Object Detection Drone

Senior Design: Group Three

Jazmine Roman

Electrical Engineering

Kevin Nilsen

Photonic Science & Engineering

Derek Murdza

Computer Engineering

Cannen Carpenter

Computer Engineering

Table of Contents

1 Executive Summary

In recent years, a lot of work has been done in the area of artificial intelligence and quadcopter drone development. Neural networks and AI training has piqued a lot of interest from researchers due to its near limitless applications in human lives. A primary goal is to make the machines people deal with every day more accessible by facilitating a natural method of communication between humans and computers. This technology can be linked to autonomous drone technology. The idea of an unmanned drone began in the latter half of the 20th century. Drones have developed greatly over the past several decades. Today the aerial robots find many uses in areas like environmental scanning, recreation, and military operations. The group wishes to combine these two impressive technologies together into an object detection drone that is able to use object detection and deep learning to direct a quadcopter drone through an environment.

The need for this type of drone can be specified by many different disciplines such as military use, where weapons or enemies can be detected while providing the low-visibility, covert components of a drone. Recently, drone research is looking at package delivery operations for quick online orders and first response. For example, if someone near you is having a cardiac arrest, the idea is that you can call an emergency and the emergency will send a drone with a cardiac arrest machine for you to try to resuscitate while the first response is on the way. These more commercial and civilian operations are currently not legalized as there are many air restrictions due to airport zones. Having these capabilities can greatly decrease time needed to manually use electronics to find objects, which in the end increases productivity and efficiency via automation. While also increasing accuracy.

If more time were given to the project, the drone could expand its capabilities from simple object detection to full autonomous navigation. Our drone has the capabilities to fly and detect safe landing zones and objects that are seen as obstacles, as pre-determined by machine learning software. Our drone has the standard design of a quadcopter and the focal point of this project is the machine learning software for its possible future vision navigation. Optical design is also a major focus as this is how the drone is communicating with the machine learning software to determine placement and what are obstacles. Machine learning is a fairly new concept within the last decade and has quickly become popular. Machine learning is the basis of artificial intelligence. Artificial intelligence is developed with a neural network mapping within the machine learning code. This neural network allows for the machine's code to learn according to categories that we will feed the machine via its optical design.

Using many different components, the drone is constructed to meet the engineering specifications and constraints provided within this document and shall follow the rubric of what is expected for the Senior Design showcase. The audience for this project is the advisors and the volunteer judges that are

evaluating what is presented within both the demonstration as well as the presentation during the showcase. It should be recognized by the professors that our project was faculty sponsored but due to a family emergency, this faculty is no longer with us. We will no longer have the mentor guidance that the team was hoping for and will try our best to execute the highly advanced software algorithms and robot building to provide a proper demonstration.

Prior to the presentation, the finalized drone shall be tested within a safe area to ensure full functionality and to ensure proper training via software. Given that this project utilizes many engineering disciplines such as computer, electrical, and photonics, the focus of study within the group is very well suited to the task of completing this project. There is a strong learning curve that comes with the process of both design and implementation, but the goal of the Senior Design course is to provide a real-world application to what can be expected in the Engineering field.

The following sections in this document detail the team's planning, design, and decision making process of the creation of this device. Notable sections include a detailed description of the components used, while also providing a comparison via decision matrices, a general overview of the testing environment that is used to ensure functionality, and also a specific plan that is followed to meet the group's goals by notable deadlines. It is important to note again that the main focus of our project is artificial intelligence, machine learning and neural networks. The drone is a standard quadcopter drone and there is no innovation in the actual build of the drone. It was also important to mention that our prior faculty sponsor stated that if our hand built drone does not fly, we may purchase a fully functional drone to implement the machine learning.

The team will do its best to build the drone from a drone kit as quickly and efficiently as possible, as much time is needed to train the drone with machine learning and artificial intelligence. The drone's software must be fed algorithms and images for at least one to two months prior to seeing any advancement within its machine learning. Therefore, our group's main focal point is training the machine to navigate with vision. Training is held with a lab given to us by the head of the electrical and computer engineering department at the university. The team will safely set up a training zone (explained in more detail below) to safely train the drone with the machine learning algorithms. The team will also seek mentorship and guidance from current machine learning and artificial intelligence professors at the university as this is a new concept and the students on the team will need guidance for strong execution.

Our team goes into great detail of the software and hardware components, prior research and current technologies below. This document has helped us determine what is feasible and what will not be feasible given the time constraints. Please continue to read and provide feedback on specifications where seen necessary. This document is crucial to successfully implementing both the hardware and software components of the drone, as well as tracking the process all the way to the showcase.

2 Project Description

2.1 Purpose

The purpose of this project is to build a lightweight manually controlled drone that can interpret software commands, translate them into instructions that the system can carry out to maneuver around a specified area while using a sensor system to detect obstacles and safe landing zones, unmanned. The drone will navigate by visual navigation and without a radio frequency controller or a person controlling the drone. The drone will determine which objects within the area to fly around, fly to, as well as rotate, change altitude, and safely land at ground level at a predetermined landing zone. These decisions are made by the implemented machine learning software and made possible through an artificial neural network.

The purpose of the team is to train the flying machine on its surroundings, object obstacles, and safe landing zones. Safe landing zones are marked with a black X on a white background. The obstacles that are introduced to the drone are stabilized balloons. Figure 2.1 shows a brief example of how the drone will recognize and analyze its environment, and Figure 2.2 demonstrates the user is able to directly interact with the drone. Our paper will dig deeper into the different machine learning algorithms that can be used and compare them effectively to determine which would be best for the mechanical and electrical components that the machine is built with.

The software will give the drone specific directional commands, and the drone will carry them out, occasionally asking for feedback whenever it runs into an issue. The system will facilitate a steady interaction between the software and the machine. The group will need to construct their own drone, and integrate the LiDAR and imaging systems directly with the artificial intelligence (AI) they develop. Several algorithms and machine learning software will either be open source or provided by university mentors (professors, phd students, etc.). The machine learning capabilities must be high to perform the algorithms we wish it to perform, therefore the hardware components of the quadcopter should be of high efficiency and dependability to be able to perform such a high intelligent level task, un-manned.

As the drone becomes constructed, different training methods are introduced in order to interpret machine learning commands and establish objects and its specific locations using a sensing operator. This document will describe the goals, deadlines, methods, and part selection that is crucial for the success of the project as a whole. The sponsor was to provide basic essentials such as parts needed in which the group shall decide which components are most suitable for not only the drone but for all other components. Now the electrical and computer engineering department will provide funding, guidance and support for our project.
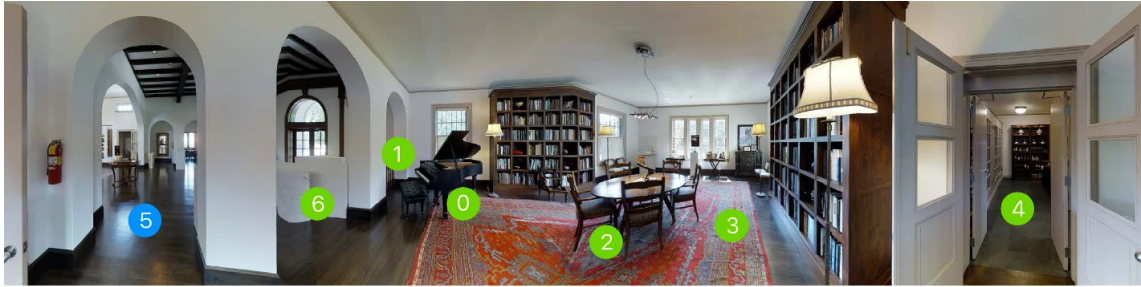
Object-Aware (OA) Module: walk through the first doorway out the three - the one all the way to the left , walk straight through the doorway directly across from in , in front of the mirror . turn right , and stop before the long carpet .

Action-Aware (AA) Module: walk through the first doorway out the three - the one all the way to the left , walk straight through the doorway directly across from in , in front of the mirror . turn right , and stop before the long carpet .

Envdrop: walk through the first doorway out the three - the one all the way to the left , walk straight through the doorway directly across from in , in front of the mirror . turn right , and stop before the long carpet .

Target Action: 5   OA Module Prediction: 5   AA Module Prediction: 1   Final Prediction: 5   Envdrop Prediction: 4

Figure 2.1: Example of logic for processing instructions from "Object-and-action aware model for visual navigation.
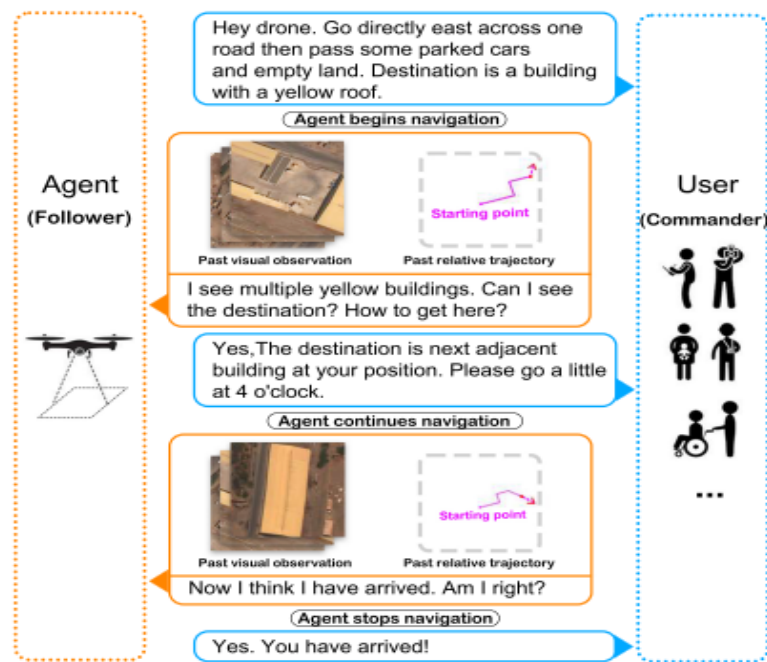

Figure 2.2: An example of a navigation system. A person can give instructions and the drone will ask questions to machine learning software for further guidance if necessary.

2.2 Motivation

We have shown interest in this project as a group because of the growing field of machine learning. Machine learning has been shown as an impressive and efficient way to train AI to complete tasks of varying difficulties. Due to the complexities of directly coding object detection into a computer software program, it is not a very efficient process, and it often requires more time and effort than one gets out of it. The group has learned that machine learning algorithms can be used to, "train," a program to do a certain objective. Through many trials, a program can more or less, "code itself," using different connections and methods to reach a goal, like identifying a piece of furniture or vocal pattern. For every iteration, the machine learning software may have many different programs at once and test them for accuracy. The program that is the, "best," according to the inputs one gives it is saved. The software then uses this best program, adapts it, and develops even more connections. Eventually, the program is able to do the specific task it was assigned very well. The group saw this project as a great opportunity that would challenge us to learn new skills and collaborate with a diverse group to attain a satisfactory response from the sponsor.

The group had to choose a mode of transportation for the system. The sponsor gave this group a choice between a car and a quadcopter drone. A sample of what the final product of a car might do is shown in Figure 2.3. This includes a full scanned map of the environment. A drone would not have such a detailed map and would need to navigate around fewer objects. Despite the drone being more complicated to construct and there being the possibility of more government regulations being a hindrance, the group chose to make the quadcopter. It should be noted that a drone requires far more time for building, testing, and adjusting than a car would, leaving the group less time to use the drone to train the AI for navigation. Since the time is only limited to two semesters, the sponsor recommended that the navigation requirements were reduced. Instead of having to navigate an entire room or hallway with a large number of potential objects and obstacles to identify, the group could instead focus on navigation around several specific objects that they choose.

The sponsor also has provided more resources for the drone navigation as opposed to the car. Algorithms for the drone movement are to be provided, as some experimentation for the drone training has already been completed by researchers under the sponsor's watch. These readily available resources are a great asset moving forward in our own experimentation and research into how we are producing our product. We believe this to be a big reason for choosing the drone over the car.

2.3 Basic Goals

The bare minimum for the project to be considered "completed" includes several specific goals. The primary goal of this robot is to navigate an environment through natural vocal instructions. According to the sponsor, the AI

must be able to recognize at least three predetermined objects placed in random spots, and move the drone around them according to human instructions. This is done using the AI that was developed to recognize human voice and associate it with specific objects and actions.

The drone must be able to hover in a single position in calm weather and move in several directions without human intervention (a handheld controller, strings, or other external control mechanism). The group should add mechanisms like an altimeter, gyroscope, and propeller thrust controls in conjunction with the sensing systems to ensure it can remain stable.

The group should create an object detection system that can determine if the drone is getting too close to an object. It should be at least able to accurately detect a white block 1 meter away in daylight. It should then send a signal to the drone's computer and advise it to stop moving. This will prevent damage to the drone and the objects around it. Such a system is done by creating a circuit with an infrared LED (or laser diode) and an infrared photodiode. Incoming light will create a potential across the sensor, and after passing through an amplifier, it is tested against a certain threshold at which a warning signal is sent.

2.4 Advanced Goals

A goal that would greatly increase the quality of our project would be to produce high-level algorithms in regards to training the drone. An efficient training method will most-likely be produced in Pytorch and applied in a simulation to speed up training by magnitudes greater than what would be achievable through live-testing.

The drone has a series of time of flight sensors around the drone that can determine exactly how far away a nearby object is. Several of these detectors should be set up in at least four places around the drone to determine if any parts of the drone are coming close to hitting an object or person. The closer the drone measures the objects to be, the more the drone can prioritize the warning signal. These sensors should give precise distance readings by creating a circuit that can accurately measure the phase difference between the transmitted and reflected wave. From this information, the exact distance away can be determined.

2.5 Stretch Goals

In an ideal scenario, the drone would be able to analyze an environment, interpret oral inputs, and navigate to its final destination through all of the obstacles in under 20 seconds. Such speed may negatively affect the drone's ability to stay up. By making the code of the drone efficient enough with fast enough processors it could be possible.

Another advanced goal would be to have a sophisticated LiDAR detection system that can create a two dimensional map of the between 1 and 2 meters

around the drone. It could also be used to create a full two dimensional map of the testing environment before carrying out instructions, possibly speeding up the completion time. This could be achieved by advancing the indirect time of flight sensor technology mentioned previously into a rotatable subsystem that continuously collects data about nearby objects in all directions.

Finally, the last stretch goal is to obtain autonomous flight. This will be tested via Python scripts and simulations to show functionality. The drone will operate the same regardless of navigation method, but a more advanced method would be preferred.

2.6 Related works

Within military use, drones are known for keeping our country safe, acting as undetectable eyes. Machine learning on drones have been mainly used in governmental operations for war use. For example, in 2011 when the United States was at war with Pakistan and in search for Osama bin Laden, war drones played a key role in finding him. The main role of unmanned aerial vehicles in war is to act as a nearly undetectable eye in the sky, which can circle any targeted area for hours on end. In addition to the video and infrared cameras, radar and communication devices, warfare drones will also carry weapons but are mainly used for target and visual guidance. Drones in war are forensic intelligence machines, where they are used by intelligence units to record video and data over days or weeks to present to officials who can trace the whereabouts of known or suspected terrorists.

Recently, drones are now being used in government, not only in war but also by the police force. Visual navigation drone technology is currently being studied to track highway speeders, to keep civilian surveillance and to track down criminals on the most wanted list. Policemen are being trained on how to use these drones for everyday use. These uses include, search and rescue disaster response missions, crowd monitoring, traffic collision reconstruction, crime scene analysis and to investigate active shooter incidents. As of 2020, at least 1,578 state and local public safety agencies across the United States have disclosed having acquired drones. Of this, seventy percent were law enforcement entities. For local statistics, Florida Polk County Sheriff's office publicized that they have flown more than 750 drone missions and this helped arrest 31 suspects and have resulted in finding five missing people. This brings attention to the importance of our project and drone using vision navigation in the present and in the future.

From a civilian standpoint, vision navigation drones are mainly being used for entertainment purposes. Such as, take aerial video footage for movie scenes or aerial photos for art. There is a current civilian used vision navigation drone that will follow the user (owner) of the drone as the owner walks/runs. This allows for users to take footage of themselves hands free. Vision navigation drones are also being used in entertainment in augmented reality video gaming. This occurs where a handheld device, such as a smartphone, is connected with the vision

navigation drone via bluetooth. The drone will then prompt the user to explore its physical surroundings with an augmented reality displayed on the screen, allowing for its users to seamlessly integrate their physical surroundings with the virtual world.

There have been some senior design projects in the past that have utilized object detection algorithms. Most recently, an automatic pet feeder was created in Spring 2022 that detected LED lights on an animal's collar with a stationary camera. Not only will this project have a more sophisticated object detection algorithm that does not rely only on an emissive light source where the distance is less of a factor, but it will also be integrated directly on a flying system where the sizes of the targets in question are constantly changing and moving around. This design is able to work in a fluid environment, not just a static one.

There is also recent work in drones being used commercially for building inspections and power plant inspections. This allows for a more accurate reading and for concrete data to solve real world engineering problems within the commercial industry. For example, once a building is built and developed, there are companies with infrared drones that scan the buildings to determine air conduction unit leaks. This is beneficial to the owners of the buildings as well as the environment in using our resources the most efficiently as possible. Another example in commercial use is on wind farms. Wind turbines are at high altitudes with extremely large blades. These blades are exposed to the external environment and need to be regularly inspected to ensure reliability of the wind turbine. Technicians do need to climb outside of the turbine to inspect the blades, this poses a high risk for personnel. Therefore, wind farm owners are investing in vision navigation drones to take aerial footage of blades to inspect for cracks in the frame and general wear and tear.

2.7 Engineering Specifications

As with any engineering project, this project has its own requirements specifications. These requirements are what the team will use as a final goal for the completed system. Prior to the final presentation of the completed system, the team will choose several of these specifications to be the most important, called the core requirements. These core requirements are what the team, and judges, will use to determine whether the team was effective in making a successful product. It is these few goals that the team is consistently striving the hardest to achieve through the design, development, and integration processes.

The team will also use the rest of the specifications as a guide throughout the design and building processes. While they may not be directly judged on them during the final presentation, they will still do anything within reason to reach them. These requirements are still very important for the group to achieve. The team wishes to make the most sophisticated system possible within the time allowed, and meeting all of the requirements is essential to this goal.

The team believes that meeting each of these requirements to their fullest extent will not only create a safer and more reliable system that can consistently deliver results, but it will also encourage synergy throughout the entire build, and reinforce the drone's capabilities so that its core requirements are met with greater confidence. Achieving every goal and requirement is the group's ultimate goal for a successful system.

As introduced above, the object detection drone is a standard small, quadcopter drone that will recognize several target objects, and be able to manually move around the obstacles and land safly at a designated landing zone. It will also have a fully rotating LiDAR scanner to measure the distance to the environment and tell the object detection software to start running. Below is a table with basic specific engineering specifications we are hoping to achieve with our built drone and built machine learning algorithms.

Table 2.7.1: The specifications that is met once the drone is completed in terms of measurements, weight, capabilities, and actions:

1	The system should be able to fly manually in all 3 dimensions
2	The system should be able to hover in place
3	The distance sensors must measure distance of target objects up to at least 1 meter with a 0.1 m accuracy
5	The machine learning model must classify objects within 5.00m and have a confidence above 85%
6	The drone must be calibrated to pass pre-arm checks for safety purposes
7	The drone should only be flown in open indoor areas

2.8 Block diagrams

In order to effectively reach the goals of the project, the tasks had to be evenly split up among each of the group members according to their specific skill sets. We have Cannen and Derek who are computer engineering majors and focused on the machine learning aspect. Kevin who is an optics engineering major who is focused on the camera and laser for machine detection and Jazmine who is an electrical engineering major who is focused on current and previous technologies, part specifications, and building the drone. Each subset of blocks is labeled according to the timeline of the project that they should be a part of.

For instance, the first chart lists the specific hardware components of the drone, and which member is responsible for the given subset. The different components, like the optical system and drone flight calibration system, must work together in conjunction with the AI to efficiently move the drone through the testing environment.

The second chart details the different aspects of the software of the drone. This describes the different types of software input that the drone will receive from all of its subsystems, and how it will use that information and interpret it. Each relevant member is in charge of a specific aspect of the acquisition of the data, the artificial intelligence's interpretation of the data, and the relevant commands and instructions that the AI will send back out to the hardware of the drone to complete.

Finally, the third chart is the action plan diagram. The group found this necessary to add because they wished to have a clear diagram to follow that outlined all of the actions that needed to be taken for the project to be successful. This action plan diagram will assist with the flow of development. Only once certain tasks are completed can the different subsystems of the project come together and be fully integrated with one another.

