home automation. The LoRa module will be used for long range communication as stated previously in the preliminary design section of this report.

ESP32 Camera

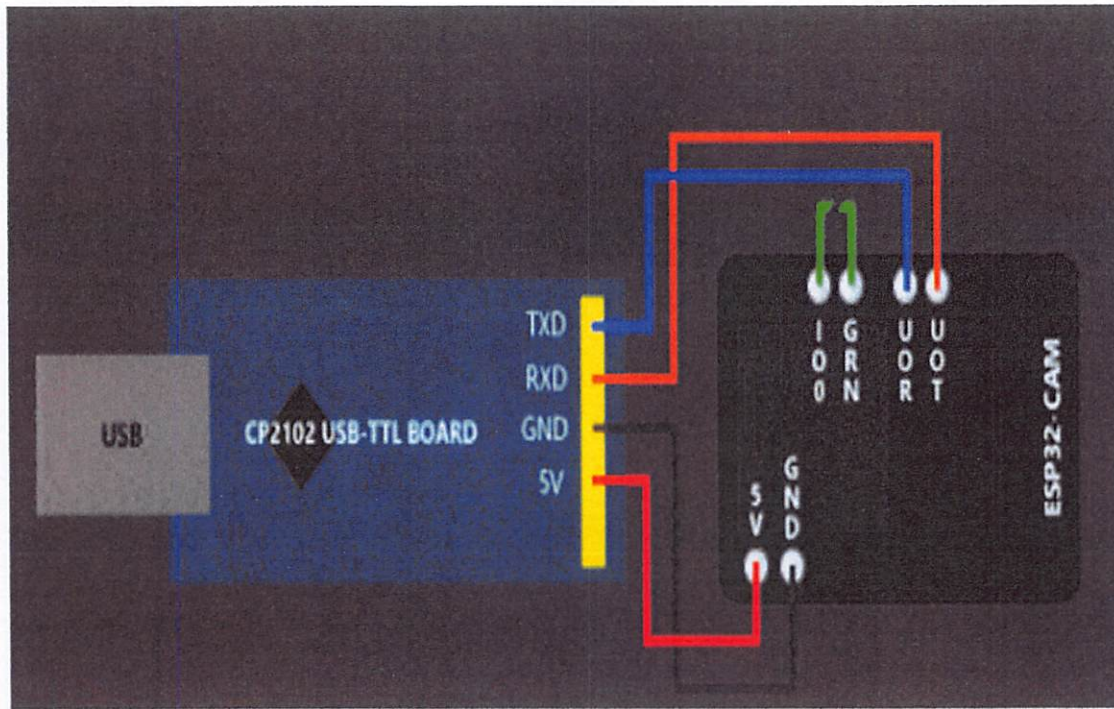


Figure 13 – Camera

Above in **Figure 13** is a schematic of our camera build. On the left is an Arduino Uno microcontroller that we wired to the ESP32 Camera microcontroller via several pin connections. The ground pin (GND) serves as the reference point for all signals within the system and measures voltage within the system. The UoR pin is the pin used to receive data. The UoT pin is used for transmitting data. The 5V pin supplies power to the microcontroller. After completing all wiring, we plugged the Arduino Uno board wired to the ESP32 Camera via a USB.

```
COM5
...
est:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
config: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:1324
no 0 tail 12 room 4
load:0x40078000,len:13480
no 0 tail 12 room 4
load:0x40080400,len:3604
entry 0x40080510

...
WiFi connected
Camera Ready! Use 'http://10.124.15.47' to connect
```

Figure 14 – Serial monitor with IP address

After completing our build of our camera and uploading the code to the ESP32 Camera we were able to receive an IP address (<http://10.124.15.47>) to access live feed and integrate our ESP32 Camera onto Home Assistant.

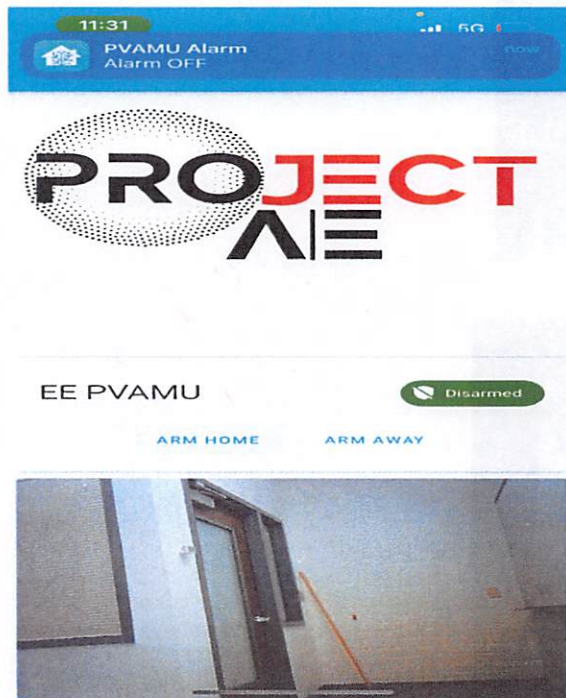


Figure 15 – Camera footage on Home Assistant platform

In **Figure 15** above, we were able to get the camera footage to display on the Home Assistant app using the IP address we received from the serial monitor on Arduino.