2.8.1 Hardware of Object Detection Drone

The hardware of the drone needed a clear outline for its functionality. The group used the following chart to characterize and organize each segment of the hardware into parts that each team member would be in charge of. This diagram will help the group divide their labor and know which member to turn to if any segment of the drone was faulty. This diagram is shown in
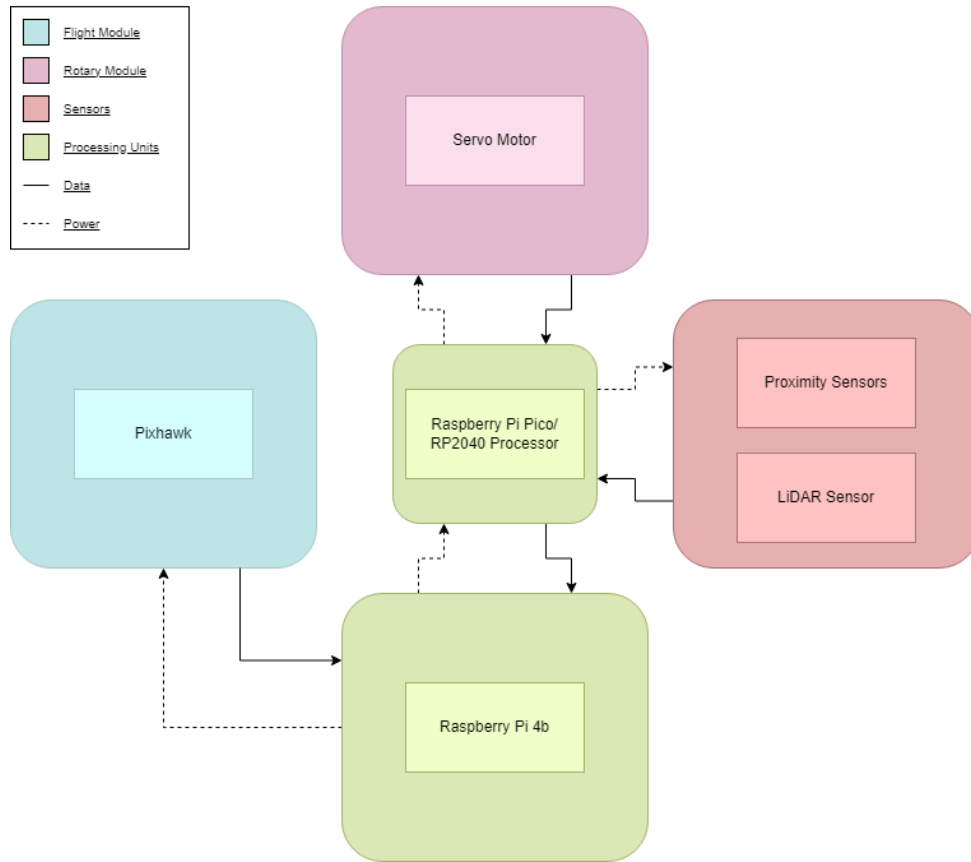
Figure 2.8.1: Hardware block diagram

2.9 Software of Object Detection Drone

The model below illustrates the steps we are taking in order to produce a trained drone. Drone input, displayed in orange, is provided via simulated camera and sensor feed. This data is then fed through our training process, highlighted in green, Our training process is a convolutional neural network built to classify images. In this portion of the project, we are giving images to look at and compare. Each image contains a label and bounding box representing what is to be identified within it. The CNN will make predictions according to the gradient in place, then adjust after being told what was correct. After many simulated training runs, and adjustments to the training gradients to get the CNN to produce a model that is as accurate as possible, we will then transfer the trained model to the real drone and do test runs in an actual course. The last section of the diagram is the section in green. This represents the transition from finalizing the CNN's training and applying it in a real-world state.
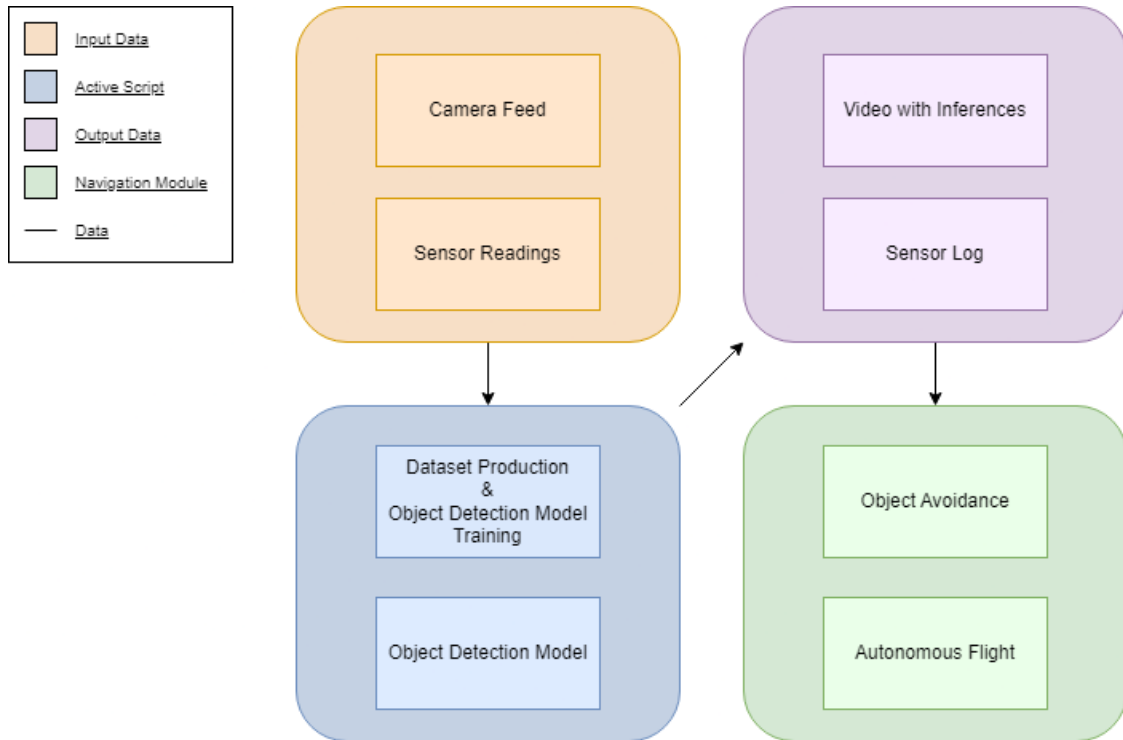
Figure 2.9.1: Software block diagram

2.10 Action Plan Diagram

The team decided to create an 'Action Plan Diagram.' This plot outlines the main jobs of each of the team members during every section of the project: research, building/testing, integration, and demonstration. It describes the general actions and segments of the system members are in charge of. This will help organize each member's responsibilities in the project. It is not as detailed in terms of functionality as the previous block diagrams, but the team decided it was necessary for them. It is less of a block diagram for components, and more of a block diagram for people. It is a guide for the development of the project as a whole and it organizes team members
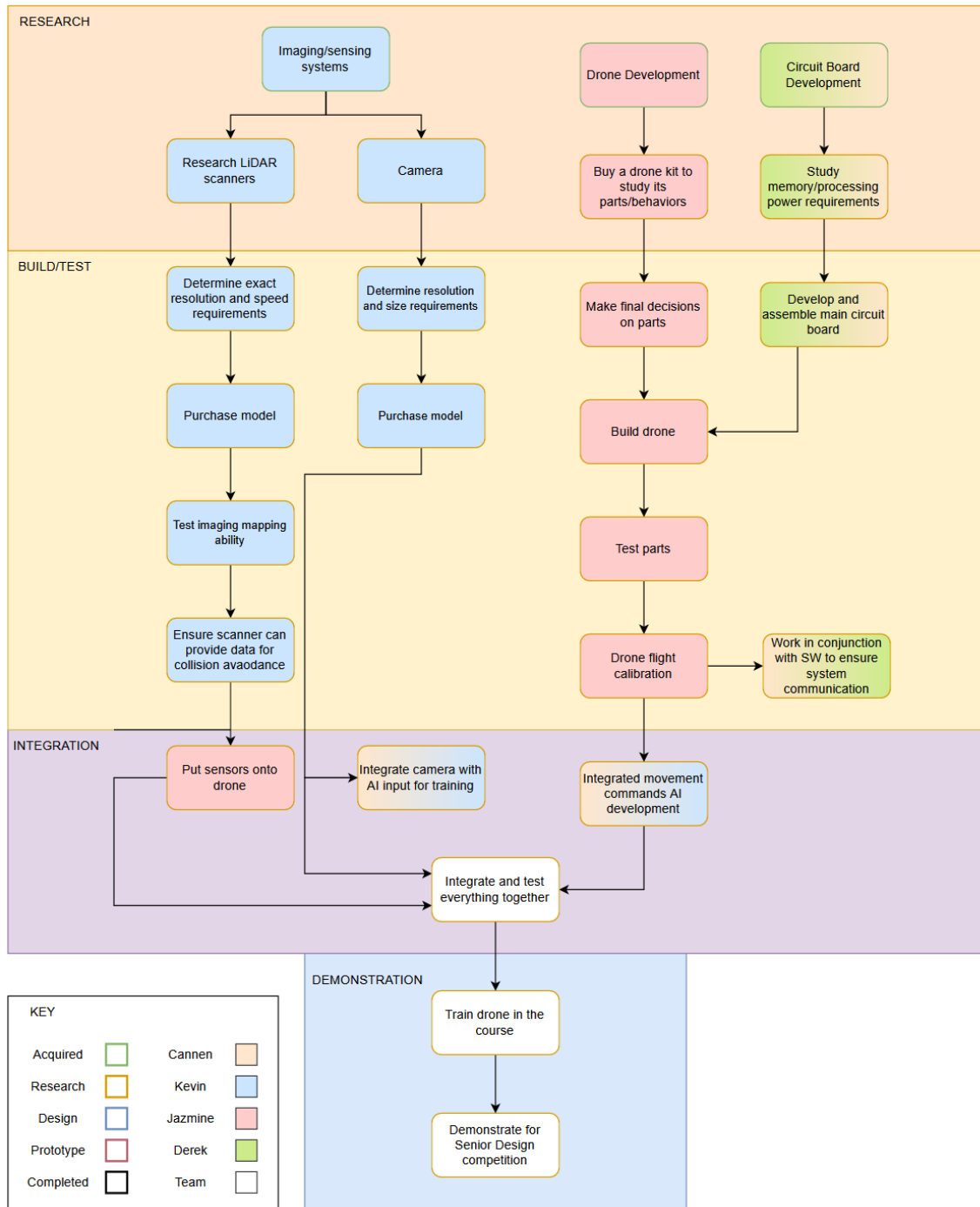
2.11 Training Reward System Diagram

In the future, if we are able to reach the navigation training system, we have chosen to utilize reinforced learning. This is basically a reward-based training system that influences the neural network to make more accurate

predictions. When the model makes a prediction, it is adjusted based on if their prediction resulted in a "reward." Below is a diagram describing the process of training the drone. Rewards can stem from any measurable source. Performing the task correctly is only one. When factoring in parameters such as collision avoidance, time constraints, and other testing factors as the diagram displays, training efficiency can be improved and specialized:
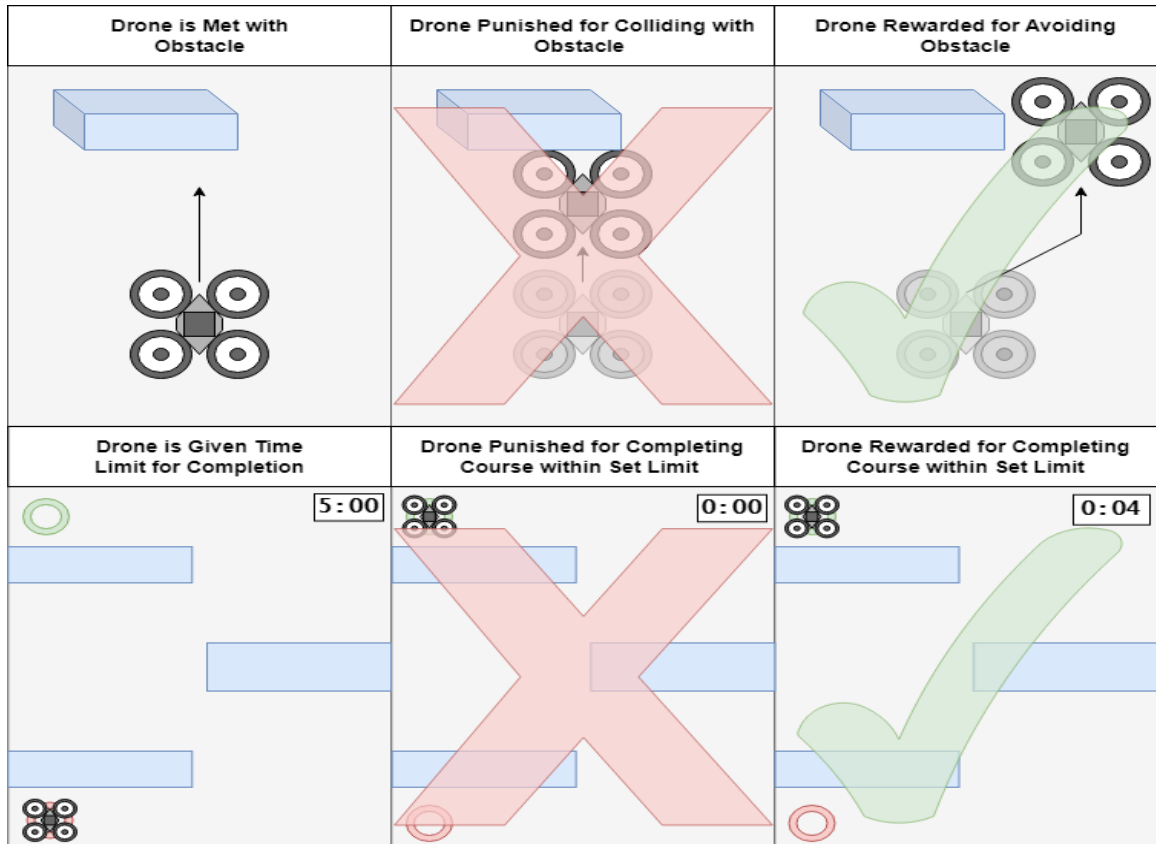
Figure 2.11.1: Drone navigation training process

3 Research and Part Selection

Today's drone capabilities are quite fascinating and the technologies that have been implemented with drones have revolutionized the flight industry completely. The devices take amazing aerial images and recently have been integrated with augmented video game playing. With the ability to take aerial imaging, drones have been used for a wide variety of tasks from entertainment to rescue and search missions. They have the ability to collect data and inspect areas that could be dangerous to humans. Drones can also be used for medical and emergency delivery during catastrophes by delivering much needed aid to areas that are traditionally hard to reach with land based vehicles or large helicopters.

Drones have provided footage to take humans where most humans cannot go and have made many geological discoveries, such as finding tribal markings on the top of large mountains or canyons. On the other hand, this same technology can be used to perform illegal activities like invasion of privacy, social disruptions and drug delivery. Therefore, a lot of research and technological advancements are occurring on drone design, safety and security.

Angular momentum is a very important physical quantity when building autonomous flying vehicles because it is a conserved quantity and the total angular momentum of the closed system remains constant. This is because the momentum has both a direction and a magnitude and both are conserved. Applications of the gyroscope are seen in gyrocompasses, replacing magnetic compasses to assist in stability and be used as a part of the inertial guidance system, in submarines as an inertial navigation systems and can be used to maintain direction in tunnel mining.

The servos is shortened for the term used, servomechanism, in control engineering which is an automatic device that uses error-sending negative feedback to correct the action of a mechanism. This term is used on systems where error control signals control mechanical position and speed. It is a powered mechanism producing motion at a higher level of energy that the input level, an example can be in the brakes and steering of cars where feedback is employed making the control automatic, hence automatic driving control or cruise control.

A question then presents itself: *What exactly is a drone or quadcopter*? From the development of small, radio controlled airplanes brought the interest to development of creating a stable flying robot that can be controlled by a radio frequency remote controller. To solve lateral flight, quadcopters depend on the four turning motors attached with blades on each end to give it thrust. Two on one side spin clockwise and the other two on the other side spin counter clockwise, this provides the possibility to spin, move forward and backward and side to side.

The four propellers at each end have a blend of pitch mechanisms and coaxial rotors. The pitch mechanism makes the drone agile and resistant to winds. The coaxial rotors allow for the two layers of the rotor to provide stability to the blades. These agile and steady flying robots are ideal for aerial and mobile vision. This design became popularized and led to quadcopters becoming the most common drone technology sold, making the drone market a $1.2 billion dollar market today.

Drone systems work with Global Position Systems (GPS) which allows the UAVs to be remotely supervised through a software embedded system working directly with GPS positioning modules that are built in. GPS utilizes two main standards for routing and positioning the unmanned aircraft. This is also called the code stage and the transporter stage. GPS is driven by satellite technology and does its best for accurate precision when the signals can be clearly sent to the device but what happens when GPS is unavailable? As GPS is lost indoors, underground, and in between buildings. For autonomous flying, drones cannot rely on inaccurate GPS coordinates due to electromagnetic wave disturbances from the environment.

Autonomous drones are currently trying to address this issue with vision navigation drones. Vision navigation allows the drone to operate and make its own decisions when GPS is unavailable. This is done by using the thermal feedback from the attached cameras to avoid obstacles while also identifying objects in its view such as trees, buildings, humans and much more.

Complete vision navigation GPS-denied systems for drones are currently used primarily for governmental purposes and are still undergoing extensive research. There are tech start up companies working on the optimal performance and safety measures for government technology as well as bringing this product to the market.

3.1 Research

This section will expand on technologies related to the project and how they are useful research pieces. It further examines the field of autonomous drones as it stands today. There has been a great deal of development throughout this area of study, with leaps in development in both the commercial sector and military sector.

How these drones fly and interact with the objects in their environment is influenced by the AI making them. This is why this section also has a heavy focus on the current state of artificial intelligence and neural networks. There is an increasing desire to lower the dependence on human controllers. People can often react too slowly to events, especially in times of high intensity and stress. So, there is naturally a desire to instead depend on artificial intelligence. In an ideal scenario, a mature AI could respond to instructions ro threat faster, and more efficiently than a human can. AI has also begun to prove itself to have more

creative abilities. Artificial problem solving intelligence is not far off from modern capabilities, and an autonomous drone system could benefit from it greatly.

The following sections compare the group's ideas and design with current methods and technology, and currently available products. They discuss in detail what these technologies are as well as how these technologies are going to be implemented with the final project in the spring.

3.1.2 Existing Similar Products

"Learning Vision-Based Flight in Drone Swarms by Imitation"[11]

Research study was submitted May 27th, 2019 and published August 14, 2019. The topic revolved around the way in which drones within a swarm detect each other. Typically, this is done by marking each drone so they can identify each other or each of their positions are broadcasted to one another so they can avoid one another. This study was conducted in hopes of creating a new method of collision detection amongst a decentralized drone swarm. There are similar paths that we are following to reach our goals. These are a few of those similarities:

3.1.2.1 Training

The team of researchers dove into the machine learning route, specifically imitation learning, which is a form of supervised learning. Our system is trained using reinforcement learning but there are still valuable lessons that can be learned from their studies.

3.1.2.2 Simulation

In their experimentation, they utilized simulations to create an easier to control environment. Simulation also makes it so one of the only limiting factors for learning speed is the hardware on which these calculations are being made. These simulations were done in a software named Gazebo in conjunction with the autopilot system PX4 Autopilot. Simulation is the best way to start our training without causing possible damage to the drone itself. Training will also be completed at a much faster rate. Gazebo may be our best choice as it is supported by PX4 and is recommended for "object-avoidance" and "computer vision", as stated in the PX4 User Guide.

3.1.2.3 Control System

Our drone will utilize PX4 as its control system. PX4 has a large directory of open-source code paired detailed tutorials that will most definitely work to your advantage. Less hassle with the control system will allow for more time actually training the drone to navigate through the obstacle course.

3.1.2.4 PyTorch:

The team made use of the machine learning framework PyTorch to create their convolutional neural network. We will also be using this framework to create our own neural network to train the drone. This will prove to be the most difficult portion of the software implementation, as there are several architectures and prediction models to use and then configure accordingly.

"Live Detection of Foreign Object Debris on Runways Detection using Drones and AI"[12]

This study goes into detail about the problem current detection systems have to deal with on airspace runways. "Foreign" objects and "debris" have the potential to cause runway disasters so there must be a way to tackle this problem. At the time of this papers' submission, the most often used method was mechanical detection methods, ie: various sensor technologies. Each of these technologies came with their own disadvantages. The authors of this paper state that with advancements in current AI and object detection, drones were proposed to be a valid way of avoiding disaster.

3.1.2.5 Machine Learning

The team utilized Microsoft Azure to fit both of their learning and object detection needs. They used Azure Custom Vision to develop their object recognition algorithm. This is a cloud service that allows those who do not need onsite machine learning or do not have someone who is knowledgeable in making a neural network. Microsoft makes it easy for anyone to implement computer vision models for image recognition, as their models can be exported directly onto a system to make predictions. We are building the convolutional neural network ourselves through PyTorch, but it would be helpful to explore the Microsoft Azure documentation for potential solutions.

3.1.2.6 Camera Clarity

In this study, two different cameras were used. One was specifically for training purposes, while the other was used for live testing. The "Mavic Air 1" that was used for training was chosen mainly due to its quality to cost/availability ratio. This camera could still support higher resolutions with decent frame rates and at the same time be replaced easily if damaged. The "Phantom 4 Pro" supports higher frame rates with more choices for aspect ratio, but was more expensive because of the higher quality video. If training with the lower end camera met expectations, it can be assumed that substituting with a higher-end camera would work even better. Their selection was highly dependent on the weather conditions and range that their drone would have to cover. In our case, these issues can be disregarded because the course is inside a wide open room with only a few obstacles. Our camera does not need to support resolutions as high as 4k nor have frame rates over 30 fps either. Our drone is moving at a comparatively slow speed and has time to calculate and make decisions. There

is no urgency to navigate the course like how they had to identify objects quickly to keep the runway safe.

"Improvement of Image Processing for a Collaborative Security Flight Control System with Multiple Drones"[10]

Research study was added to the archive on July 23rd, 2018. The study was done to improve automated drone flight control systems due to the rapid rise of the drone market. The concept shown was that of a "slave" drone tracing a "host" drone. Their problem was when the "host" drone reached max speeds, the observing drone could no longer recognize the "host" and trace it accurately. They already had a method in place but were looking to improve on the speed of it.

3.1.2.7 Image Recognition:

The method of image recognition was done by using OpenCV. OpenCV is another open source library dedicated to Computer Vision applications. This library is for the languages C++ and C but has expanded to Python as well with OpenCV-Python. They stated the processing time but did not go into the learning process. The purpose for this experiment was to improve the algorithm that was already in place, because the drones were moving too fast to be recognized. Fortunately this will not be a problem for our system, as there is no object with motion that will need to be identified. Processing rate per frame will not be a deciding factor in our case when we are not moving at speeds such as them (as high as 8 m/sec). Their solution was to calculate the optimal image size, as well as distance to trace the drone effectively. The graphed results display the extreme drop off of recognition rate for four different image sizes at each of their optimal follow distances. This may foreshadow issues we could have if we were to expand the dimensions of our training environment so it is something to consider.

3.1.2.8 Camera Clarity

The requirement for a high quality camera was evident by the problem they were having. Tracing a drone moving at 8m/sec would require a high frame rate and high resolution camera. They utilized a camera with a 4096 x 3072 resolution, higher than 4k. Again, the velocity of the drone is their cause for this requirement. A camera module like this would be excessive for what we are trying to achieve.

3.1.2.9 Objects

The objects needing to be identified in their studies are astronomically smaller than the obstacles in our course. A nail or a bolt can cause problems for vehicles on the runway so it is necessary for the very high resolutions, as pixel data is vital in that instance. Our objects within our testing environments are large and contrast heavily with their background. Camera resolution will just need to be a reasonable 720p or 1080p which is possible for our Raspberry Pi camera while

maintaining at least 30 fps. Initial testing in our project also only calls for pre-mapped object placement. This means that the drone will know it is there, it will just need to recognize and classify the obstacle in its way, rather than have no context to the environment.

3.1.2.10 Propeller Safety

"Safer and Longer-flying, Actively Tethered Drones Open Skies to More Uses"[10]

Research was done at the University of Cincinnati to create a drone with the goal to create a simple way of navigation by implementing a tether that is user-controlled. Tethered drones are often used for stationary monitoring, relying on a source of electricity on the ground to provide longer flight times. UC helped the Ohio Department of Transportation implement some of these systems to monitor traffic or construction projects where an eye in the sky provides a valuable perspective.

Tethered drones are connected to the ground to supply the energy required to allow for omnipresent surveillance over a limited area. Moreover, the tether allows for secure communication and transfer of information such as a video feed from the air vehicle to a ground station, making it resilient to cybersecurity threats.

For this project, rather than having a tether to prevent crashing since the user has control over the whereabouts of the drone, we will implement a propeller cage which, even in the event of a crash, will decrease any possible safety risks, although in the development stages this method could become a possibility due to its ease of control even with machine learning.

3.1.2.11 Live Testing

For their live testing environment, they streamed the video live to their database. This is probably most optimal because the drone does not react to the detection results in any way. The point of the drone is to relay what may be on the runway and that is it. There lies the difference between our problem and solution. Our drone is going to have to make decisions after recognizing what is in front of it. Streaming would still be plausible but onboard decision making would result in quicker course completion. Onboard processing ends up being more expensive, but reduces potential faults that may stem from failure to stream data between the drone and a processing unit.

3.2 Image Classification and Object Detection

Image classification is the process of scanning large sets of images and analyzing them to discover features that make them identifiable. This is done by first breaking each image into a grid composed of their pixel values. A low-level use of these values would be for edge detection. Object detection is the step

above image classification. This includes drawing conclusions as to where the object is within the picture itself.

3.2.1 Classification

Expanding upon the concept, classifying the outline of a common item such as a balloon would be manageable. If given a set of images after the edge detection process is done, some of which are balloons, some are animals, some are buildings, we as humans would be able to organize them into their own categories. Without color or context, the shape of a regular latex balloon is still very much identifiable.

A latex balloon is ovaloid with one end being larger than the other, and a string hangs below the smaller end. We can categorize these images based on our own knowledge learned from years and years of experiencing the world. Image classification is about applying our human logic in this same way. We want to teach our system to be comfortable in identifying objects by feeding it images, letting it pick them apart, then telling it what each image is. Then when images with similar results are fed in again, they can make guesses. This goes cycles and cycles to improve results.

3.2.2 Object Detection

Progressing past the idea of image classification/recognition, object detection is a more detailed process. Conceptually, an object detection model reads an image and makes a prediction as to what is contained within an image, as well as its whereabouts. This can include multiple predictions within one image. The imageset that the model would train on includes another label, turning this into a multi-class classification problem. This new label is composed of x-y coordinates paired with a height and width. The label acts as an indicator of where the identified object is located. Predictions after the model is trained would then be these coordinates with their associated object name.

3.3 Convolutional Neural Network (CNN)

The training, as discussed before, is done with a CNN. A neural network is a network of algorithms used to digest data in a way that is similar to how a human would. A single network consists of many nodes, each with their own input, weight (indicating level of importance), bias (indicating when the node should pass its output), and an output. The nodes act as different groups and the pathways between the nodes send the relevant information to the grouping in which the machine learning software believes the classification it belongs to.

For example, a CNN that we use generally within the public is within website login security. When a cache box is given for the website to determine you are not a robot and hacking the network, you need to identify particular objects within a frame. The objects that are chosen must fall into the specified

category in order to pass the cyber security algorithm. This online security is just a minor example of everyday usage of CNN and neural network algorithm use in everyday machine learning. In our project, the CNN is used for image classification, which will then aid in navigating through the course.

3.3.1 Layering Process

We will first need to implement the input layers for our CNN. This is done by introducing a very large dataset full of images. These layers are then to be fed through convolutional layers. Convolutional layers are made up of filters with 2-dimensional properties that are learned as the input is convolved with these layers. Convolution involves taking the dot product of sections of the inputted image with the convolutional filter.

An activation function is applied, creating an activation map in the process. Every activation map is then going to be combined to create the output. This output is fed to the next layer in the CNN is the pooling layer which uses a technique called "down sampling" which simplifies the learning parameters and lessens the computational load. The process starts all over with the node feeding this output to the next node as its input. An illustration of this is displayed below in Figure 3.2.
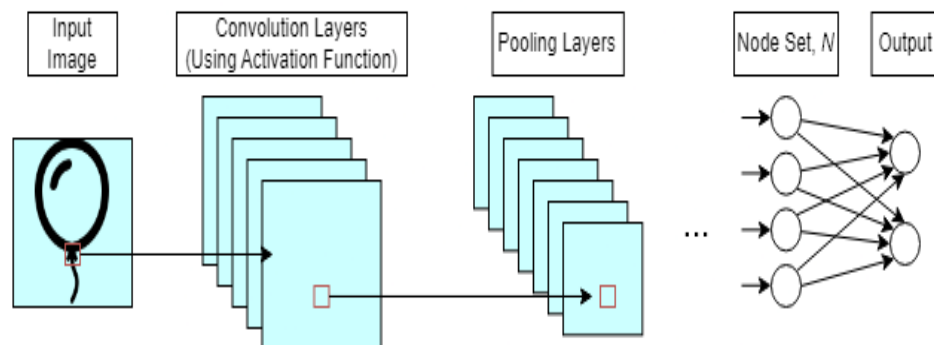
Fig 3.2: Convolutional Neural Network Diagram

Many take this idea of a convolutional neural network and produce their own architecture that fits their needs. We will expand upon several architectures that are open-source. The key take-away is that CNNs are mostly used for image classification. The convolutional layers just allow the network to distinguish between pixel properties in a way that other baseline neural networks can't.

3.3.2 Reinforcement Learning

The collective effort and knowledge learned from each of the nodes are combined to make a decision on how to react when met with a task. We

construct our CNN to digest a very large image-based dataset in order for it to pick up on the details that make our obstacles distinguishable from each other. Then when our models are used in conjunction with the drone simulation, the CNN can make accurate predictions on how to navigate appropriately.

At that point, the CNN is taking in live-video feed, analyzing each frame and making decisions that fast. All at the same time, we can reinforce its learning potential by signaling when it is correct with a "reward." This reward is essentially giving it the "correct" value in comparison to the prediction that it made. For example, when our CNN is being trained and is fed an image of a balloon, if it predicts that the image is a ball, then we can penalize it. The reinforced learning affects the CNN's ability to create its weights and learn when it is succeeding without having to intervene and configure the CNN after making a prediction.

3.4 Recurrent Neural Network (RNN)

A Recurrent Neural Network is another type of neural network. These neurons feedback between each other within the hidden layer indicated in Fig. 3.3. This allows them to hold all previous information that has flowed through them and current. This neural network will not be useful in our case though as it is typically used for language and speech recognition purposes. This makes sense with its ability to build up context within a sentence by learning how the placement and sequence of words are utilized to form a sentence with a specific meaning.

With the purpose of an RNN being defined, we can say that we will not be using one for our image classification. There is no possible way for the RNN to pick apart images in the way that the CNN can, so there is no reason to utilize it. If we were to train to understand speech for the drone commands, then I have no doubt that we would be able to make use of one. This was an original goal of ours but have scrapped that idea as it may make the overall task we are trying to perform unrealistic to acheive.
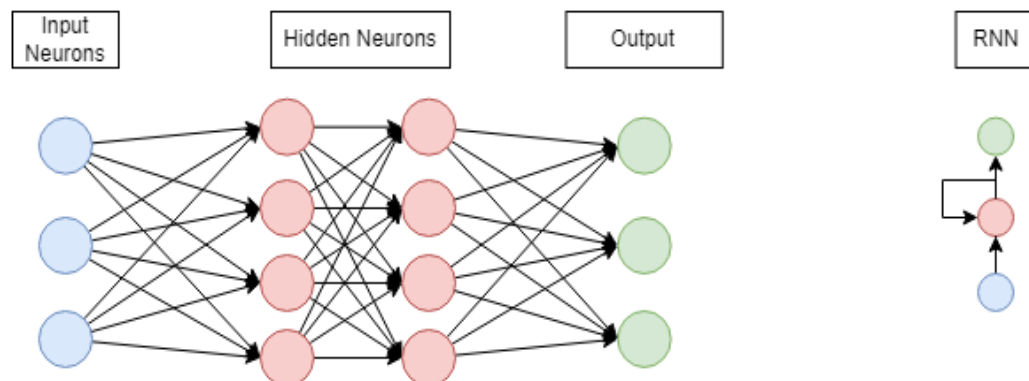

Fig 3.3: Recurrent Neural Network Diagram

3.5 Activation Functions

For the construction of the convolutional neural network, there is a decision to be made about the type of activation function that we is using. An activation function adds non-linearity to the CNN, allowing it to conform to more complex non-linear patterns. After the input is convolved, it is fed through an activation function that decides if the neuron is activated or not based on weights, biases and the type of function used. ReLu is one of the most popular activation functions today because of its ability to avoid the "vanishing gradient" problem. This problem describes the outcome that other activation functions have, such as sigmoid, when there are many layers. Sigmoid takes any input and compresses it into an outcome between 0 and 1, meaning the derivative becomes increasingly smaller as it passes through neurons.

3.5.1 ReLU

The Rectified Linear activation function, ReLU, utilizes the max function to produce its output. The equation for ReLU is provided below.

$$Output(x) \ = \ max(0, \ x),$$

Where x is the input value

This is a simple comparison between the value of the input and the number zero. If the input is positive, then the output equates directly to the input. If the input is negative, then output is zero. This makes it non-linear to negative input, and linear to positive input. ReLU is probably the most common activation function, because it is so simple and it doesn't have the problem of "vanishing" gradients that other non-linear activation functions have. We will most likely be using this activation function for its simplicity and performance.

3.5.2 LeakyReLU

LeakyReLU attempts to solve the problem of a "dead" neuron that occurs with the ReLU function. The equation for LeakyReLU is provided below.

$$Output(x) \ = \ max(0.01x, \ x)$$

As described previously, ReLU neurons can essentially become permanently inactive depending on the input. Rather than sealing a neuron to output 0, reducing the input to one-hundredth of its original value keeps the gradient intact, even though it may be small. Conceptually this makes sense, but often in execution it can fall short of expectations. This is because non-linearity is lost when multiplying by this constant. The dying neurons are potentially resolved, but there is loss of efficiency due to the introduction of more training. Graphing both ReLU and Leaky ReLU, we can see slight differences in their

output. YOLOv8 makes use of the Leaky ReLU model though, so we may need to make use of it.
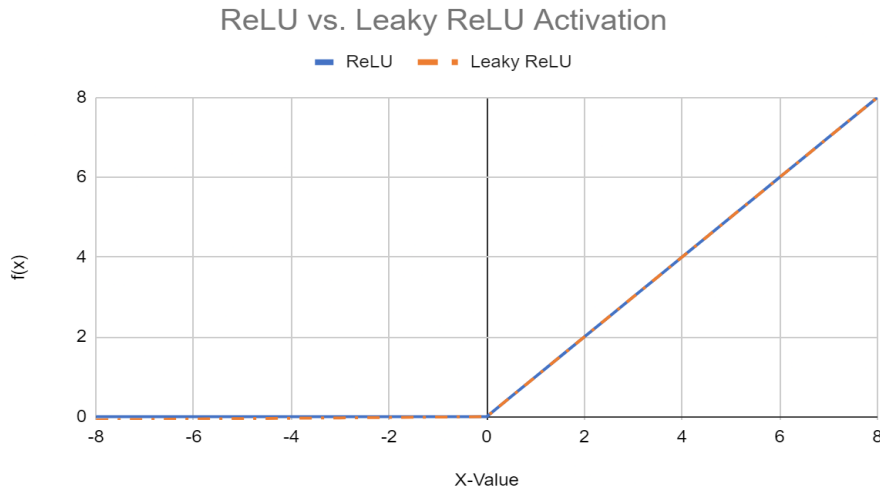
Figure 3.5.2: ReLU vs Leaky ReLU functions graphed

We can see the similarities very well between the two activation functions, but those slight differences don't seem to have much weight to them. Graphing the derivations at each point one would then be able to see the key difference between the two functions. This is shown below in Fig 15.5.
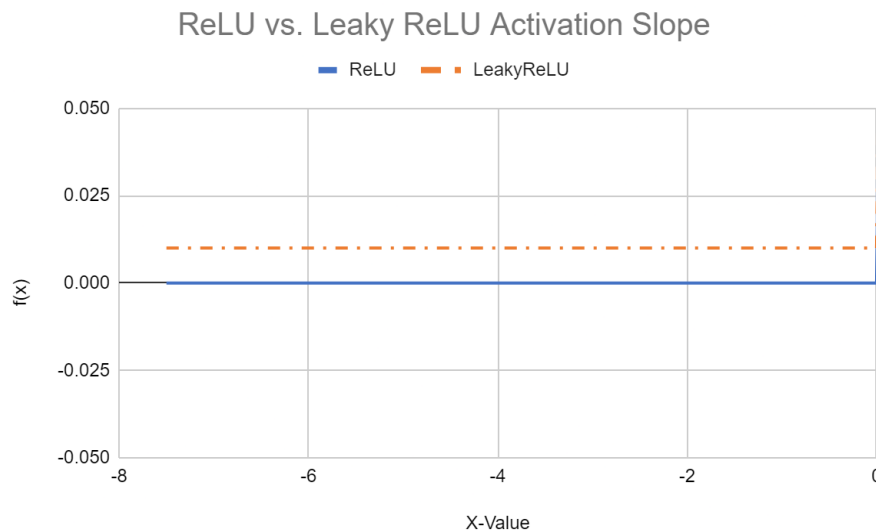
Figure 3.5.2: ReLU vs Leaky ReLU functions point derivations graphed

With ReLU having a slope of zero, we can see the non-linearity and where the dead neurons originate from. In the case of Leaky ReLU, there is still hope for neurons, but the function becomes a linear one, as negative values become a multiple of itself, rather than being cut off.

3.5.3 Sigmoid

The Sigmoid function is used to crush large inputs in the range of [0, 1]. This simplifies the output of the neuron to in an "on" or "off" state. The equation for Sigmoid is provided below.

$$Output(x) \; = \; \frac{1}{1+e^{-x}}, \; where \; x \; = \; input \; value \qquad (3.5.3)$$

The problem with this activation function is that through many layers, the gradient will vanish. This is because of the fact that even with large changes in input, the resulting output will always be between 0 and 1. For this reason, Sigmoid is not a preferred option in today's world. We will most definitely not be making use of it.

3.5.4 Tanh

Tanh is very similar to the Sigmoid function, but the range is expanded to [-1, 1] and is now centered at the origin. The equation for Tanh is provided below.

$$Output(x) \; = \; tanh(x) \qquad (3.5.4)$$

The reason tanh is an improvement over sigmoid is because the output ends up being normalized, averaging at 0. This still doesn't solve the problem of "vanishing" gradient though. Tanh is not much of a choice nowadays, and we will not be utilizing it either. Both Sigmoid and Tanh are graphed in Fig 3.5.5, where we can better see the similarities illustrated.
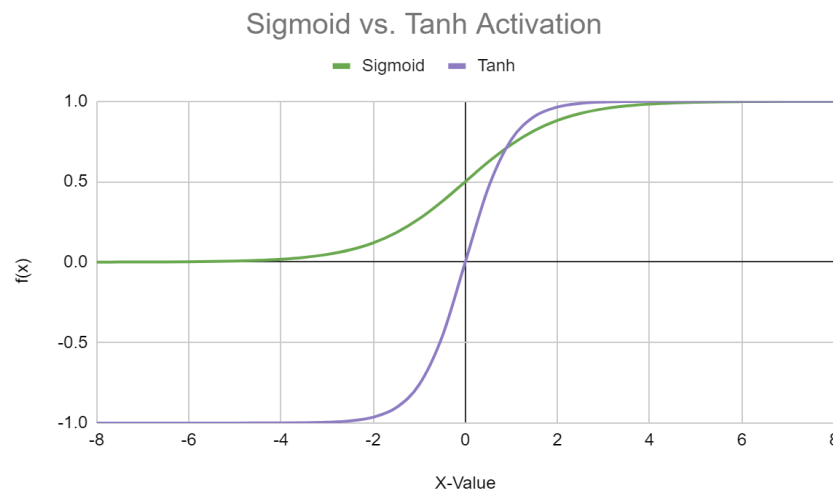

Figure 3.5.5: Sigmoid vs Tanh functions graphed

3.5.5 Swish

Swish was introduced by Google and acts as an alternative to ReLU. The swish function makes use of Sigmoid within its equation

$$Output(x) \; = \; x \, sigmoid(x), \, where \, x \; = \; input \, value$$

The function removes the flattening effect the original sigmoid function can have by multiplying the result by its original value. References showed that this function worked better on deeper models with higher complexity datasets. This activation function is utilized in the YOLOv5 model, which is one of the most popular real-time object detection models out right now.

ReLU vs. Swish Activation
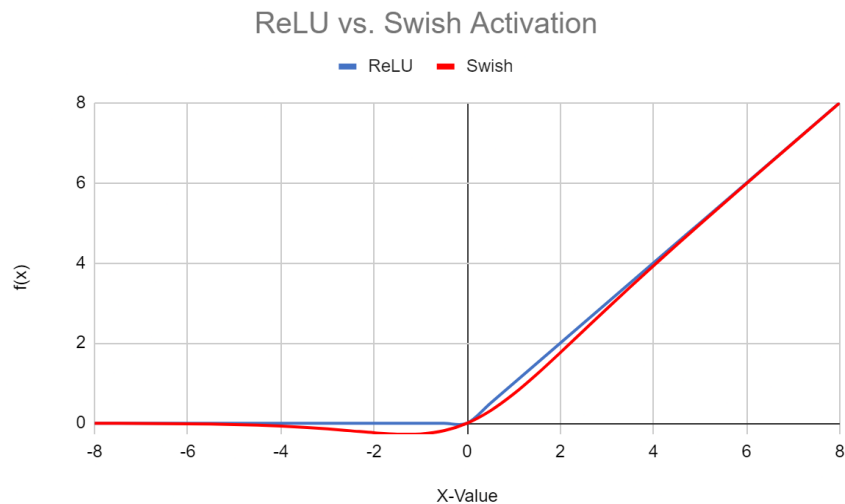
Figure 3.5.6: ReLU vs Swish functions graphed

3.6 Loss Function

Loss functions are the functions used to measure how well a model is being trained to work the input dataset. There is a desire to keep loss low, which in turn makes our predictions more accurate. When a neuron is met with an input that when fed into a loss function, results in a high loss, it should then adjust its weights to reduce it. There are a few functions to choose from, examples being: Cross Entropy, Mean Absolute Error, Mean Squared Error, Root Mean Squared Error, and Hinge. These functions are also ready for use in Pytorch by importing the torch.nn module

3.6.1 Cross Entropy

Starting with the cross entropy function, we have identified its functionality and its typical application. There are actually two versions for Cross Entropy, defined for when the number of classes is equal to two, and when there are more than two classes. When there are only two classes, the Binary Cross Entropy function is used. When there are more than two, the Multi-Class Cross Entropy function is used. These functions measure loss between pairs of probabilities for

a randomly chosen variable. The equations for both versions of this loss function are provided below.

$$B\ Loss\ =-\ (ylog(p)\ +\ (1\ -\ y)log(1\ -\ p)$$

$$MC\ Loss\ =-\ \sum_{j=1}^{n} y_{o,j} log(p_{o,j})$$

Where 'y' is the binary value, 'p' is the predicted probability, and 'n' is the number of classes

These functions are the most commonly used loss functions for classification. The random selection of comparisons that are made in addition to the logarithmic loss formulation combine to create an effective way to train a model. The logarithmic loss accentuates large errors, making them more likely to get snuffed out and improve the model's predictions. Random selection removes some of the biases that would have been found in normal means. From our observations, many CNN packages utilize this loss function, so we will as well.

We graphed the Binary Cross Entropy Loss for both binary inputs. In this graph, it is easy to see the logarithmic trend for the function. When the binary indicator is true, and prediction probability is high, then loss is low. The opposite is true for when the binary indicator is false.
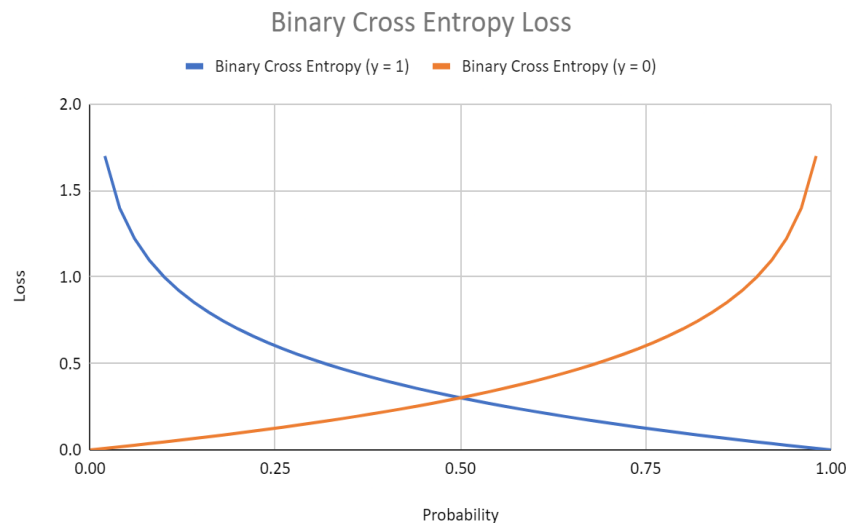
Figure 3.6.1: Binary Cross Entropy Loss

3.6.2 MAE

Mean Absolute Error, also known as L1, is one of the most commonly used loss functions for linear regression. MAE takes the sum of the absolute

value of differences between the actual value and the measured/predicted value. The equation for this loss function is provided below.

$$Loss = \frac{1}{n} \sum_{j=1}^{n} |y_j - \hat{y}_j|$$

Where 'y_j' is the target value and '$\hat{y}_j$' is the measured value.

In many cases, MAE is seen as more resistant to outliers because of it being the average error. A single outlier may only move the average slightly if the number of errors being summed is large. The function is graphed below in Fig 15.8. Predictions that result in output values closer to zero from either side will end up with lower loss values. The opposite is true in cases where there is an output that scales farther and farther from zero, and it will end up with an equally great loss value.
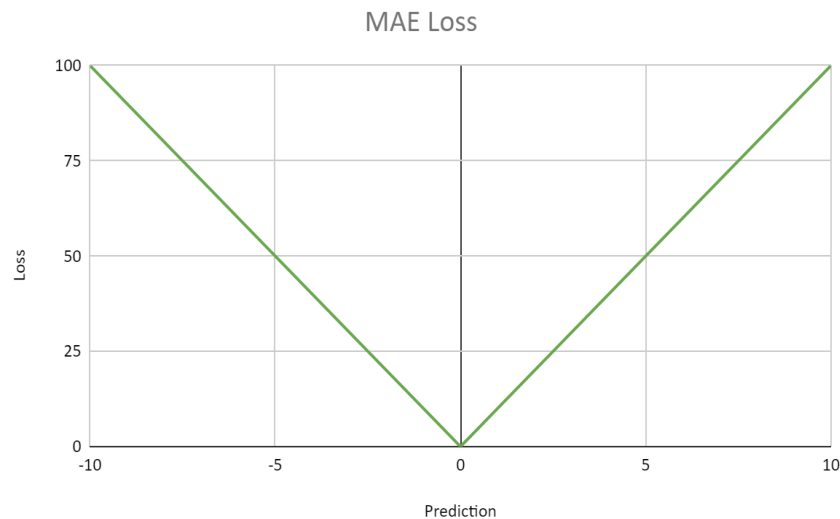
Figure 3.6.2: MAE Loss Graphed

3.6.3 MSE

Mean Squared Error, also known as L2, is the most popular function for linear regression. MSE builds off MAE and squares the difference between the desired and measured value, then sums each iteration together. The equation for this loss function is provided below.

$$Loss = \frac{1}{n} \sum_{j=1}^{n} (y_j - \hat{y}_j)^2$$

When the difference is greater than one, the loss becomes much greater due to the squaring. This means detection for errors is harsher and will train the

model to become more accurate. This also means that outliers have a greater effect on the overall training. The network will end up overcompensating for even a single outlier as it may alter the overall loss that greatly. The MSE function is graphed below in Fig. 3.6.3. We can see the similarity to the MAE function, but loss scales exponentially now instead. This illustrates the heavier bias against an error value that the function has over MAE. Small error values that may be acceptable when fed through an MAE function, would not be under MSE.
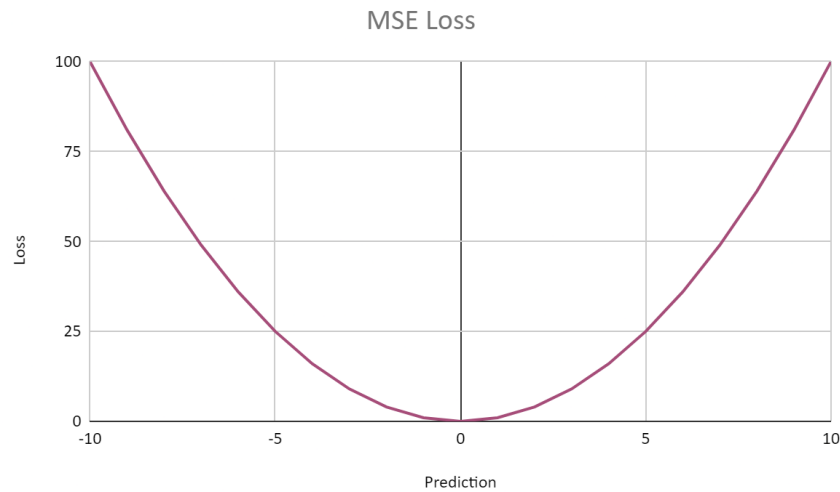
Figure 3.6.3: MSE Loss Graphed

3.6.4 Hinge

Hinge is another classification loss function. Hinge's primary use is geared towards support vector machines though, which is another method of classifying data over a neural network. The equation that describes the loss function is provided below.

$$Loss = max(0, 1 - \widehat{y}_j * y_j),$$

Where 'y$_j$' is the target value and '$\widehat{y}_j$' is the measured value

The Hinge loss function sets a boundary for values to be measured against. There are also two outer bounds, ± a set distance from the main boundary. Values that plot outside the outer bounds are given a loss of zero or one, depending on the bound they lay outside of. Within the outer bounds but not touching the main boundary are given a loss between zero and one. And those values that touch the main boundary are automatically given a loss of 0.5. This organizes the inputs into three categories that make adjustments clearer to compute.