Below are images of the breadboard build, power supply build, and the completed build of the security alarm system. The breadboard build includes a passive buzzer, LoRa module, resistors, and an ESP32 microcontroller with the camera attached or inserted on top of the microcontroller.

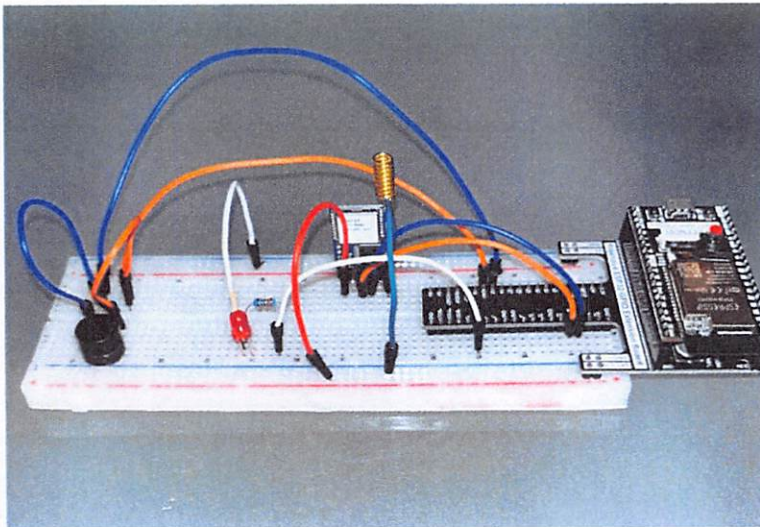


Figure 16 – Breadboard Build

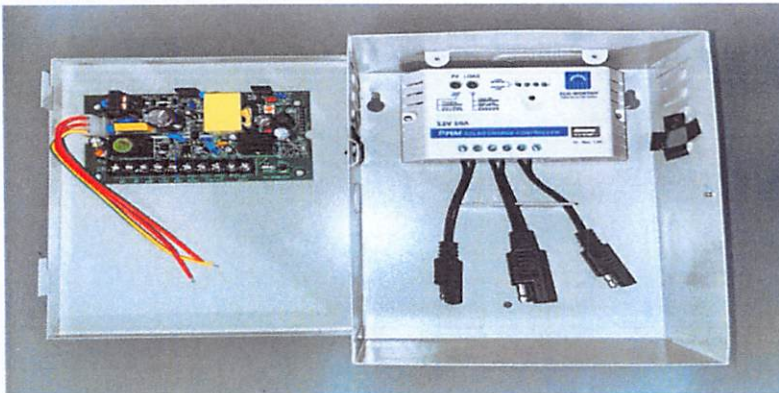


Figure 17 – Power Supply

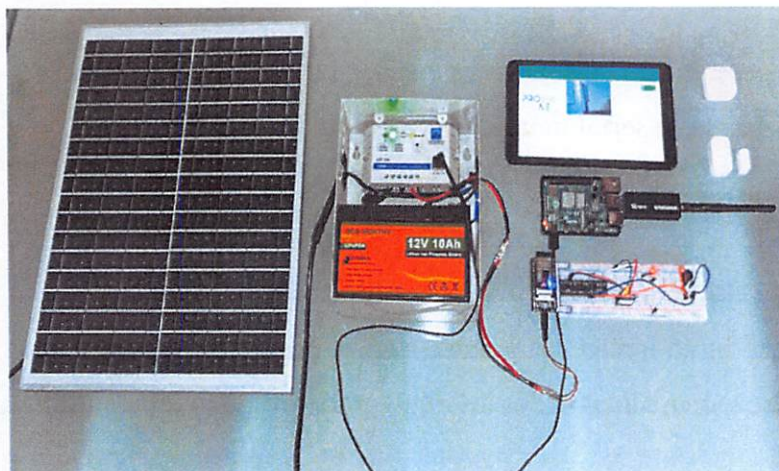


Figure 18 – Completed Final Build of Security Alarm System

RESULTS AND DISCUSSION

Summary of Goals and Objectives

The main objective of our project was to solve burglary cases in homes and offices by designing a functioning security alarm system that is reliable and affordable. The design also uses Home Assistant and Zigbee Alliance to unify devices and brands customers love to use. The goal of our team was to make our low-cost design function efficiently during power outages using batteries.