In Fig. 3.6.4, we have simulated the separation process for the Hinge Loss function. Analyzing the points on the graph, we would be left with several conclusions.If desired, the predictions outside the lower bound can be given a loss of one, labeling them as highly undesired, or as one and accept them. The three points within the outer bounds are given an arbitrary loss value. Lastly, the prediction lying on the main bound is given a loss of 0.5. The predictions are now organized and the CNN has the ability to weed out undesirable predictions depending on which outer bound we want to label as a high loss identifier.
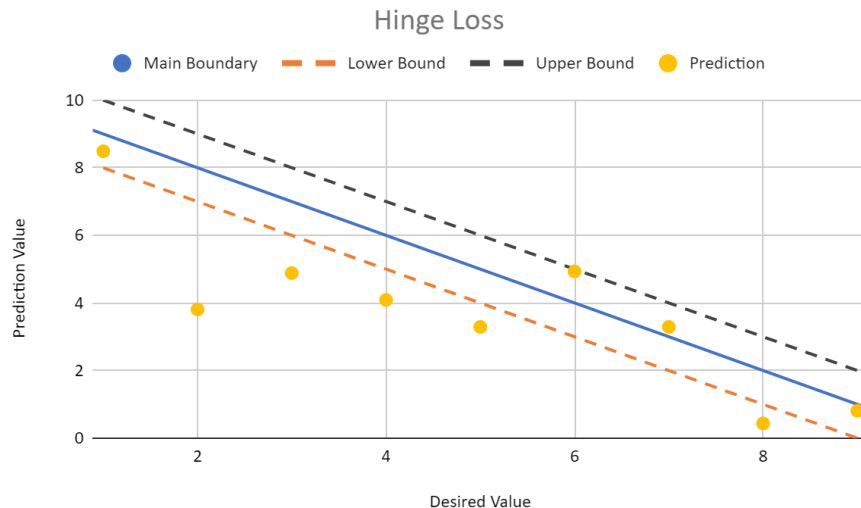
Figure 3.6.4: Hinge Loss Graphed

3.7 Programming Language: Python

Python is one of the most popular programming languages, and reasonably so. Python values accessibility above all else with its simplified syntax and many abundance of useful packages and libraries. Python isn't limited to a single OS, as it can be run on Linux, macOS, Windows, Solaris, etc.. Pair this with the fact that Python scripts can be made to run at a much faster rate with a GPU, this makes it a great money-saver for companies that don't want to invest in cloud services from companies like Microsoft (Azure), Amazon (AWS), Google (Google Cloud). GPUs can cover some of the ground lost by not investing in these more expensive "Machine Learning as a Service" options without losing out on too much productivity. The possibility of GPU utilization is a big contributor to a project as well. We are using a system with a NVIDIA RTX 3060ti, which has 4864 CUDA Cores, and 152 Tensor Cores. This should prove to be valuable in reducing our training time.

3.7.1 Distribution

PyTorch and TensorFlow are the two machine learning frameworks that are popular in today's world. Those were also the two options given to use by the original sponsor as well. Our sponsor highly advised us to utilize PyTorch though.

This was because the resources they had already dedicated to their own project were done using PyTorch. That meant that we would have references for moving forward with our project. Doing our own research though, we found that there were more reasons to select PyTorch over TensorFlow. One was because of the ease of implementation. PyTorch is described as being more in-line with the original Python libraries. We believe this makes Python 'easier' to learn when you already have knowledge of Python. Python is also known as an 'easy' language to learn compared to others. The community backing PyTorch is also massive and out-matches TensorFlow. For our own research, this would be much more advantageous as this is our first time utilizing a framework at all. We can really make use of the open-source packages that are out there to improve our convolutional neural network. A downside for PyTorch is that we have to source software to deploy our model onto the drone, but there is plenty of material to cover that just from what we have found .

We are using Anaconda to manage our packages and run Pytorch on our simulation system. This was recommended mostly for its access to a large number of data science packages, which should aid us in building our CNN. Pip is necessary for PyTorch installation onto the Raspberry Pi so its only utility is with pre-trained models. Using Pip within the Anaconda environment will just install the packages as normal with no complications tied to it. Anaconda also grants the ability to update all packages with a single command. The package management of Anaconda will make handling the CNN safer and easier

3.7.2 Libraries & Packages

The most important python library that we are using is obviously PyTorch. PyTorch provides a deep box of machine learning tools along with an API with extensive documentation to back it.

3.7.3 Machine Learning Language

Python is often the first choice when pursuing the machine learning route. This is due to its amazing collection of packages and libraries, as well as a large backing from the community that uses it. Because of its ease of accessibility making it easier to learn and digest, open source code is compatible with many other projects. The readability is just as beneficial to beginners as it is to seasoned veterans of the language. Machine Learning can be very intimidating with the implementation of complicated algorithms, but the overall simplicity of Python itself relieves a lot of that tension. We are creating our convolutional neural network in Python so we hope to take advantage of the benefits that it provides.

3.8 CUDA Cores

CUDA stands for Compute Unified Device Architecture and CUDA cores are components of Nvidia Graphical Process Units (GPU)[13]. They handle similarly to a CPU core, but deal with a lower-level instruction set and are less

complicated. They are especially good at parallel computing because of how many Nvidia employed within a GPU. This works well in our case because less complicated but more abundant sets of instructions are exactly what make up our machine learning task. The fact that we can make use of these CUDA cores will vastly improve our training speeds in every way.

3.9 Training Model Packages

In researching image classification and neural networks, we had made the decision to use a convolutional neural network. We learned quite a lot about the functionality of a CNN and the ways in which it can be configured. We also made the decision to alter an open-source CNN to fit the goals and objectives of our project instead of creating one from scratch. This was because we now know the level of coding needed to create one, and it may take far too long and may be beyond what is realistically achievable given what the rest of the project demands. In this section, we will explore the options that we observed, and also decide on which we are choosing, with a detailed reasoning to back our choice.

3.9.1 Microsoft - CNN

In searching for image classification models to choose from, big tech companies were bound to be a possible selection. Microsoft has invested extremely large sums of money into research based around artificial intelligence, expanding yearly. This includes a hefty amount of resources dedicated to growing the Machine Learning field. Microsoft is already known to provide cloud-based applications through their Azure platform, and this includes machine learning ones too. Azure Machine Learning is Microsoft's method of distributing machine learning models. The amount of resources they have dedicated to improving not only their ability to provide a machine learning product, but to the growth of machine learning in general is impressive.

This gives them quite a bit of credibility and piqued our interest when we found a sample CNN with instructions on how to deploy it for image classification. Through further research, we found that their set of instructions utilized four software design choices we were already fixated on. The first was obviously that it was a CNN. This is important as it is optimal for image classification purposes, which they note in the article.The second being that it was done on Pytorch. As we have already decided on using Pytorch, this was another great thing to see. The third and fourth were that they implemented the ReLU and Cross-Entropy loss functions. These two configuration options were two that we had already been looking out for, so it confirmed some of our hypotheticals.

These combined components meant that we would seemingly make little to no alterations to the network in order to fit our project. Pair this along with the fact that every bit of code is paired with instruction and explanation behind it, using Microsoft's model is a seemingly great route. In the complete set of instructions, they deploy the model onto a Windows machine. We will most likely

be deploying on a linux-based system for the simulation portion of the project, and then onto a Raspberry Pi for real testing. We can just cut out that part of the instructions though, as the Pytorch model is what is most important.

3.9.2 Google - GoogLeNet

Google is another big tech company that invests an extremely large amount of resources into artificial intelligence, which bleeds into machine learning research. Google took a stab at it after being influenced by the winner of the 2012 ImageNet challenge, LeNet. They produced their own open-source architecture GoogLeNet in 2014. The neural network is one derived from CNNs but is known as an Inception Network due to how the convolutional layers are compartmentalized into what is called an "Inception Module". Each inception module is made up of convolutional layers with differing sizes and then connected to sequential inception modules.

The code for this architecture is readily available on the Pytorch site with a github repository and is a pre-trained model. There isn't instruction paired with this implementation but there is plenty of supplemental content to choose from that are IEEE published. We see this as the least welcoming of the options, as it has the least direction tied to it.

3.9.3 YOLO Detection - Ultralytics

After doing thorough research of image recognition models, we found the open-source object detection system YOLO. YOLO stands for "You Only Look Once." This algorithm was originally introduced to the public in 2016 and has gone through many iterations since then. Originally, YOLO was written in the framework Darknet. The latest versions however have implementations using PyTorch, so this piqued our interest. We found more information regarding this algorithm on the website Roboflow.

Roboflow is a computer vision platform that specializes in classification services. On this site, they provide custom dataset creation, which includes object detection and image classification annotation that can be edited, saved and downloaded at any time. Roboflow also provides open-source notebooks that allow us to train an object detection model how we see fit. This would save us a ton of time building our own model, as well as remove reliance on a high performance local system, as training would be done online. The notebook allows CUDA usage as well, essentially performing better than local training would. The platform goes a step further and has their own python package that can be downloaded and used to locally download a dataset, a model, and run said model on any stream of data. They also provide a plethora of articles covering steps on completing this process, not only on a Windows/Linux/Mac machine, but also deploying onto a Raspberry Pi. This is extremely valuable in our case as our drone is to be run on a Raspberry Pi. This is most definitely the leading choice for obvious reasons.

3.9.4 LeNet - *"Ben Trevett's Pytorch Image Classification"*

The next architecture in the list is the "LeNet" architecture. This architecture is made up of two pairs of convolutional layers that are subsampled in between, and then are followed by three "fully-connected layers". The author explains that the architecture is modified and optimized for "classifying handwritten characters", so it may not be applicable in our project but we may have to test on a larger scale. The architecture has its roots set all the way back in the year 1998, so it has been through decades of testing, resulting in decades of optimizations. Again the testing is done to identify a written character, using the same dataset as before. The CNN allows for a lot more precision in distinguishing the pixels in the images apart so we expected far better results. Trevett also makes use of the ReLU function as the CNN's activation function. This is convincing us to use the ReLU function over the Leaky ReLU function in our own project. He also goes on to utilize the Cross-Entropy loss function, which influences our loss function choice as well. The architecture yields varying results, with some images predicted at a rate of 99%, and some as low as 85%. However, we did observe that the images with lower accuracy were difficult to identify from our own eyes. This may be a suitable architecture, but more testing would be required.

3.9.5 AlexNet - *"Ben Trevett's Pytorch Image Classification"*

The third architecture listed is "AlexNet". AlexNet is another model that uses a modified CNN. Originally, this architecture was meant to utilize parallel processing between two GPUs at the same time. The reason for this is that there are two "paths" of training going on in parallel with each other that then converge to make a prediction. Each path consists of alternating convolutional and max-pooling layers. This was the winning architecture for the ImageNet challenge in 2012. This led it to become a very popular image classification model. The author presents an altered version of AlexNet that allows for use on only one GPU. There does not seem to be an explanation for this within the notebook, but from what we found, it is possible through copying layer data to the CPU for temporary storage. There are also memory allocation methods from Nvidia that may make this possible as well.

Moving onto the experimental results displayed in the notebook; the dataset used is much more complex than previously used. This set includes many small full color images and it is called the CIFAR10 dataset. The fact that full color images are being used instead of gray scaled images made this a much more promising architecture. If it can classify complex images that are found in this very popular dataset, we are confident that we can alter it to fit our needs. ReLU was again used as the activation function. The loss function was also Cross-Entropy again, so there appears to be popularity with this activation-loss function pair. The results were much less accurate than the previous two models, but this is to be expected. After only 25 epochs at ~14 seconds per epoch, the training accuracy was up to 78.1% and the validation accuracy was 75.9%. We came to the conclusion that with longer epoch times or more epochs, the

accuracy could be a bit higher. The author also found an error within the dataset where the model predicted correctly, but the dataset labeled the image incorrectly. Overall, this seems like a possible implementation for our project, but the YOLO algorithm still reigns over it.

3.9.6 VGG - "Ben Trevett's Pytorch Image Classification"

The next architecture is another commonly implemented one, and it is named "VGG". Previously discussed, AlexNet was the winner of the 2012 ImageNet challenge. VGG was the winner of the 2014 ImageNet challenge, which lends an explanation as to why it is so popular. This set of models is another one that utilizes a CNN style. VGG is commonly notated with a number to signify how many convolutional layers, for example: VGG-16 has sixteen convolutional layers. Trevett explores configurations with 11, 13, 16, and 19 convolutional layers.

Within this experimentation, Trevett also makes use of a pre-trained model. The pre-trained model can "transfer" its learnings and apply them to another dataset. This saves time, but brings the model to a point where it learns much slower. This is seen within the results with each epoch taking around 8 minutes. The prediction accuracy ended up being much higher than that of AlexNet though. After 5 epochs, the validation accuracy was up to 93.6%. We believe this to be a reasonable choice for our project. The dataset used was much more complex than in the MLP and LeNet, so it can match up to the input we have for it. The training time will not be a problem due to our ability to utilize CUDA cores on a strong Nvidia GPU. This is another suitable configuration for our image classification.

3.9.7 ResNet - "Ben Trevett's Pytorch Image Classification"

The last architecture is labeled "ResNet". This is another winner of the ImageNet challenge, winning in 2015. This one takes after VGG in that it has configurations that are noted to signify the number of layers and their sizes. This architecture is seemingly much larger than the previously described ones. The highest configuration within the notebook is ResNet152, which means it has 152 convolutional layers. The actual build of ResNet is very similar to that of VGG, a chain of convolutional layers that are increasing in size, but many more layers in comparison. The key distinguisher is the fact that the input of one layer and output of the next layer are connected via residual connectors. This allows for data to "skip" to other layers, which can prevent activation functions from diminishing them. There is definitely more to research for this configuration, as the residual connectors and sheer abundance of layers makes this intimidating. The training done by Trevett is done with the CUB200 dataset, which is another popular dataset used in classification training.

The method of calculating accuracy differed from the other architects though. Instead of using the prediction accuracies as is, they are softened due to the complexity of the images within the dataset. Travett describes the images

being much harder to classify to even humans, so it would be more realistic to curve the prediction results. He does this by comparing five predictions instead of just one to the actual label. This obviously raises the accuracy quite a bit, from 79.5% to 95.5%. We don't believe we would be employing this method of curving the prediction results. If we were to deploy this method then we believe there would be conflict with how the drone interacts with its environment. We may not have to inflate the results anyways though, as our obstacle course will not consist of objects with low color variety. Our objects will highly contrast with the environment, made with solid primary colors. We would have to test it ourselves because this implementation does not give a direct comparison to AlexNet nor VGG due to the dataset and curved results.

3.8 Gazebo

Gazebo is the 3D open-source simulator previously described in the relevant tech that was referenced. That team made use of it to simulate drone swarms and experimented with machine learning collision-avoidance. Since our project is not too far off from theirs, we believe it is advantageous to utilize it as well. Gazebo will allow us to create our obstacle course in the simulated environment and control nearly every variable possible. Testing data is much more readily-accessible from a simulated environment than real sensor data. The ability to repeat trials at any pace and without risk of causing harm to the drone makes an invaluable testing component.

3.8.1 PX4 Support

Gazebo's support for the PX4 control system simplifies the build and testing process by orders of magnitudes[14]. There is no complicated process that we have to produce to get a one-to-one representation of our drone in the simulated environment. We just have to run it on a Linux system, utilize the necessary plug-ins, and connect the two via MAVLink, which is a messaging protocol.

3.9 PX4 Autopilot

PX4 Autopilot is the open source flight control system that can be applied to the drone kit we are purchasing. We were fortunate to come across the research study that utilized PX4 to control their automated drones. The system also comes with a built-in simulator, but does not meet the creative needs that Gazebo can provide us. The simulator, jMAVSim may be used for initial movement testing but nothing past that stage.

3.10 Mission Planner and Calibrations

Mission Planner is the software that is used for drone calibration and flight planning and is crucial for successful flight. It contains multiple testing

applications in which the user can fine-tune all flight components of the drone, including the compass, electronic speed controllers, the motors, and the GPS module.

Figure 3.10.1: Mission Planner Data Interface

This software implements pre-arm checks as well which allows the drone to be armed for flight given that all prerequisites are met. Here are a few examples of messages that are displayed if the drone cannot be armed due to not clearing pre-arm checks:

- No GPS Fix: The drone needs a valid satellite signal in order to be armed
- Throttle Failsafe: The RC controller is set to full throttle
- RC Failsafe: There is no transmitter detected

The calibrations are also required to arm the drone and include multiple different fields. The compass is the most important as it gives the drone full awareness of direction which is key to stable flight. The radio calibration, as shown below, allows the user to fine tune minimum and maximum outputs for throttle, pitch, yaw, and roll for more defined flight. The ESC calibration allows the PixHawk module to determine how much power to provide to the motors and also is the main source that allows the motors to spin.

3.11 Robot Operating System (ROS)

ROS is an open source API for automating vehicle control within the Gazebo simulation environment. If we were to utilize it then there may be complications with PX4 but PX4 provides a workaround within their User Guide. ROS would assist in managing the drone's data sources, which may in turn

streamline the input for our CNN. The framework also runs in real-time so it might save us trouble when analyzing our data.

3.12 Technology Comparison

The commercial drone technology industry continues to push its limits and there are large companies doing extensive research including Dji, Autel Robotics, Parrot, Walkera, Blade, Helimax, and many others. Focusing on the two largest drone companies in the United States market, DJI and Parrot, which are both renowned companys and the two first billion dollar drone companies. The table below shows a comparison between the two drones models.

Table 1. Product Comparison

	DJI Mavic 2 Pro	Parrot Anafi
Photo Product		
Flight time	31 Minutes	25 minutes
Flight Speed	20 m/s	15.3 m/s
Flight Distance	18 km	4 km
Weight	907 g	320 g
Volume	6518.57 cm^3	2730 cm^3
Dimension	241 x 84 x 322 mm	175 x 65 x 240 mm
Price	$ 1.729 (DJI Store)	$ 700 (Parrot store)
Video Recording	2160 x 30 fps	2160 x 30 fps
Main Camera	20 MP	21 MP
Field of View	77°	84°
Battery Power	3850 mAh	2700 mAh
External Memory	128 GB	16 GB
Product excellent	Has a serial shot mode Can create panoramas in-camera Obstacle detection	Smaller product dimensions Product price is lower

Source: versus.com

Figure 3.11.1: Comparison of two drones

As one can see in Figure 3.11.1, the DJI drone is the leading competitor in drone technology with the highest flight time, flight speed and flight distance. Also with higher battery power and a larger external memory availability. Although video recording holds the same frame per second rate, the DJI model has the ability to shoot photography in serial shot mode, create panoramic views and can detect obstacles.

Obstacle detection is where the neural network software uses the camera on the drone to identify objects in its view. This is commonly seen to the general public by green box detection in videos. The machine learning software in DJI technology research are doing current studies on developing a drone that is controlled by human hand gestures. With this being said, the DJI model drone is the strongest competitor in the commercial drone market and with that comes

high end prices, where one will pay more than double for a DJI drone versus a Parrot drone. The project is focused on Visual Navigation and reviewing the most popular Vision Positioning System (VPS) drone on the commercial market. This drives one back into DJI technology and introduces the DJI Phantom 4.

Looking back at the cost and performance, we want to relate to a product tested that is low cost with generally good performance. In 2010, the Parrot company started a project called the AR.Drone, which had from 5 to 12 engineering from Parrot with support from SYSNAV and academia MINES ParisTech. SYSNAV provided deep technical support and MINES ParisTech provided research and testing on navigation and control design. The AR.Drone aims to produce a micro Unmanned Aerial Vehicle at low cost for indoor navigation for the mass market of video gaming and home entertainment.

Figure 3.11.2: Augmented reality vision based drone control

The augmented reality drone design is primarily controlled by a remote that still detects obstacles in its view for augmented gaming and cinema experiences. This device has an integrated WIFI chip and bluetooth for the purposes of a user-friendly graphical interface on Apple products such as the iPhone, iPad and iTouch. This drone is currently available online and in retail stores in selected countries at a price below $330 USD. The device is currently available for sale on Amazon but with unavailability to be purchased in the United States. This drone was developed and distributed to sell by French engineers in France. After more research, it is seen in many online photos that although the drone was intended to be used in-home and confined spaces only, the general public still took the device outside for gameplay and this caused some disturbances in governmental regulations. Hence, why it is also being discontinued by the Parrot drone company for general public usage.
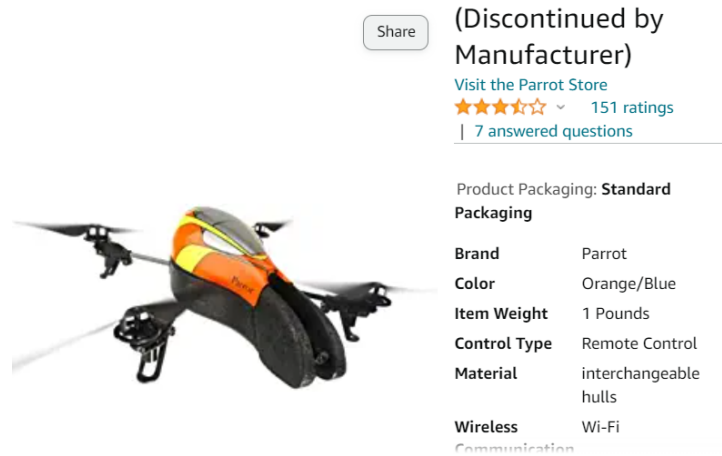
Figure 3.11.3: Parrot drone on Amazon for sale in outside countries but discontinued by manufacturer

3.13 Part Selection

The following section primarily serves as a way to showcase all of the key components that the drone system will use in its operation. It shows the models of the parts that the group decided on

Figure 3.13.1: F450 450 4 axis quadcopter frame kit for Pixhawk Flight Controller), Propellers eight needed - four for usage and four for backup and propeller guards

Figure 3.13.2: Electric speed controller for brushless motors and the power distribution board

Figure 3.13.3: Autopilot Pihawk PIX4 ARM Flight Controller, Raspberry Pi4 & High Quality Camera(2), 3.7V 6000mAh Universal Lipo Battery, 4K 64G SD card Brushless Motors(6), Vibration damper, GPS Module, fire proof case for to store material for safety, FPV Video transmitter, pin space wires and gps mount

3.14 Part Comparison

3.14.1 Sensor Comparison

	LiDAR	Stereo Vision
Area Coverage	100 m to 10 km^2 250 m to 50 km linear	0.20 m to 20 m approx.
Power Consumption	10 W to 50 W	11 W to 25 W
Operating Environment	No reflection in open space	good lighting condition with texture rich environment
Processor Platform	ARM v9	ARM FPGA
Cost	upto 25,000 USD	500 to 800 USD

Figure 3.14.1: LiDAR and Stereo Vision Comparison

Accuracy of GPS coordinates depends on the sensors. The assessment of the GPS location of moving objects is a challenging task. The below graphic displays the core differences between the LiDAR sensor system as well as basic stereo vision.

For this project, even with a smaller floor area of approximately 50 square feet to test the functions of the drone, area coverage / field of view for the sensor is very important. The table shows that the LiDAR system offers a substantially greater amount of coverage compared to stereo vision which covers less than approximately twenty meters.

In terms of power consumption, both sensors produce similar voltage output. As long as the drone's breadboard battery has enough voltage to power the sensor without any shorting or other issues, then either sensor will suffice in this case.

For the operating environment, the drone shall only fly within the mesh cage that is built in a well-lit room on campus. Therefore, the LiDAR would be more sufficient in this case as there won't be any reflection from direct sunlight since the testing environment is solely indoors.

Finally, the cost and processor platform are not necessarily significant for this project since the drone is small in size and doesn't require any highly advanced technology in terms of processing, thus lessening cost overall.

Table 3.13.2: Camera Module Comparison

Spec	Arduino Uno	Raspberry Pi 3 B
CPU Type	8-bit Microcontroller	64-bit Microprocessor
Operating System	None	Some flavor of Linux
Storage	32 kB flash	Depends on size of SD card
Memory	2 kB	1 GB RAM
Speed	16 MHz	1.2 GHz
GPU	None	Built in
Networking	None	Ethernet, Wi-Fi, Bluetooth
Price	$20-$22	$35
USB ports	1	4
Power consumption	Can be < 0.25 W	Several watts

The camera is also a key component for the drone to clearly detect objects, which is the main function of this project. The group selected the Raspberry Pi module due to its compatibility with other parts, although similar to the Arduino Uno module which is compared in the table above. The 64-bit feature

allows for faster processing speeds which in turn may result in better quality when viewing the drones perspective mid-flight.

The biggest takeaway between the two cameras is the clock speed. The Arduino Uno only has a maximum clock speed of 16MHz, while the Raspberry Pi has capabilities of up to 1.2 GHz, or 1200MHz, an increase of nearly 75x. Although this comes with higher power consumption, this is not a very significant issue for this project.

Finally, the cost between these two modules is rather similar, but with the faster speeds and higher RAM, the Raspberry Pi is clearly the better option for the drone.

3.15 Software Design Comparison

There were essentially two options given to us when we decided to take on this project. The machine learning framework is an important part in not only how the training is computed, but also brings in different coding libraries that train differently. PyTorch and TensorFlow are the two machine learning frameworks that we were given to choose from. The table below outlines some of the characteristics of them.

Table 3.14.1: Comparison of Machine Learning Frameworks

Framework	PyTorch	TensorFlow
Founder	Facebook	Google
Method of Computation	Dynamic - computational graphs are created when necessary	Static - computational graphs are statically defined when compiled
Distributed Training	Built-in	Manual SetUp
Pro	Stronger Community Presence and is more Python-like, making it easier to learn	Better Visualization of Training and Larger Libraries
Con	Little to No Visualization of Data Training, can be more difficult to identify issues	Does NOT FEEL like Python, so it takes more time to learn

The table below is a comparison of potential activation functions that we can implement for our convolutional neural network. Activation functions play an integral part in determining a neural node's output, which can heavily affect the quality of predictions that are produced. Our choice is made through thorough simulation testing and comparing real results. The final decision will most likely be ReLU or Leaky ReLU based on their attributes in comparison to other functions.

Additionally, the loss functions play a very important role in producing much more accurate predictions within the convolutional neural network. Again, our choice will stem from rigorous testing and then comparing the results of the functions shown in the table. Cross-entropy will most likely be our choice because of its pros shown below in Table 3.14.3. The few cons that this does have can be easily worked around and overcome with some additional effort from the team.

The table below is the comparison of open-source training model packages that we researched. These are options that we were able to find and they are the determining factor for how well our object detection model performs. We can make alterations but the preset is important for us to get a foundation to build upon.

Table 3.14.2: Comparison of Activation Functions for CNN Configuration

Activation Functions	ReLU	Leaky ReLU	Sigmoid	Tanh
Equation	g(z) = max(0, z)	g(z) = max(0.01z, z)	g(z) = 1/(1+e^(-z))	g(z) = tanh(z)
Description	Lower threshold implemented at 0	Similar to ReLU but lower threshold is product of 0.01 and z, rather than cut off at 0	Crushes input to coincide within the range [0, 1]	Similar to sigmoid but centered at zero for both the x and y-axis, so range is [-1, 1]
Pytorch Implementation	torch-NeuralNetwork library	torch-NeuralNetwork library	torch-NeuralNetwork library	torch-NeuralNetwork library
Pros	Gradient does not vanish as in Sigmoid or Tanh Simpler to implement than most other activation functions	Attempts to compensate where ReLU may fail, a neuron will never remain inactive permanently because weights will adjust to small output, rather than have no slope at all	Flattens gradients to have a neuron activate, "1", or remain off, "0"	Average output is closer to 0 than Sigmoid Better normalization
Cons	Neuron can remain inactive depending on how the weights are changed in response to an input	May still have original ReLU problem of neurons permanently going inactive	Gradient can quickly vanish	Gradient can quickly vanish

Table 3.14.3: Comparison of Loss Functions for CNN Configuration

Loss Functions	Cross Entropy	MAE	MSE	Hinge
Description	- Loss decreases as probability of prediction increases at a log rate - Low measured value equates to a high loss value	-Loss calculated from the sum of absolute differences between desired value and measured value	- if loss becomes much greater when the difference between desired and measured value is greater than 1 - Makes loss more volatile, model will reduce to get below loss of 1	- Separates measured values based on distance from a set boundary - Not on boundary equates to a loss within the range of (0, 1) - On boundary, values are set with a loss of 0.5
Pytorch Implementation	torch-NeuralNetwork library	torch-NeuralNetwork library	torch-NeuralNetwork library	torch-Neural Network library
Pros	- Log rate causes higher loss to be exemplified and stand out more to the network, calling for better optimization	- Resistant to outliers	- Adjustments made post-loss function result in smaller changes compared to MAE	- Separated loss values are easier to distinguish and allow the network to make more defined predictions
Cons	- Optimized for classification models	- Dependent on optimized incrementation of adjustments, if not then the difference per iteration may be too large, resulting in skipping over a viable parameter configuration	- Outliers can cause problems due to the loss being squared per iteration, this will cause the network to "overreact" to the single iteration and fail to make an accurate prediction	- Optimized for Support Vector Machines

Table 3.14.4: Comparison of Training Model Packages

Training Models	Microsoft Sample	GoogLeNet	YOLOv5 (real-time)	YOLOv8 (real-time)	Pre-trained VGG	ResNet
Type of Network	CNN	CNN	CNN	CNN	CNN	CNN
Activation Function most commonly used	ReLU	ReLU	Swish	Leaky ReLU	ReLU	ReLU
Loss Function most commonly used	Cross Entropy	Cross Entropy	Cross Entropy	Cross Entropy	Cross Entropy	Cross Entropy
DataSet Trained (source-results)	CIFAR10, 50,000 training images and 10,000 testing images	ImageNet, 150,000+ images from 1000 categories	COCO	COCO	Image Net & CIFAR 10	CUB 200-2011, very complex set of birds
Accuracy (source-results)	70% (Top-1, short training period)	89% (Top-1) 93% (Top-5)	50.7% mAP	53.9% mAP	94% (Top-1)	80% (Top-1) 95% (Top-5)
Implementation Difficulty	Low - tutorial	Medium - only code	Low - tutorial, little to augment	Low - tutorial, little to augment	Medium - tutorial	Medium - tutorial

4 Design Constraints and Standards

The creation of the drone has multiple constraints that the group must work around in order for the project to be successful. This section contains lists of the different types of challenges that a completed drone will face in order to have a proper flight. These include the size of the system, the weight, and safety concerns among others. Safety is a key factor when creating a drone, and particularly so with an autonomous one. The drone should avoid making any collisions with humans or other private property. This is one of the functions of the object detection system.

Often, flying a drone in public areas may require several government permits. The group must take these into consideration, and they are the primary administrative restrictions that they have. The group is already aware that permission is required to fly any drones on UCF's main campus, which is where the drone testing is likely to take place. Additionally, to prevent any issues with the school, the group is planning to create an indoor testing environment inside of a private laboratory on campus. They will acquire the permissions and access codes and keys necessary to access the lab, and

Most of the other restrictions are due to the physical and technological limitations of the system. These factors include things like the required size, weight and thrust of the system. With any flying object, and especially a hovering object, all of these must be balanced in perfect harmony. Otherwise, the entire system risks catastrophic failure. An imbalance can easily lead to too much or too little lift, resulting in the whole system crashing. Any accidents at significant speeds and heights may completely destroy the flight systems as well as any number of other component systems. It is for these reasons that these constraints are considered by the group to be some of the most important in the entire project. Above all else, these constraints will make all of the difference between the success or the total failure during the final project and presentations.

There are also certain environmental factors that can alter the effectiveness of the system. It should be noted that the group will only require the image recognition and flight systems to operate at their fullest capacity in a controlled indoor environment. However, they still desired to think about how the flight system would operate in an uncontrolled outside environment. Though these are not essential requirements, the application of this constructed technology would be widened exponentially if it could handle the basic outside environment. These constraints could look like light to moderate wind speeds and general atmospheric turbulence.

Beyond these environmental factors, there are also some health and safety standards that are necessary to follow in order to prevent serious injuries. The Federal Aviation Administration has put out universal standards that must be followed in order to legally and safely fly a drone in any public area, regardless of

whether it is being flown indoors or outdoors. These constraints create little worrisome for this project due to the enclosed testing environment, but it is important to go through them as needed.

This chapter will take a deep dive into the physical, environmental, and administrative constraints and standards that they will need to address. The constraints of the system will directly lead to the development of a great deal of the defining features of the system. The group has learned how to address these challenges that are inherent when creating an autonomous drone. Even though they are confident that they have discovered nearly all of the major constraints and standards of the project, the group believes that if there are any more of these challenges that come up later on in the development process that they can swiftly and effectively confront and handle them

4.1 Physical Constraints

The team also needed to consider the physical constraints of the system, Given that a drone can be quite a sensitive system, there are many mass and weight distribution requirements that the group must face to construct a functioning drone. Considering the construction of the drone, the robot is constructed with four legs (or arms) with the PCB board in the middle connecting the four legs. The four legs will each have brushless motors attached to the top plane (or y-plane in space) of each leg.

On the top of each motor, there is a set of two propellers. The propellers are the mechanical aspect of the robot that catches the wind and allows movement in space and time. The quadcopter has a set of two propellers, two that connect clockwise and two that connect counterclockwise. These set of propellers are a pair by legs, therefore one pair of legs is connected via one node and set as the forward propellers and the other set of legs is connected via one node and set as the backwards propellers. The propellers will move at a high speed of anywhere between 8000rpm to 9000 rpm. The propellers are to be of the highest constraint because at this high speed and the novelcy of the team with robot building, it must be held to the utmost safety.

Table 4.1: List of all of the basic project constraints

4.1.1	The system has a weight of less than 10 lbs
4.1.2	The system has the correct permits to fly legally
4.1.4	The system must be able to remain stable during small gusts of wind
4.1.5	The system must follow the rules of the selected programming language(s) used throughout the software
4.1.6	The system must contain parts that are within a reasonable budget
4.1.7	The system must avoid putting humans in harm's way
4.1.8	The system has elements to protect the propellers, and by proxy protect people and objects around it from the propellers

Not alot of drone models are currently seen with propeller protector rings for the safety of human interaction. There has been instances described in research, where a human may be distracted in their research and not completely aware of their surroundings and have collided with the drone causing impactful injuries to the face that require immediate health care professional attention. For this reason, we are to only test the drone in its appropriate testing zone but to also put on propeller protector rings for the safety of all those within the lab of the testing zone.

The testing zone details are described in depth later on in this research paper. For visuals now, the testing zone is in an indoor area, on an elevated table with mesh netting around the table being held up by piping. This standard testing area is set up at a lab on the University campus, as promised from our original sponsor and now confirmed by Dr. Reza. This standard testing area will ensure safety as well as accurate testing zone since the zone will not be modified.

4.1.1 Propeller Safety Factor

The way our team will solve the propeller safety factor that will ensure safety is have propeller protector rings. Propellers are the aerological part of the robot that allows the drone to fly. The propellers are attached to the brushless

motors. There are four motors on the end of each leg of the drone with two propellers attached to the end of the motor, this allows flight upwards, downwards, forward and backwards. In order for the drone to fly forward, the flight controller speeds up the designated corresponding brushless motors which are attached to two out of the four legs of the drone that share the same node. The two legs that are used for movement must be connected in series. The two brushless motors that are attached to these corresponding legs move at a faster rate of about ten times while the two back motors stay at a slower speed or at the speed it was prior to forward / backwards / side command. This causes the two front propellers to go faster, therefore propelling the drone in the direction in space that was determined by the controller.

The propeller rings will circle around the propellers of each set on each electronic motor. This will create a barrier between the propellers and any outside objects. As explained above, the propellers must move at a high speed to take the robot from off from the ground. Safety is one of the group's greatest priorities. A drone is quite a sensitive system, and can easily lose control if one is not careful. With this in mind, the group decided that it is important to keep the speed of the drone at a minimum. The team does not predict that our drone will maintain a very high altitude during its testing. It is predicted that the drone should only be about to fly about two to four feet above the testing table for a maximum of 12 minutes.

4.1.2 Battery Constraints

In order to achieve the expected flight height while also achieving battery life, the battery is another key component needed for extensive research for reliability, durability and safety. Our team is looking to start with a battery that is around 2300-2300 KV with a 3s battery voltage applied to it. This will cause the motors attached to each leg of the drone to spin at about 28,980rpm. More calculations and research within the area will need to be done and should be the team's main focus moving forward.

The battery is the key component for reliability because each motor, the vision system using the camera, the control system and the microcontroller of a raspberry pi will all rely on its efficiency to provide the correct voltage to each component for proper flight control and safety. The battery will also be the component that is causing the most heat. These thermo waves do need to be controlled in order to ensure that the robot does not set on fire neither does it melt or burn any of the wires or components.

This aspect of the robot does require a project constraint of heat omission and will require great attention to prevent fires. For safety, the team will purchase a fire casing to enclose the battery chosen as well as a fireproof bag to place the drone in to prevent any spread of the fire. The team should also be aware of the closest fire extinguisher near the testing lab and have their devices in hand to call on emergency if needed. To address some of these concerns, there should also be at least two people in the testing lab at a time when performing testing.

Now that we have gone through the two major safety concerns of the robot and the constraint that is needed for the best safety measures. We would like to go into a bit more detail of each component that makes up the quadcopter. The first part of the robot that we will need to build is the frame. The frame is the main body of the drone, where all the parts are mounted together. To keep minimal weight, the drone is made from carbon fiber. The frame, as stated above, will consist of four legs connected by a rectangular body center.

Table 4.2: Drone frame size comparison

Propeller blade size	Drone frame Size	Leg/arm thickness
2 inches	95mm	2.5mm
3 inches	130mm	3mm
4 inches	180mm	3mm
6 inches	250mm	4mm

On the end of each leg of the quadcopter there are brushless electric motors. The motors are the powerhouses of the robot that give the quad copter the thrust and propulsions needed to provide speed for the propellers attached. To build the drone and ensure that the robot can be lifted from the ground, we must take important account of the specs in the motor that we will use. When choosing a type of motor and manufacturer we must look at the size of the motor. The size is typically noted in a XXYY format, with the XX referring to the stator diameter in millimeters and the YY referring to the height of the magnets inside of the motor. The larger each of these numbers are, the higher the torque the motor will produce. The larger the size of these, the larger the weight of the drone as well. For this reason, our motors have a small diameter and small magnet size which will result in a less heavy robot and slower robot, ensuring safety from the constraints explained above.

These motors are connected to the main flight control system by electronic speed controllers. The ESCs are the devices that will allow the Pihawk control system to control and adjust the speed of the electric motors by raising and lowering the voltage to the motor as required. This is what changes the speed of the propeller causing movement, speed and flight direction. Its purpose is to vary the electric motor's speed, direction and also acts as a dynamic brake.

When looking for a motor, we will look to find a specification table that gives details on thrust with different props and amp drawing. In general, we will follow a 10-1 thrust to weight gap ratio when building our drone.

Table 4.3: Size comparison of brushless motors with its corresponding components

Propeller size	Stator Size (diameter)	Magnet Height	Brushless Motor in KV	ESCs
2 inches	11	03-06	4000-8000	6-12 A
3 inches	13-14	06-07	3000-4000	12-20 A
4 inches	13-22	04-07	2400-2900	20 A
6 inches	20-23	06-08	2200-2800	30-40 A

4.1.3 Flight Controller (FC) Constraints

Now to get to the brain of the quadcopter, the flight controller. The flight controller of the drone will take into account the angle of the drone and will control the input to calculate how fast the motors should spin and send these signals to the electronic speed controllers or ESCs. Flight controllers are built for different softwares, and for our case we are using the Pixhawk flight controller because it is easily integrated with a Rasperrypi4 and the pi module camera. Software constraints for the flight controller are explained in the software section below.

This flight controller will not be the cheapest nor the most cutting edge development. The flight controller chosen is for novice drone builders that are also performing engineering research such as the one we are performing, vision navigation. For this reason, the flight controller has a good processor. The microprocessor will work hard to keep the robot in the air while also processing the visual input and enable decision making. With this, machine learning has to be coded onto the chips to enable proper decision making. This is why we are predicting the flight controller to be one of the most expensive components on our drone. This will ensure proper safety and control of the aerial machine.

The flight controller and the electrical components mentioned above are connected to a Power Distribution Board (PDB). The PDB will intake the battery voltage provided and is used as a layout to set various points for the team to connect up the electronics. The PD board will feature regulatory power switches

to lower voltage to certain components. There are many things to consider when choosing the appropriate PDB. Such as voltage requirements, connector locations and the maximum current draw to each electronic device location. The constraints determined for each of the requirements stated above will also be key in reliability, efficiency and safety of the drone.

For the voltage requirements of the PDB board, it is important to consider that the flight controller will require about 5V to run off and the camera can require another 5-12V, this is a high voltage that cannot be connected directly to the battery without causing a fire due to the constant high voltage transmission, giving the essential constraint of needing a Power Distribution Board to connect the devices too. The PDB will also need to contain voltage regulators or Battery Eliminator Circuits to provide the power output to the designated port that is needed. When planning the drone build, the team has to try to visualize where we want to put the electrical devices and if the pads of the PDB are actually where we want there.

4.1.4 Visual Constraints

Now moving along to the eyes of the drone, it's visual system, which starts with the input signals received from a camera that is attached to the drone. There are two cameras attached to the drone, one facing the z axis of the drone and one facing the x axis of the drone. This will allow the drone to see forwards and below it. For our vision navigation drone test, we are requiring the drone to stop at the vision obstruction of an obstacle and also find its landing target. This is the reason why we choose to use two drones. One to detect obstacles, such as balloons in its x axis view and another to find its proper landing target labeled on the target below it. This vision navigation machine learning technique is expanded more within the optics and software sections respectively.

When choosing a camera, it is important to consider the constraints on sensor type, pixel resolution and latency, the built in camera features, the lens and the video transmitter that is attached. For vision navigation drones, the cameras usually have a CMOS or a CCD image sensor inside. CMOS cameras are less expensive than CCD cameras but lack the ability to react quickly to changes in light. This is important to take into account when choosing the camera for our drone because any sudden change in light that causes a lack of visibility can cause confusion within the drone flight controller system and cause a crash. A CCD camera will give us the best result for a vision navigation drone, which uses a Sony Super HAD II sensor which is one of the best in the market for vision navigation drones. On the topic of resolution, the higher the resolution, the slower the latency so this is also a large topic to dive deeper in when looking at the type of resolution we would like our vision navigation drone to have.

The group also wishes to create a testing environment for the drone. This setup will let the group run the many tests required to train the AI. Understanding what is needed to create a specific testing environment is essential to understanding how the drone should fly. The constraints of the testing

environment are very important to understand when developing the system as well as important for safety measures.

Table 4.4: The parts needed for the training zone that is set up in the lab:

4.4.1	Green Trable
4.4.2	Netting that will surround the testing environment, this will ensure the safety of anyone in the room if the drone loses control
4.4.4	Tubes to hold up netting. This will ensure the structural integrity of the nets. Having nets move around as little as possible reduces risk of drone colliding with them
4.4.5	Environment model. This will simulate an environment a drone may find itself in. It helps train the AI to differentiate between the target and the local environment
4.4.6	Block targets. These will act as the primary targets for the drone to move around
4.4.7	String/rope to hang targets from the ceiling

Table 4.5: The parts need to build the drone with the parts' approximate weight:

4.4.1	Drone Frame	60g-140g
4.4.2	Four Brushless Motors	25g each 100g total
4.4.3	Eight Propellers (four for backup)	17.3g each set of four
4.4.4	Propeller protector rings	12g
4.4.5	Four Electric Speed Controller	4-6g
4.4.6	Power Distribution Board	25g
4.4.7	Flight Controller	20g
4.4.8	Battery	170g-270g
4.4.9	Battery Charger	30g (will not be attached to drone)
4.4.10	Battery Fire-proof case	40g
4.4.11	SD card and SD Reader	315g
4.4.12	Camera	734g
4.4.13	RC Video transmitter	453g=1lb (not attached to drone)
4.4.14	Sensors	5.8g
4.4.15	Wires	7.5g
4.4.16	Raspberry Pi 4B	46g
4.4.17	Vibration Dampening Plate(3D printed)	unknown
4.4.18	GPS and Compass Module	18g
4.4.19	GPS mount	20g
4.4.20	Screws	9.4g
4.4.21	Screw driver	N/A
4.4.22	940 nm LED	9g
4.4.23	NIR Photodiode	9g
4.4.24	LM358 Op Amp	N/A

4.2 Data Sheet Material

The data sheet below describes the specifications of the drone system as well as the conditions necessary for successful testing. These measurements include the specifications for the mesh cage testing environment as well as the capabilities of the drone itself. Utilizing these parameters is important when it comes to determining area coverage using the camera which can speed up overall object detection. The table also describes the takeoff and idle altitude while in the testing environment.