Summary of Constraints and Codes met by your Design

As far as constraints, a constraint of our design was trying to stay under a \$600 budget while building a robust smart security alarm system. Most smart systems with smart features can be expensive but due to our thorough research of components we were able to stay significantly under our budget. Another constraint was EMP resistance within our system, unfortunately we were not able to purchase an EMP resistant box for our system due to time constraint towards the end of the term for this project.

The following codes were used to meet the final design:

- IEEE 7.8 – Code of Ethics:
 - 1) To hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices.
 - 5) To seek, accept, and offer honest criticism of technical work.
- IEEE 7.1: U.S. Engineer Registration
 - This code informs us that we must professionally recognize the safety, welfare, and health of the public by ensuring our design does not harm any customers rather than help them.
- IEEE/ASHRAE Guide for the Ventilation and Thermal Management of Batteries for Stationary Applications
 - Some of the components of our design are in a box and ventilation and thermal management is needed to prevent damage to system or operator.
- IEEE 7.6: Protection of the Public
 - Our design must function properly to protect customers and/or their property.

Abiding by these codes, we were able to successfully complete our project. We kept in mind the IEEE 7.8 – Code of Ethics most importantly, we built our design to keep in mind the safety, health, and welfare of our customers. We touched on sustainability by building a system that uses renewable energy (solar) and we were able to build a system that allows you to reuse any

previous components that you have used in a previously owned security alarm system instead of throwing away those parts. Home Assistant allows you to integrate almost any alarm system component into your system.

Discussion of Results and Other Use Cases

Discuss our results...

CONCLUSIONS AND RECOMMENDATIONS

The development of our security alarm system is a solution that protects homes and businesses, thus reducing the price of other companies' expensive alarm systems. The final design was achieved through a few iterations of trial-and-error and could not have been successful completed without assistance.

As a recommendation, our team suggests selecting a security alarm system that provides the best protection at an affordable price. We also suggest an easy-to-setup security alarm system that brings comfort to homes and offices.

LIST OF REFERENCES

- B. McAfee, "How to keep your home safe during a power outage: Crime prevention," *CPSS*, 03-Jul-2017. [Online]. Available: <https://www.cpss.net/about/blog/2015/08/how-to-keep-your-home-safe-during-a-power-outage/> . [Accessed: 01-May-2022].
- "The Cybersecurity Workforce Gap," *The Cybersecurity Workforce Gap | Center for Strategic and International Studies*, 27-Apr-2022. [Online]. Available: <https://www.csis.org/analysis/cybersecurity-workforce-gap> . [Accessed: 01-May-2022].
- A. Lauren, "How a power outage puts your home security at risk," *Reader's Digest*, 11-Aug-2021. [Online]. Available: <https://www.rd.com/article/power-outage-home-security/> . [Accessed: 01-May-2022].
- "Home & Business Security Systems in Jacksonville, FL," *SafeTouch*. [Online]. Available: <https://safetouch.com/security-during-a-power-outage/> . [Accessed: 01-May-2022].
- Carmen, "Security system creativity for remote off-the-grid living," *Off The Grid News*, 24-Oct-2019. [Online]. Available: <https://www.offthegridnews.com/home-defense/off-grid-security-system/> . [Accessed: 01-May-2022].
- EPS Security, "9 important home burglary statistics," *EPS Security*, 09-Sep-2019. [Online]. Available: <https://www.epssecurity.com/news/home-security/9-important-home-burglary-statistics/> . [Accessed: 01-May-2022].

References (continued)

- Dutta, Pallavi, et al. "5 Major Reasons for 'Why Is Cyber Security Important?'" *Security Boulevard*, 13 Apr. 2021, <https://securityboulevard.com/2021/04/5-major-reasons-for-why-is-cyber-security-important/>.
- by, Written, and Katherine Torres. "8 Benefits to Owning a Home Security System." *SafeWise*, 22 Apr. 2021, <https://www.safewise.com/blog/top-8-reasons-get-home-security-system/>.
- "Cybersecurity for Physical Security Systems." *UL*, <https://www.ul.com/services/cybersecurity-physical-security-systems>
- Tatro, Kaitlyn. "Light Energy: Our Wasted Resource." *Consilience*, no. 22, Columbia University, 2020, pp. 65–72, <https://www.jstor.org/stable/26924963>.
- CSA. 2022. *Matter | Smart Home Device Solution*. [online] Available at: <<https://buildwithmatter.com>> [Accessed 1 March 2022].