Table 4.6: Technical Specifications and operational conditions

Specification	Value
Technical Specifications	
Climb rate	2 m/s
Cruising speed	3 m/s
Peak thrust	7.5 KG
Vehicle mass	2566 g
Recommended payload mass	TBD
Maximum payload mass	TBD
Dimensions	1.5m between opposite rotor shafts
Flight time	10-12 minutes
Operational Conditions	
Flight radius	6 feet by 8 feet
Ceiling altitude	12 feet - ceiling of lab at campus
Take-off altitude	4 feet on table of testing area

The team also wanted to organize all of the types of data that the drone would be dealing with during the flight. Table 11.2 describes these settings in detail and some of the ways the data is transmitted.

Table 4.7: Data required and settings used for experiment

Data type	Setting value
Flight area	48 Square Feet at 4' Altitude
Width	6 feet
Length	8 feet
Camera Specifications	Pi module v4 camera - Sony IMX477R stacked, back-illuminated sensor, 12.3 megapixels, 7.9 mm sensor diagonal, 1.55 µm × 1.55 µm pixel size
Output	RAW 12/10/8, COMP8
Back focus	Adjustable(12.6 mm - 22.4 mm)
Lans standards	C-mount, CS-mount (C-CS adapter included)
IR cut filter	Integrated
Ribbon cable length	200 mm
Tripod mount	1/4"-20
Distance Data From image detection system	Too close or safe
Power consumption	10 W
Processor Platform	ARM v9

4.3 Health and Safety Constraints

For any type of drone, there are many regulations that must be followed to safely fly in specific areas. Here are some basic standards that must be followed when the drone is in operation, regardless of autonomous piloting or not:

- Fly at or below 400 feet

- Keep your drone within sight

- Don't fly in restricted airspace

- Don't fly near other aircraft, especially near airports

- Don't fly over groups of people

- Don't fly over stadiums or sporting events

- Don't fly near emergency response efforts such as fires

- Don't fly under the influence

There is a future implementation of a remote ID for drones, but for this project and during this timeframe, it is not needed. In terms of UCF requirements, permission from the school is required to fly in any area on campus. Although the drone shall be tested in a safe area within a lab or room, permission is still necessary in order to build and conduct testing within the environment.

4.4 Test Environment Standards

Testing consists of three separate stages: image classification, simulation, and real-world. We have structured our testing plan in this way to reduce more costly faults that may occur within the real-world testing stage. This will also prove to be the fastest way to get our desired results, as simulation gives us infinitely more control over the way in which we train our drone. Following successful simulation runs, the group is able to construct and practice running the drone in a real life environment. Since the drone has much of its training completed, the group is able to have far more confidence that the drone will perform well. A greater level of performance in the beginning of real world testing will result in fewer crash incidents. This prevents too much damage from happening to the drone. Extensive damages that need repairs can easily drive up the final cost of the system.

4.5 Image Classification

The first set of trials is based around training the system to identify objects through the video obtained from its camera module. The convolutional neural network is fed a large dataset full of a variety of images, trained using that data,

and then testing can be conducted with recorded video of our obstacle course. The input videos would be of consistent length with each being treated as a separate test subject. We will attempt to reach high prediction rates on each of the videos to ensure that we will not have any trouble moving forward with the navigation portion of the testing.

4.6 Simulation

Initial testing is conducted within the Gazebo simulation environment. This ensures consistent control over all variables. A significant portion of the testing is done in the navigation training section. The environment will consist of a caged room with 3 floating balloon objects of varying colors. We will run the simulation with the drone at the same starting point and end point, feed it the desired route until the system can navigate it with satisfactory results. Satisfactory results for us means that the drone does not stray far off the optimal route with the least distance traveled, does not collide with the obstacles, and does it in a timely manner. If we are successful in this, we hope to advance past that stage and move objects around, alter their colors, and even alter the start and end point. Once we have succeeded in this portion, we can transition to testing in the real obstacle course.

4.7 Real-World Testing

In order to effectively test the drone's functions as well as develop a stable learning mechanism for the drone to detect obstacles, an efficient test environment is to be created. The test environment shall consist of the following:

- **Mesh Cage**

- **Three Colored Obstacles**

The mesh cage shall be constructed in order to meet safety requirements within the testing space and to protect the conductors from failed tests if the drone were to steer off in the wrong direction. The mesh is lightweight to prevent any damage to the drone should the testing process go wrong.

The obstacles, which will also be used for the end-of-year demonstration, will consist of three unique objects, potentially differing in color, which shall be tied to the top of the cage and hung down for the drone to detect and fly to based on the given commands. There may be obstacles placed on the ground for low-altitude testing, if applicable.
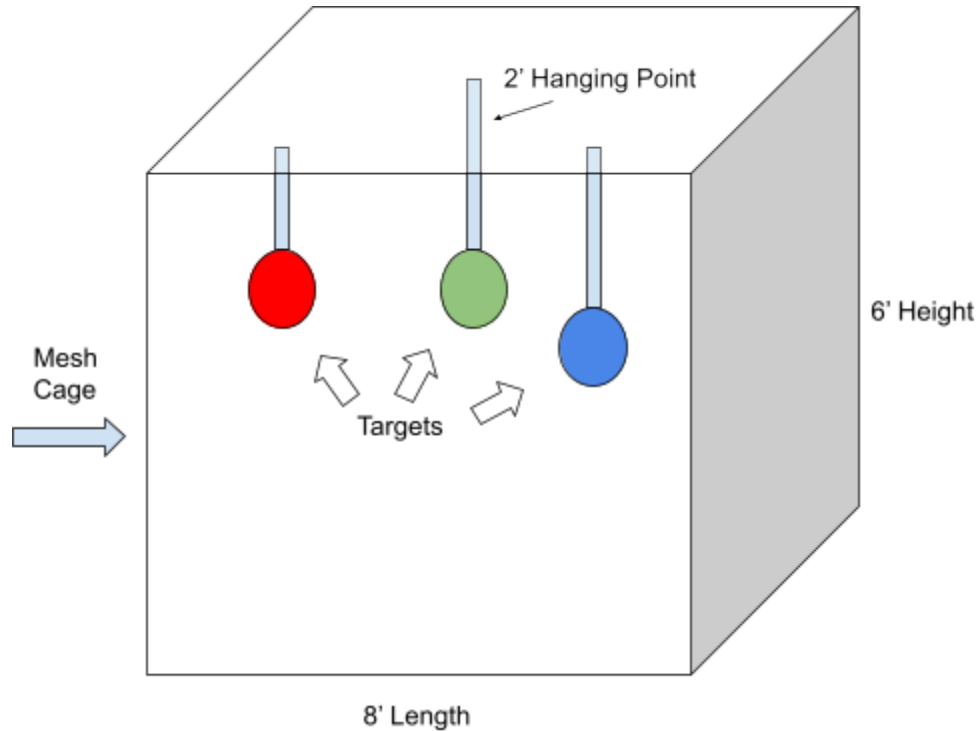
Figure 4.7.1: Schematic of the testing environment

4.8 Video Demonstration

For the demonstration, the test shall be recorded using the drone's camera (Raspberry Pi). The demonstration will consist of the instructions provided in Section 17.0 and will also utilize the real-world test environment as described in Section 16.3. Since the showcase will not allow the opportunity to present a real-time demonstration, the video must ensure that all requirements are met and that the drone completes a successful flight. At the time of demonstration, the drone shall be completed in its entirety and provide the observers a detailed perspective of the drone's full functionality.

4.9 Testing Instructions

4.9.1 Flight Test

In order to have a consistent testing method, the team wished to outline the basic steps that the drone will typically take during its runtime for any test trials. This basic series of steps will provide some of the foundation to the instructions to the code. With a fundamental pattern hard coded into the system, the group should be able to predict what the drone will do.

Table 4.9.1: Basic flight testing process

4.9.1	Power on the Drone
4.9.2	Enable Drone to Reach Required Altitude within the Mesh Cage
4.9.3	Turn on Camera
4.9.4	Provide Data to Test Object Detection via Software
4.9.5	Ensure that Drone Completes the Path Correctly
4.9.6	Return Drone to Initial Location
4.9.7	Return to Ground Level and Power off the Drone

4.10 Object Detection Test

The group also wished to have a basic system similar to the flight testing process, but for image recognition testing. Table 17.2.1 defines an outline of the steps that the object detection software will take after it is activated when the drone is in flight. This consistent process may allow the group to determine exactly where a bug occurs. From there, the group can efficiently and effectively find areas of the code that must be fixed.

Table 4.10.1: Basic process of the object detection test

4.10.1	Determine if Drone Detects Objects within the Software
4.10.2	Ensure that Drone Flies to Correct Object in the Correct Order
4.10.3	Confirm that the Entire Desired Flight Path is Detected Post-Flight
4.10.4	Ensure that the Sensing System is Off when Drone is Turned Off

5 Design

5.1 Reward Based Machine Learning

The software component of the project consists of three parts: training, simulation, and real-world application. Training is done through the collective effort of three softwares: PyTorch, Gazebo, and PX4. PyTorch is used to create the convolutional neural network that handles the image classification during the navigation through our pre-mapped course.

5.1.1 Training

We will first train the CNN with a large open-source data set inside of PyTorch. Providing rewards whenever a successful state is reached while training our CNN will hone its image classification skills up to whatever standard we want to employ. After so many runs of configuring our CNN and achieving a satisfactory image classification rate, we can then link our CNN to the simulation portion.
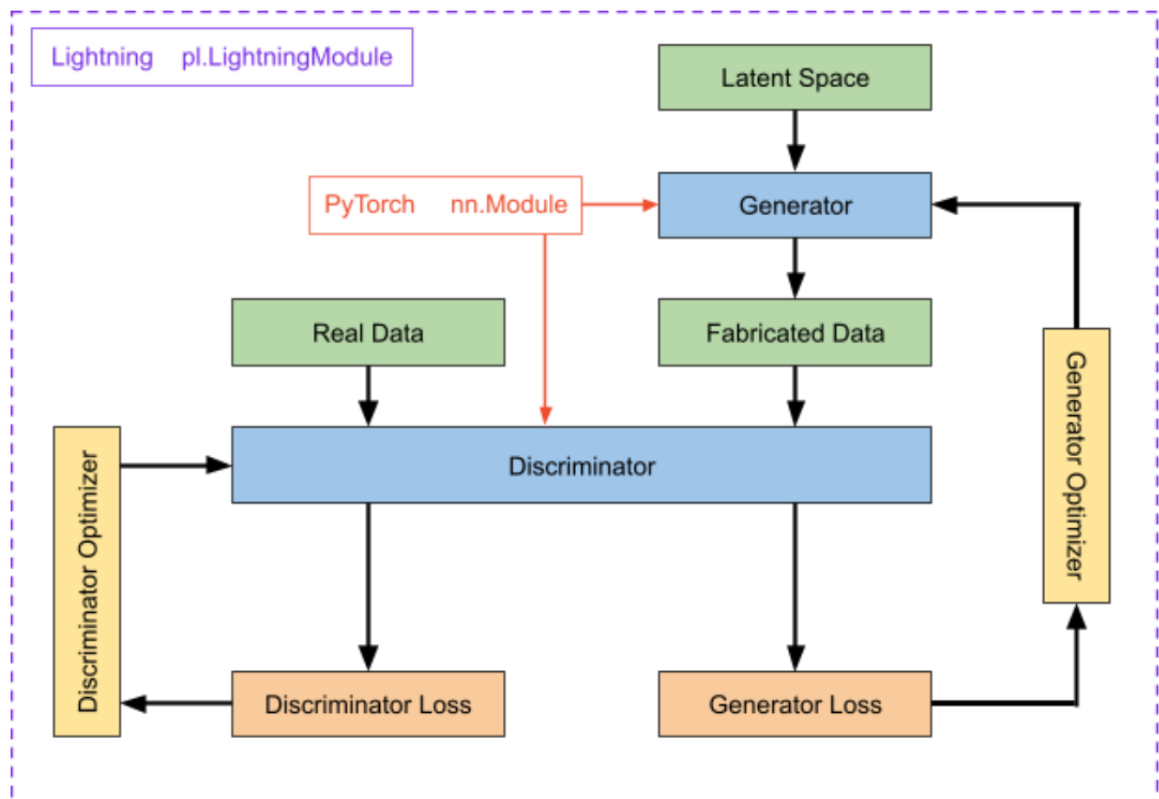

Figure 5.1.1: An Example Training Loop using PyTorch Lightning

For our CNN training model, we have ultimately decided on the YOLOv8 model. We have chosen this model because it covers exactly what we are looking for in an object detection model. The model is credible and is easy to

implement since there are plenty of open-source notebooks paired with packages and articles going into detail about the functionality of the algorithm. The Roboflow platform also lets us create our own annotated dataset. Any changes we then make to the code can be reflected on since we then have a reference of how the model should function. Everything before deployment is essentially done with online resources, which saves us from losing any progress if something were to corrupt. This also saves resources as we can train our model overnight without having to monitor it if necessary. There are almost too many benefits for utilizing this model and platform. YOLO is top of the line for real-time object detection so this had to be our best choice going forward.

Training essentially ends there, but we do have the stretch goal of now including navigation and collision-avoidance. Rewarding the drone based on how fast it may move through a course, or when the drone takes a favorable path. Training is where a majority of the coding is done, all of which is done in Python. We may also alter the frame rate or resolution to test if we get higher recognition rates or not. Depending on those results, we can really optimize our system to maximize performance and accuracy. This would be a much simpler algorithm that we would run many times more than our object detection, as it includes much more free will with its access to a 3D space.

5.1.2 Simulation

First, we are experimenting with the control system of PX4 within the simulation environment. Movement is basic, as altitude will not be a variable within our project, but gaining an understanding of the physics environment and how to track velocity/position components of the drone is extremely important. Given that Gazebo and PX4 support each other, the resources they provide within their Guides/APIs is essential to furthering our understanding of how the drone reacts to its environment. Creating as close to a real-world environment will also be key to have the drone transition out of the simulation as effectively as possible. Improving our success.Monitoring every component of the simulated drone gives more opportunities for "rewards" to then improve our predictions. Cameras and sensors being simulated gives us a huge advantage as well, as long as they are one-to-one representations of real-world field of views and sensing capabilities. Otherwise, this may cause our predictions to be incorrect and under/overestimate the abilities of the drone.

On the following page is Figure 5.1.2 that describes the general flow of a general PX4 simulation protocol. This shows the logical progression of communication that allows the system to train effectively
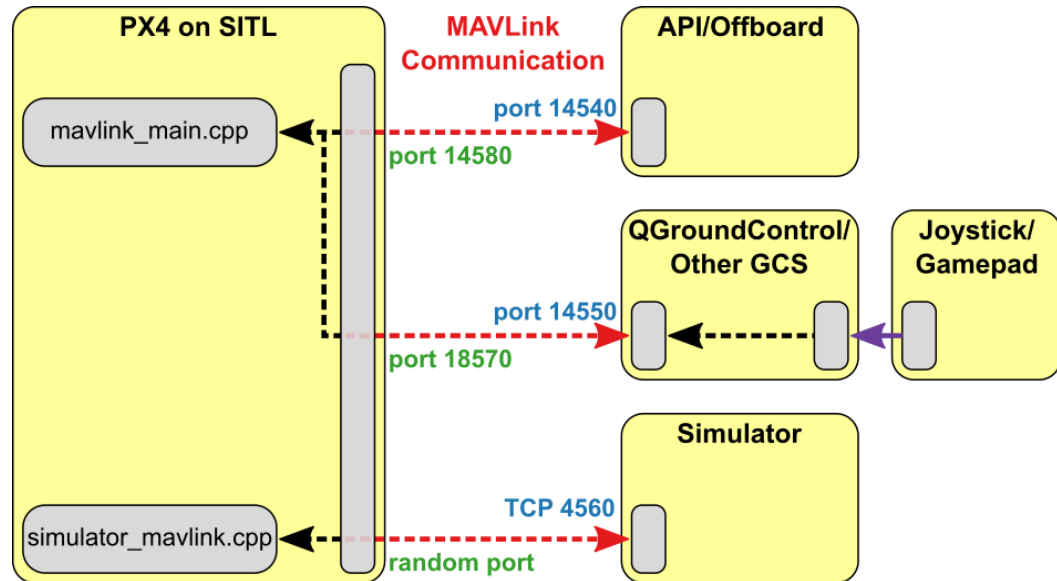
Figure 5.1.2 Flowchart of a General PX4 Simulation Protocol

5.1.3 Real-World Application

Transitioning to real flight is the final step for our software development. Ensuring that the resulting trained model stands up to its real-life counterpart is the deciding factor for this project. With very minimum environmental effects to take into consideration, the simulation should provide an authentic enough experience. Mapping our obstacle course accurately will obviously result in a more "prepared" state. Hardware capabilities may limit our ability to train on the drone itself so accurate mapping is even more important to the process.

5.2 Raspberry PI Integration

We have decided on utilizing RaspberryPi 4 Model B to handle our PyTorch trained model. We chose this version because of its out of box support of PyTorch This makes the build process much smoother for us because it alleviates a lot of the hassle of working with plugins or open source projects to get fully-accessible PyTorch version onto our drone. With out of box support, and Raspberry Pi camera module we selected, it also gives us the ability for "real-time inferences."

5.2.1 Post-Training

After training is completed, the model is then deployed on to the Raspberry PI. We then have onboard image processing and classification, getting full use of our convolutional neural network. The setup for this is to first install the 64-bit ARM Raspberry PI OS onto the board computer; PyTorch will only support 64-bit ARM so this is required. Once the OS is set up, our package manager MiniConda and package installer Pip are installed. We use these to then install the Raspberry Pi Docker tool which is utilized in tandem with a

package we will install. All of our heavy impact packages are then installed, which include: Roboflow, PyDrive, OpenCV-Python, RPi.GPIO. The model is downloaded using a function from the Roboflow package. The Roboflow package uses the Docker tool to connect to the Roboflow server in order to grab our model.

5.3 Low-Level (Firmware) Programming

This type of programming is used for the drone's firmware, which basically coordinates the hardware aspects of the drone such as the motor(s), propellers, ESC, battery, etc. The drone's firmware allows the drone to have its basic functionalities and operations. The firmware, for example, handles things like determining the exact amount of power that should be delivered to the motors by analyzing the information coming from the drone's IMU (Inertial Measurement Unit). This will allow the drone to perform a stable and level flight, the most basic and yet most important function of any drone.

For this level of programming, C and C++ are mainly utilized. Additionally, some drones may utilize a low-level machine language such as Assembly, depending on the desired functionality. The Raspberry Pi module, which is used for this drone, will utilize low-level programming in order to function.

Type of Language	Example Language	Description	Example Instructions
High-level Language	Python, Visual Basic, Java, C++	Independent of hardware (portable). Translated using either a compiler or interpreter. One statement translates into many machine code instructions.	payRate = 7.38 Hours = 37.5 Salary = payRate * Hours
Low-level Language	Assembly Language	Translated using an assembler. One statement translates into one machine code instruction.	LDA181 ADD93 STO185
	Machine Code	Executable binary code produced either by a compiler, interpreter or assembler.	1010100011010101010100100101 010101

Figure 5.3.1: A table describing the implementations of low-level and high-level programming

5.4 High-Level Programming

In this type of programming, more complex languages are used such as Python and Java, to provide functionality to the desired applications, such as speed and altitude, for the drone. Functions like controlling the drone to a certain altitude via the Flight Controller (FC) or interpreting information from GPS so the drone can move to a GPS waypoint, and even increasing/decreasing speed are handled by high-level programming. API is used to link the code to the drone's interface

which will help with object detection as well as determining the flight instructions, which shall be completely dependent on the color, shape, and size of the desired object.
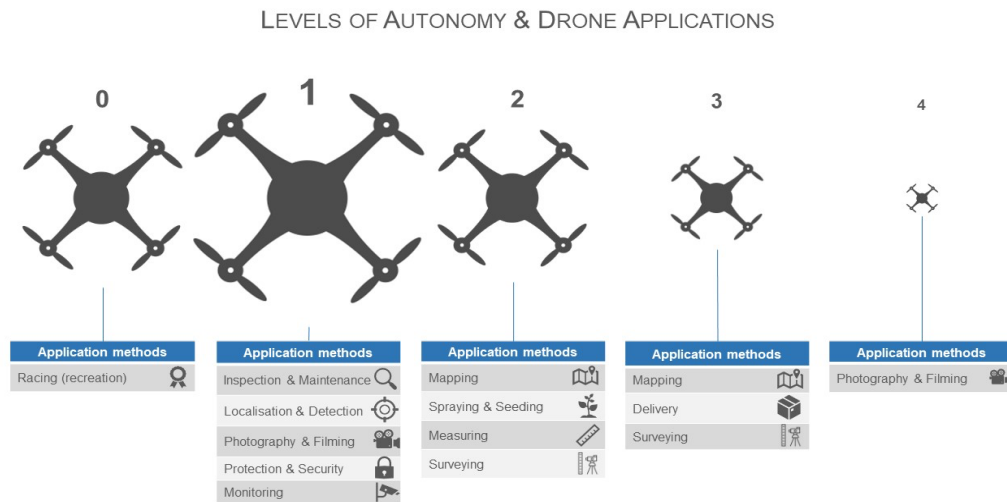
LEVELS OF AUTONOMY & DRONE APPLICATIONS

0	1	2	3	4
Application methods	**Application methods**	**Application methods**	**Application methods**	**Application methods**
Racing (recreation)	Inspection & Maintenance	Mapping	Mapping	Photography & Filming
	Localisation & Detection	Spraying & Seeding	Delivery	
	Photography & Filming	Measuring	Surveying	
	Protection & Security	Surveying		
	Monitoring			

Figure 5.4.1: A table describing the types of autonomous drones and how high-level programming has different functionalities for different purposes

5.5 Optical Design

5.5.1 Motivation and Methods

The team members wanted to prevent the drone from colliding with objects as it flew around an area. The group decided that the color camera used for the object acquisition alone was not enough to reliably detect whether environmental objects were getting too close to the drone. They believed that this would require too much computation. The images from the video would need to be segmented, objects from all over the image would need to be isolated, and then these objects would need to be analyzed over several frames to see how the size and position of all of them change as the drone moves through space. Only then can the drone do even more computations and identify a specific object, or set of objects as an immediate threat.

The problems with processing the camera's images for object avoidance are twofold. For one, this lengthy process diverts computational resources away from the image detection, which needs as much of the small computer's processing power as it could be given. Not only that, but in the time it would take to process and analyze all of this data and determine if a specific object is a threat, the drone may have already gone too far and collided with something. It simply may not be fast enough to respond to a threat. When threats are present, the drone must be able to respond as quickly as possible. Such a lapse in decision time could result in the destruction of parts of the drone as it collides

with the environment. The parts of the drone frame can become quite expensive. Crashing the drone would drive up the final costs of the project considerably. Therefore, damage to the drone must be avoided whenever possible. This is the primary motivation for this optical detection system.

Analyzing the distance from an object using screenshots of an image would require a lot of processing power, and the group wanted to avoid this intensive approach. So, the plans for an independent optical detection system that could determine if the drone was close to collision was devised. This could drastically reduce the object detection processing requirement, and allow for most of the processing power to be allocated toward the image recognition software. This system would, in its simplest form, send a direct warning to the computer that an object has become too close to the drone. This could then be used to directly command the drone to stop its movement, and likely faster than the image analyzing tool could do

A Light Detection and Ranging (LiDAR) system is a method of using electromagnetic radiation, typically within wavelengths the near infrared range, to detect the presence of nearby and far away objects. This range, usually between 750 nm and 1500 nm, has several key advantages over other possible wavelengths of light. Firstly, it is a range that the human eye cannot see. So, it will not interfere directly with people's daily life. Also, it can be done with less powerful beams since other wavelengths of light can be filtered out. The atmosphere is also quite transparent to these wavelengths of light, and they are not used in other forms of communication. This means the beams will not interfere with any communication systems such as radio, Wi-Fi, and bluetooth.

The final system design has several components that can sense the environment immediately around the drone. The primary pr prevent damage to the drone or others. Its object avoidance system has two primary optical elements that will help it navigate its environment. An object detection sponsor (or time of flight LiDAR scanner) that is used for object avoidance as well as a full color camera that is used for the image recognition algorithm.

The minimum requirement for the group is to develop a system that is able to detect whether the drone is becoming too close to an object. This object detection sensor will use a near infrared LED and photodiode to detect reflected light. Several of these sensors could be implemented around the drone to detect dangerously close objects from several directions. It should be able to at least detect a white object a foot and a half away. To give the drone some time to react and stop its movement if necessary.

Figure 5.5.1 An example of infrared sensor being used to avoid obstacles

If time and resources permit, and the group wishes to meet some of their advanced goals, then they will develop a time-of-flight sensor, which can analyze the exact distance an object is away from a specific section of the drone. If it performs well enough, this would give the group more accurate information, and could be used in tandem with the image recognition system to tell exactly how far away the drone is from the target object.

Furthermore, if the group finds the ability to reach their stretch goals, they will adapt this time of flight design into a full 360 degree lidar scanner. This system will contain at least two time of flight sensors that spin in a circle, taking data constantly. Data is taken from these sensors and creates a two dimensional map of the area immediately around the drone. This data would be used to recognize incoming objects and even determine the distance away from the target the drone is. Theoretically, this could also be used to create a full map of the environment.

Another goal of this design is to create a device that is cheaper than commercially available alternatives. If the group decides to reach its stretch goal of a full two dimensional LiDAR scanner, then it should have a competitive cost. The cheapest 360 degree LiDAR scanners that are publicly available cost about $100. So, one of the priorities of building a scanner is to make its total cost significantly lower than this.

5.5.2 Object Detection Proximity Sensors

An object detection scanner is used to have the drone detect objects in the drone's immediate environment in order to avoid collisions. These systems use an active imaging method of remote detection that emits light. Infrared light is often chosen because humans cannot see it, and sensors can be made readily available. They send out pulses of light that reflect off of the environment around them. Some of these will naturally bounce off in many directions, but others will return to the senor. As a beam spreads out it will naturally lose power, so the

detector must be sensitive to the specific wavelength used by the emitter.This power can be determined by analyzing the solid angle of the object one wishes to detect relative to the sensor's location. One can use this size to calculate the expected power returning from a given reflection by analyzing how long it takes the pulse pattern to return.

An important factor to consider when creating a LiDAR system is the power of the beam and the sensitivity of the sensor. Since any beam sent out will diverge, only a small fraction of the power sent out will hit the object one wants to measure. The power will decrease as a function of distance, and the farther the object away is, the less power. The rate of the power loss depends on the angle of the beam divergence from the optical axis. Due to the Law of Reflection, the light will reflect back at the same angle, and the calculations to get the power returned is quite similar. These are all assuming an ideal scenario, and the surface of the object is perfectly flat with the surface normal vector facing the sensor. Despite this, these assumptions are still useful for getting a close approximation to the power one should expect to see return from the sensor. Once one has their desired range, they can adjust the power of the beam such that their chosen sensor can detect it. They could also change the sensitivity of the sensor, but this may be more difficult to do.

One must also consider the effects of the magnitude of beam divergence on the resolution. If one wants to detect an object far away, they may need to have a very small beam divergence. If an object were closer, a much wider divergence could give a similar resolution.
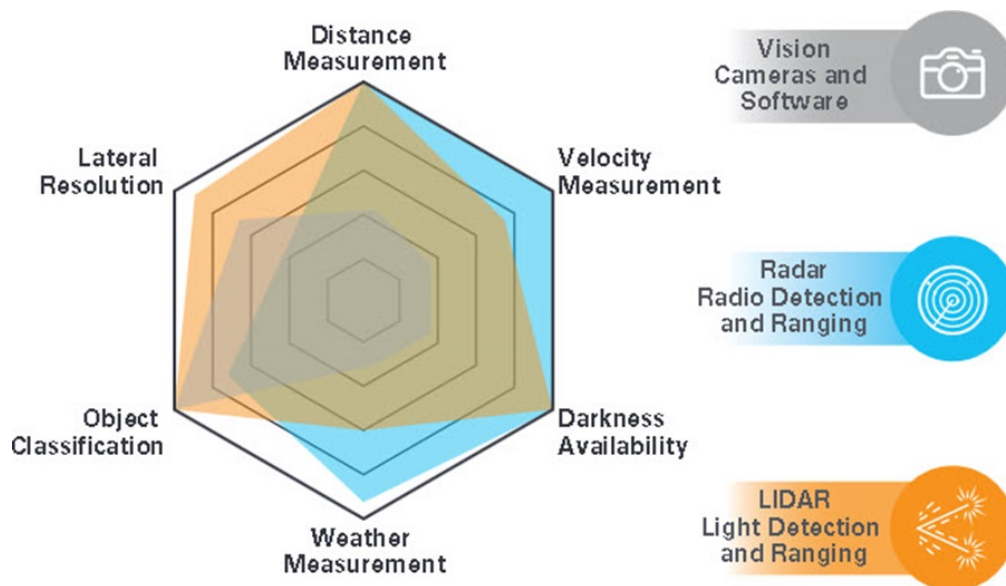

Figure 5.5.2.1: A comparison graphic between LiDAR sensing, typical cameras and radar/radio detection

The team needed to decide between two types of LiDAR scanners: one that can only see in one plane, or one that can create a full map of an environment. After some thought, the team decided to go with the 360 degree flat plane scanner. A LiDAR device that is capable of a complete room scan can cost thousands of dollars even for the cheapest models. This is far beyond what the team considers to be within a reasonable budget. A flat plane scanner is far cheaper, and may be just enough for the project.

The group had to consider the implications of having a horizontal only scanner. Since the chosen LiDAR scanner can only measure in a plane that is parallel to the drone, the team will supplement the blindness in the vertical direction with a simple rangefinder or an altimeter to assess the distance from the ground at any time.

Furthermore, since a drone will naturally tilt away from the horizontal plane in order to move around, the flat plane LiDAR system will need to be on a gyroscope to ensure it is always level with the ground. The height measurement tool could also be integrated into this design.
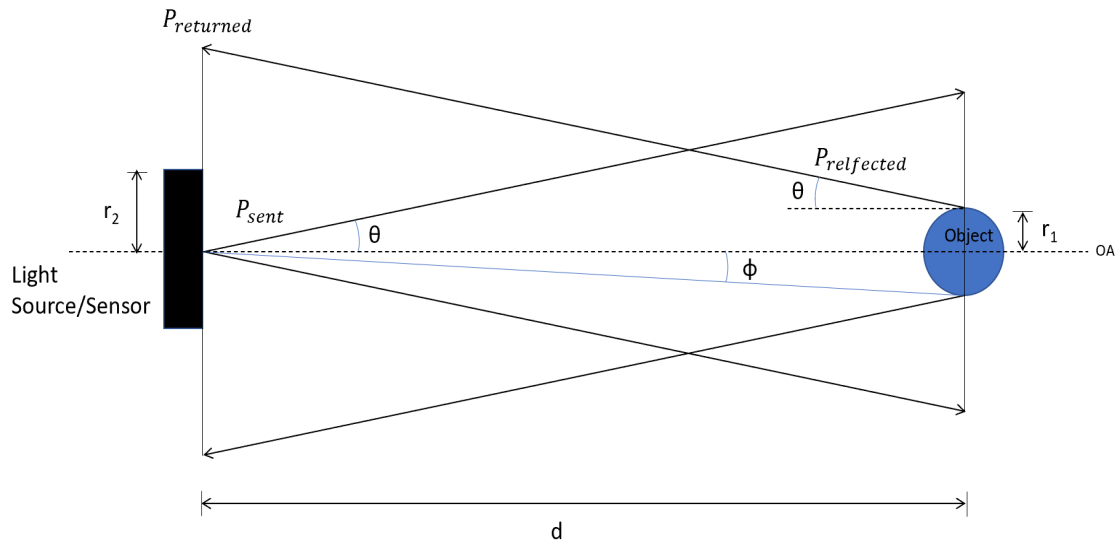
Figure 5.5.2.2: A basic diagram of light being emitted from a device, the light reflecting off of the object, and returning to the sensor.

The power reaching the object is a function of the initial power (Pi), the relative size of the object, reflection coefficient(R), and the distance. The power that the object receives is determined by the beam divergence (θ) and the angle that the object takes up in the sensors field of view (ϕ).

$$P_{object} = P_{initial} \frac{Solid\ angle\ of\ object}{Solid\ angle\ of\ beam} = P_i \frac{2\pi(1-cos(\phi_1))}{2\pi(1-cos(\Theta))} = P_i \frac{1-cos(\phi_1)}{1-cos(\Theta)} \quad (5.5.1)$$

The power reflected off of the object is found by multiplying this power by the fresnel reflection coefficient of a material.

$$P_{reflected} = R \cdot P_{object} \qquad (5.5.2)$$

And finally we can approximate the final receive power by using a similar equation to Eq. 5.5.1

$$P_{received} = P_{reflected} \cdot \frac{1-cos(\phi_2)}{1-cos(\Theta)} \qquad (5.5.3)$$

Combining these together give a general expression for the approximate power reaching the

$$P_{received} = P_{initial} \cdot \frac{1-cos(\phi_1)}{1-cos(\Theta)} \cdot \frac{1-cos(\phi_2)}{1-cos(\Theta)} \qquad (5.5.4)$$

It is important to note that the variables ϕ_1 and ϕ_2 are both functions of the distance between the sensor and the object (d) and the respective sizes of each (r_1 and r_2)

$$\phi_1 = arctan(\frac{r_1}{d}) \text{ and } \phi_2 = arctan(\frac{r_2}{d}) \qquad (5.5.5)$$

From this, a model object detection system was created using a breadboard

As seen in the Breadboard Testing section, this system is made up of a 940 nm LED and NIR photodiode, and a variable resistor. By increasing the resistance using this piece, the sensitivity of the sensor could be changed. This means that the distance at which an object could be detected can be easily changed by varying this resistance. The example mentioned is able to reliably detect whether there is an object that is 30 cm away or closer.

For the midterm demonstration, a simple object detection system was created using these outlined principles. A simplified diagram of the completed setup is shown in Figure 5.5.2. It has a simple setup where a near infrared LED (wavelength of 940 nm) is shined onto a surface, and the amplitude of the reflection determines whether the system will detect an object or not.
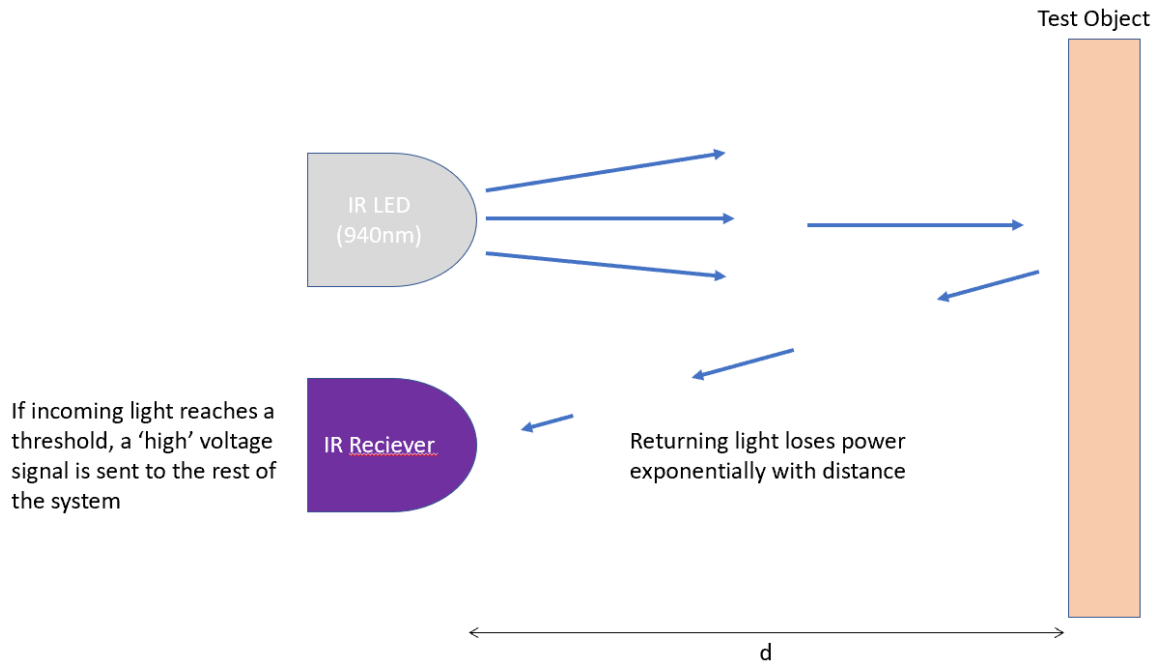
Figure 5.5.2: LED and Receivers reacting to light reflection

5.5.3 Time of Flight Sensors

Continuous wave time-of-flight sensors have several benefits over the simple object detection sensors that have been described previously. They measure the time (or phase) difference of transmitted and received signals rather than rely on its brightness. This allows them to more accurately detect objects that are not perfect reflectors. These devices can measure exact distances from the edges of the drone. This more accurate information would be used by the drone to more effectively detect nearby objects and react accordingly

By analyzing the direction the beams were sent out at, the distortion of the signal, and the timing of the pulses, the device is able to determine exactly how far away a given point in the environment is from the sensor. The two most basic metrics one must consider when searching for such devices are the range and resolution. There are several important factors that influence these characteristics: the power of the beam, the beam divergence, the desired range of the device, and the sensitivity of the sensor.

If the team were to mature this technology, then a system could be established such that it can be mounted on a rotating component. Data is collected continuously, and a two dimensional map of the area immediately around the system would then be constructed for the drone to use for obstacle avoidance. This stretch goal is not necessary for success, but it is something the team could benefit from greatly. It would allow for very fast responses to objects coming from any direction.

5.5.3.1 Direct Time of Flight

Direct time of flight sensors operate on a rather simple principle. They send out a pulse (or series of pulses) of light, and record the time it takes to reach the object, reflect, and return to the sensor. Often, near infrared light (between 900 nm and 1000 nm) that the atmosphere is transparent to is used in these sensors for maximum transmitting distance.

The speed of light in air is approximately three hundred thousand kilometers per second. It would take a pulse of light only 3 nanoseconds (3×10^{-9} seconds) to travel one meter. So, naturally, this direct time of flight process requires an extraordinarily fast driving circuit and signal detector. Most affordable components cannot easily be manipulated to modulate this fast. Given the nature of a direct time of flight circuit, there would be a larger minimum detection range due to the limitations of the technology available. The team could only acquire components that are rated for a certain speed, so this would be the limiting factor in the development of such a circuit.

The drone must be able to detect nearby objects, and the components that could be acquired easily may only be able to detect objects that are farther away. Given that the purpose of this design is to create a detection system of nearby objects, the ability to only see far away objects is not very useful to the team. With all of this in mind, the group wanted to explore a different method of distance detection where the transmitted and received signals can be measured with a more reasonable driving circuit before making a final decision.

5.5.3.2 Indirect TIme of Flight

Indirect time of flight sensing is a slightly more sophisticated method of distance detection. This method sends out a periodic signal, and the period of the signal is on the order of nanoseconds. This signal travels outward and reflects back to the sensor. The sensor will pick up the reflected signal. By comparing the incoming signal directly with the outgoing signal, the device would be able to measure the difference in phase between the incoming and outgoing signal. The difference in phase is directly related to the distance that the pulse has traveled. Figure 5.5.3 shows that with the square wave pulses being generated by the modulating LED, there is a time delay in the received signal from the sent signal. This will result in a certain phase difference between the two beams. A reliable method to measure this phase difference must be created. The goal of the design is to correlate this measured phase difference with the time the light had spent traveling back and forth. Once this is quantified, it can then be directly correlated with the distance the light has traveled since. The speed of light is always constant (in standard temperature and pressure)
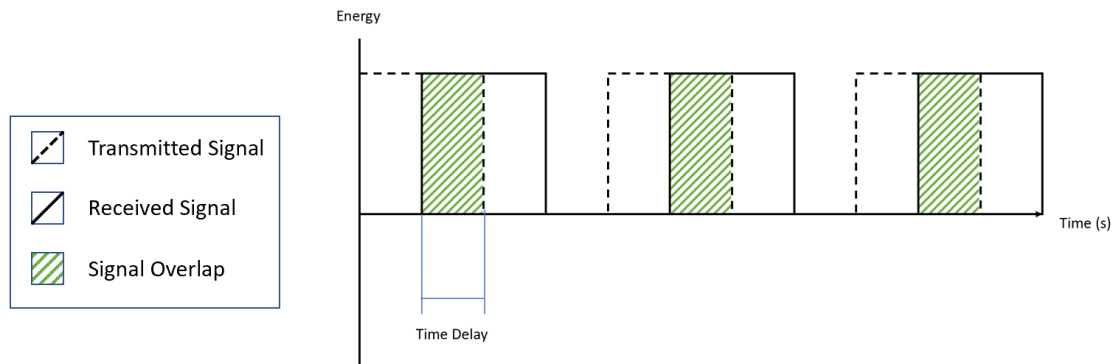
Figure 5.5.3: The time delay between the sent and received signals

Equation 5.5.6 describes the distance to the object using the relative phase difference. The 1/2 term is there since $\frac{c\phi}{2\pi f}$ is the total distance traveled, and half of that would be the distance to the object being examined. In Figure 5.5.2, once can see that
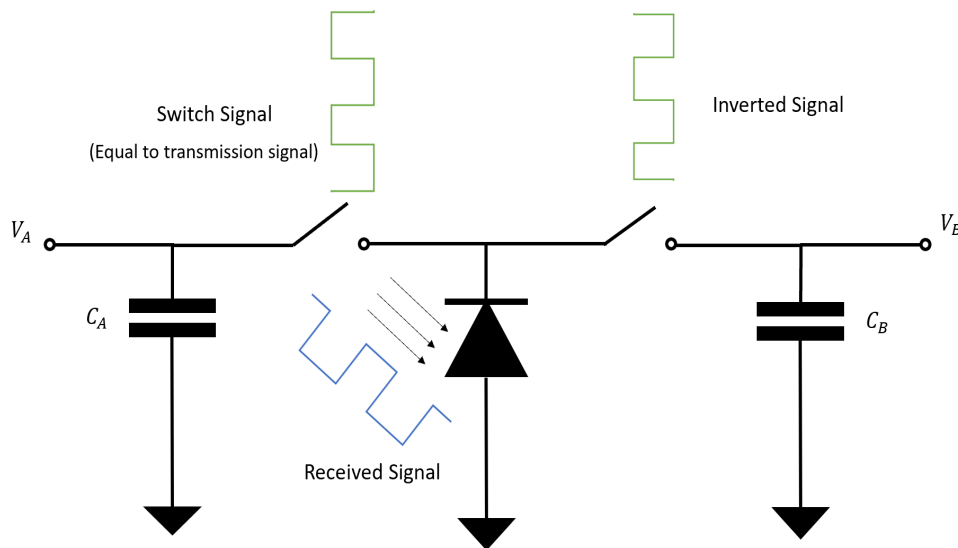
$$d = \frac{1}{2} * \frac{c\phi}{2\pi f}$$
(5.5.6)

Figure 5.5.4: Simple diagram of electrical circuit for the indirect time of flight sensor

As seen in Figure 5.5.3, two switches are modulated with opposite amplitudes. One switch is modulated with a signal that is at the same rate and timing as the transmission signal, and the other is modulated at the same frequency, but with a 180 degree phase difference, effectively flipping the amplitude. So, whenever one switch is open, the other switch is closed. One

switch is always closed, so charge is being depleted from one capacitor at a time whenever light is present on the photodiode. If light is shining on the system, then current will flow through one of the capacitors, depleting some of its charge. After one period, the total phase difference of the wave is proportional to the charges depleted in both of the capacitors, as seen in Equation *5.5.7*.
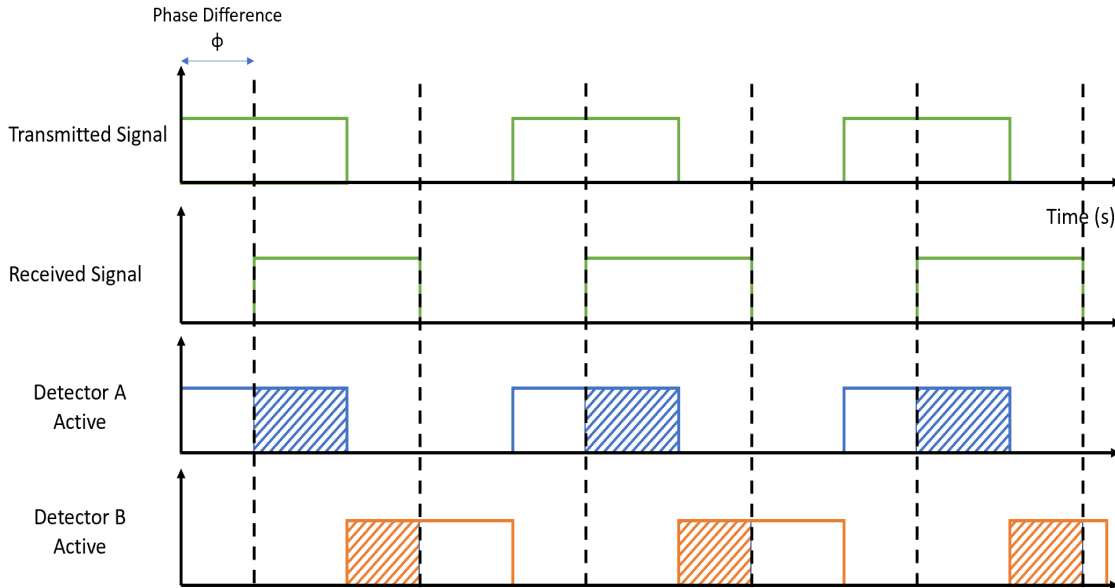

Figure 5.5.5: Plots showing how the received signal will always overlap and drain one of the capacitors

Figure *5.5.5* demonstrates that for a signal that is not perfectly in phase, the time that the incoming light overlaps with either capacitor is different. Therefore, the charge in both capacitors at the end of the measurement period is different. The difference in phase between the input and output signals is proportional to the charge (voltage) in the two capacitors and the reset voltage (V_{dark}) of the system. Therefore, the phase can be easily calculated from the voltage measurements of the two capacitors.

$$\phi = \frac{T_{delay}}{T_{ON}} = \frac{V_{dark} - V_B}{2V_{dark} - (V_A - V_B)} \qquad (5.5.7)$$

Combining Equation 5.5.6 and Equation 5.5.7, we can then accurately calculate the distance to a given object from the total phase shift of the reflected light. This expression, Equation ****.****.8, would then be used by the program to calculate the distance to the nearest object to the drone.

$$d = \frac{1}{2} * \frac{c}{2\pi f} * \frac{V_{dark} - V_B}{2V_{dark} - (V_A - V_B)} \qquad (5.5.8)$$

It should be noted that the distance equation is a ratio of the two voltages in the capacitors. The amplitude of the signal does not really matter in the final calculation. This means that the brightness of the reflected light is not a factor in the measurements. As long as the light reflecting is bright enough to trigger the detector, the phase measurement is accurate. This was a notable flaw with the more simple object detection system. The indirect time of flight sensor will give the same distance reading no matter what the absorption of the object is. This makes the system far more versatile than before. This versatility is why the team wishes to study this technology so extensively. Accurate and consistent distance measurements would be very helpful for this detection system.

5.5.3.3 Phase Overlap

The transmitted signals being used are periodic, meaning they repeat in a regular pattern. One may notice that after a certain distance away, the phase shift of the signal becomes 360 degrees (or 2π radians) and the signal completely overlaps. This phenomenon is called phase wrapping since the phase has completed a full period shift and has 'wrapped around' to the next period. This is the main limitation of the indirect time of flight system for distance detection. After this point is reached, the actual distance away the object is from the sensor becomes unclear, since the same phase shift This maximum range of clear measurements is called the unambiguous range.

The unambiguous range is based on the frequency of the modulation signal. This maximum range is shown in Equation 5.5.9. As one can see, the higher the frequency, the shorter this range becomes. Although, a higher frequency also means that the axial resolution of the image is higher. These two properties must be well balanced when designing a system.
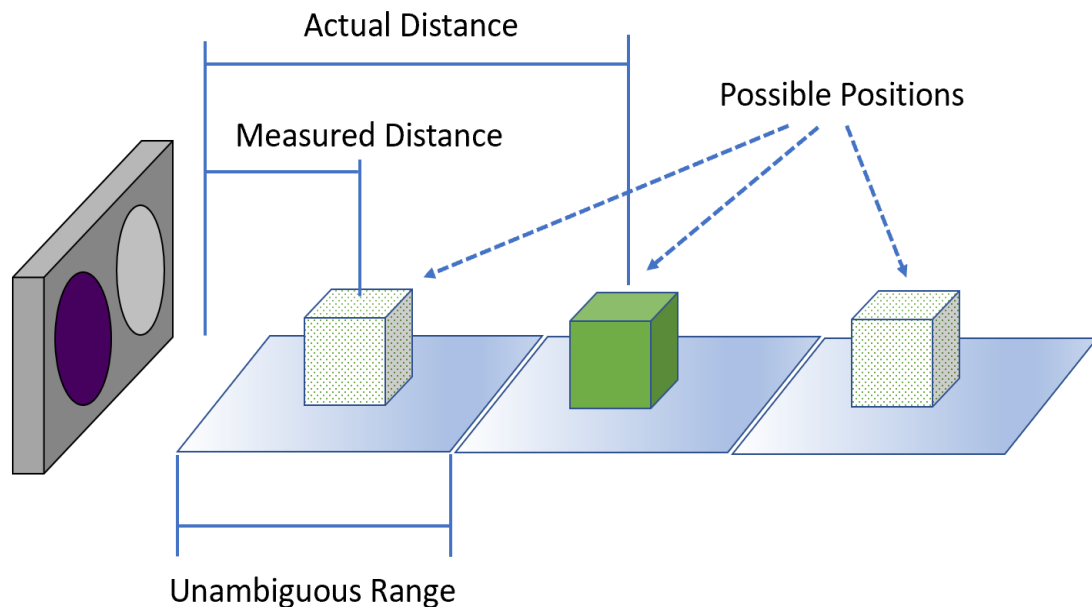
Figure 5.5.6: Illustration of consequences of unambiguous range from phase wrapping

$$Unambiguous\ range\ =\ d_{max}\ =\ \frac{c}{2f} \hspace{2cm} (5.5.9)$$

As an example, sensors like the MLX75027 can support a frequency range of up to 100 MHZ, which gives an operating range of about 1.5 m. The modulation frequency also affects the minimum operable range of the sensor. In general, a higher frequency will allow for a higher resolution, but a closer operating distance. This unambiguous range must be accounted for, and labeled accordingly during analysis. The code generated should make it clear when a measurement is out of range, and ensure that no unambiguous measurements are used by the drone's navigation system.

5.5.3.4 Multiple Phase Approach

One method to accommodate for the unambiguous range of the system is to use several modulation frequencies in tandem for the transmission signal. This maximum measurable distance is then found by the greatest common denominator among the several frequencies (Whyte, 2020). This greatest common denominator is now the effective frequency of the system. As demonstrated earlier, the smaller frequency will result in a longer effective range, giving the system a more complete scanning range.

5.5.3.5 Background Noise Correction

Even though the near infrared wavelength that was chosen is meant to be less abundant as background noise, there is still some in the environment, particularly outside where the primary source of background infrared radiation is the Sun. This would naturally result in some inaccuracy in the measurements. Stray infrared light from the sun would be detected by the system's sensors. This drains the capacitors prematurely and it will directly result in incorrect distance calculations.This would be part of the stretch goals for the group. Developing a system to account for any background noise from the environment is necessary for the most precise measurements.

5.5.4 Driving Circuitry

5.5.4.1 LED vs Laser Diodes
Whether the chosen optical system relies on direct time of flight or indirect time of flight, the system will require a very fast driving circuit for the transmissive optical element. This has been one aspect of concern for the team as they strive to create a proper time of flight sensor as an improvement to the simple object detection. These time of flight systems require very high modulation frequencies in order to function properly. In some cases, an LED may not be fast enough, and the team may need to switch to using a laser, which can be modulated far faster,

and there are a multitude of ways to complete this task that this section will examine.

The group had to determine whether to use a light emitting diode (LED) or a laser diode. There are certain benefits to each. LED's are definitely far cheaper, and can be easily replaced if broken. They can get reasonably bright, but their range is quite short compared to a typical laser.

Alternatively, laser diodes can be used. Laser diodes have the advantage of being able to create focused signals that travel farther distances. However, in some cases, the spot size may be so small after traveling so far, that the returning light may need separate optics to be refocused onto a given sensor. This may require additional testing. For now, the team has acquired a laser that has the ability to change its spot size.
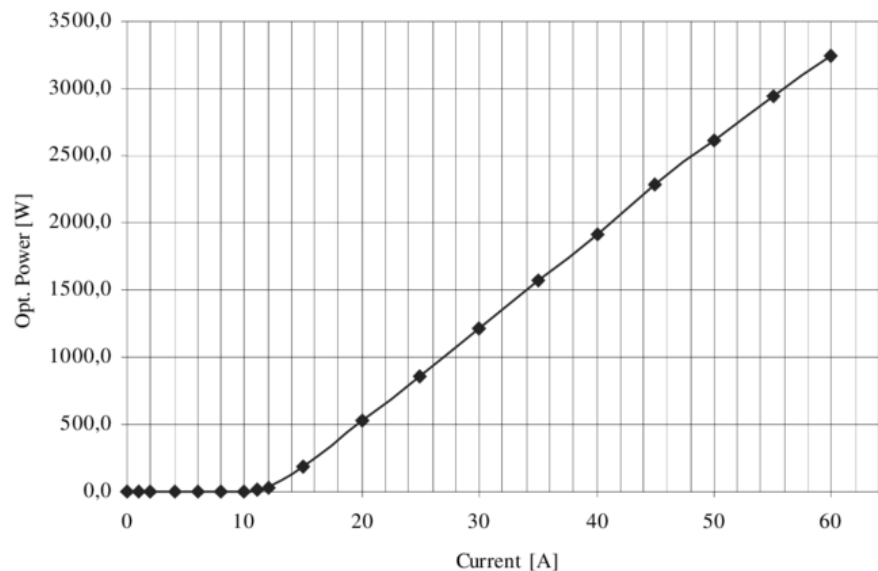
Figure 5.5.4.1: A Diode Laser Stack in terms of Current and Wattage

A laser diode functions similarly to an LED, but with a few key differences. In order to lase, the diode needs three things: a gain medium, a constant pumping mechanism, and a set of semi-transparent mirrors to cause oscillation of the light. A gain medium is necessary in order for a signal to be absorbed and emitted. In a laser diode, the cleaved facets of the double heterojunction structure act as mirrors due to Fresnel reflections. This reflectivity causes the diode to become a Fabry Perot cavity, and light can oscillate back and forth within the diode.
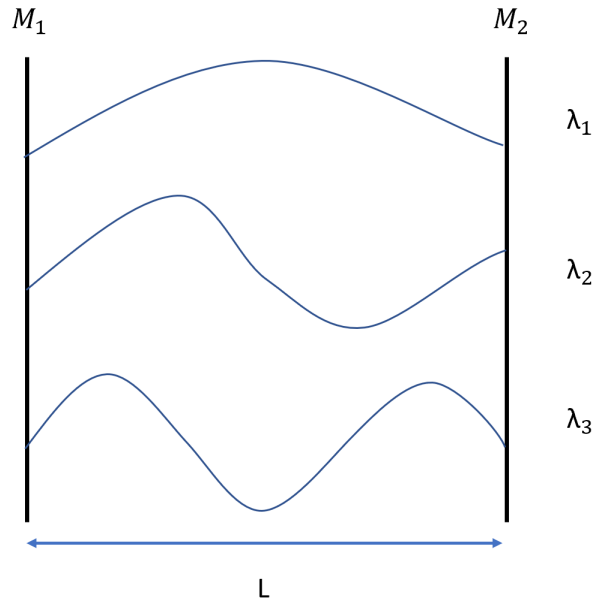
Figure 5.5.7: A Fabry Perot resonator

As seen in Figure 5.57, one of the primary features of this type of cavity is that only certain wavelengths of light can exist within it. This spacing, as defined in Equation *5.5.*10 is called the free spectral range, and it is defined by the speed of light (c), the refractive index of the medium (n), and the cavity length (L). With any Fabry Perot resonator, only the wavelengths that can exist in a standing wave can stay within the cavity. These wavelengths exist over a wide range, but the spacing between them is always periodic.

$$\lambda_{FSR} = \frac{c}{2nL} \qquad (5.5.10)$$

It is only these wavelengths that are able to lase in the cavity. The gain medium that is chosen has an explicit spectrum in which light can experience gain. Figure 5.5.8 shows this spectrum combined with a typical gain spectrum of a gain medium. When these two spectra are combined it will show the specific wavelengths that are able to lase. One of these, called the central wavelength, comes more abundant and overtakes the otters as it begins to stimulate other electrons, and only a single wavelength will last.
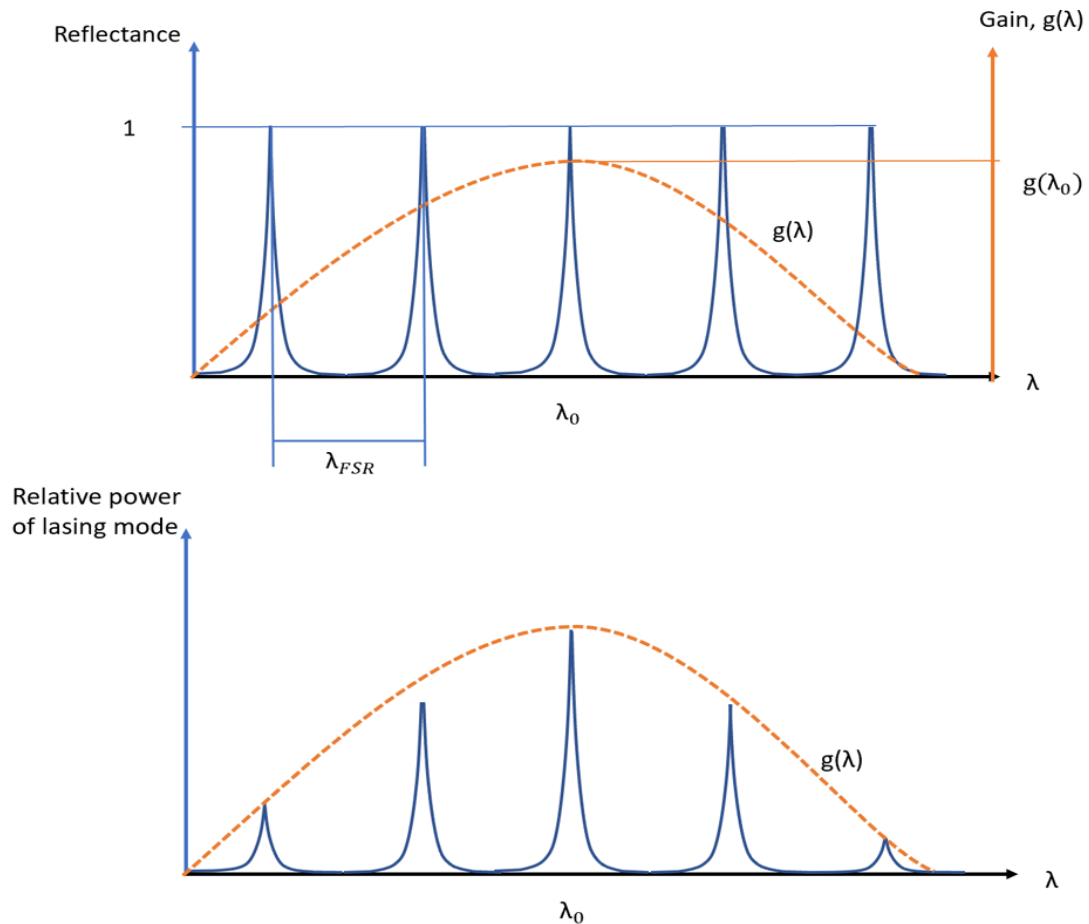
Figure 5.5.8: The gain spectrum overlapping with the Fabry Perot spectrum, and the relative power of each mode.

5.5.4.2 Laser Modulation

Figure 5.5.9 shows the simplified version of an I-P curve of a typical laser diode. As one can see, the diode will operate in what is called the "LED region." Here, there is some spontaneous emission, and the current in the material causes some photons to be released. However, after a certain threshold current, population inversion is reached, and stimulated emission occurs. It is this region, the "lasing region" where the laser diode should be operating. It is this area of the curve that is also the most efficient, and where the laser will experience its maximum power output possible for the diode. While the actual line in practice follows an exponential function, the figures in this report use a linear approximation to help facilitate understanding. These figures have been shown to be good estimates of the true behavior of a laser
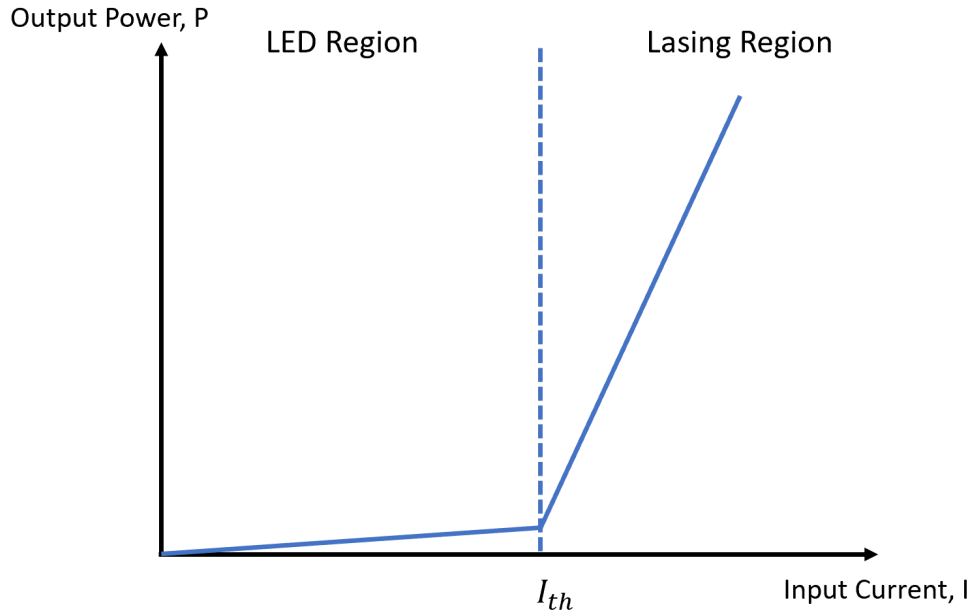
Figure 5.5.9: I-P Curve of a laser diode

In this lasing region, the metric known as the slope efficiency is quite high. The slope efficiency can be simply defined by Equation 5.5.11:

$$\eta \; = \; \frac{Change\ in\ Power}{Change\ in\ Current} \; = \; \frac{dP}{dI} \qquad\qquad (5.5.11)$$

This high slope efficiency means that small swings in the input current in the lasing region can create very large swings in the laser's output. This can be used to amplify small signals. For instance, if one were to send a relatively small amplitude electrical modulation current signal into the laser diode, the laser would amplify this into a higher amplitude signal. This amplification would allow the system to send out a higher power pulse, which has a greater chance of being reflected back in significant quantities to be measured by the detectors. This would be one of the primary benefits of using a laser diode instead of an LED in the object detection system.

5.5.4.3 Direct Modulation

Direct modulation of a laser diode is the process of creating a varying output signal by changing the input of the laser diode directly. There are several different methods to do this, and each has their own benefits and drawbacks. The three primary methods that are examined here are as follows: small signal direct modulation, large signal direct modulation, and pulse code direct modulation.

The primary metrics that each are judged by are influenced by their theoretical driving speed, the effects of relaxation oscillation, extinction ratio, and power consumption. These few factors will give the best outlook for a possible

modulation system. Balancing them out will ultimately give the best quality signal to use for the time of flight driving circuit.

The driving speed is a measure of the method's ability to switch on and off, and it can be influenced by several factors like the time electrons spend in the excited state in the gain medium before being stimulated and releasing a photon. This is largely inherent to a laser diode, so selecting a diode with a shorter electron lifetime would be the most ideal.

The relaxation oscillation is an effect that occurs when a laser is turned on. A buildup of charge carriers occurs just before a cavity can lase, and the gain medium is being pumped, and whenever the threshold is reached, there is a short period where the peak output power is quite high before the system relaxes and an equilibrium state of the system is reached. This time frame before a signal relaxes can drastically limit the driving speed of the circuit because the signal will not be able to repeat until it is finished.This relaxation oscillation is demonstrated in Figure 5.5.10.
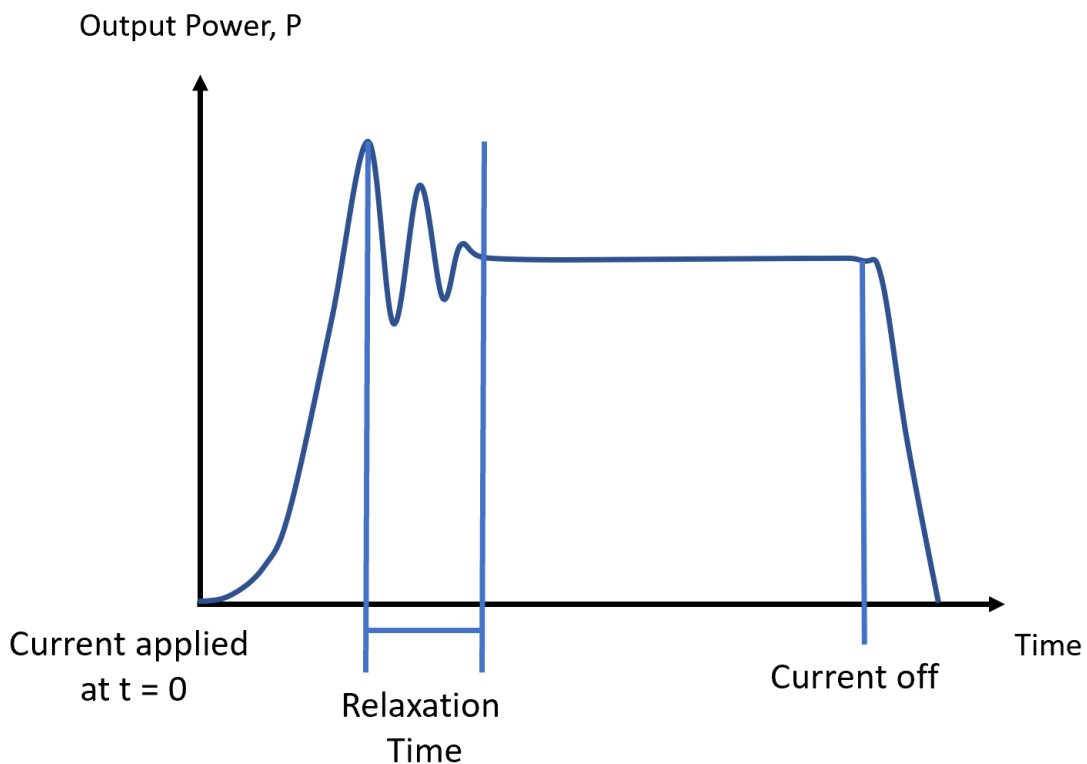

Figure 5.5.10: Diagram of relaxation oscillation of a laser

The extinction ratio is an important metric for the system because it is a measure of the ability to differentiate different parts of the signal.The higher this ratio is, then the more likely a detection system is able to tell the difference between the "highs" of the signal and the "lows" of the signal. To put it in another way, if the "on" state and the "off" state voltages are very close to one another,

then it is very difficult for the detector to differentiate between the two. If this extinction ratio is very large, and there is a big difference between the on and off states, then one can easily tell how the signal is structured.

The final measurement of interest is the total power consumption. Since the modulating current is not necessarily above zero for every configuration, the average power of each will differ slightly. In the interest of using as few power resources a s possible, the group wanted to measure this characteristic. The drone has self-contained batteries that the entire system will most likely draw from. So, the method that uses the least power is of interest.
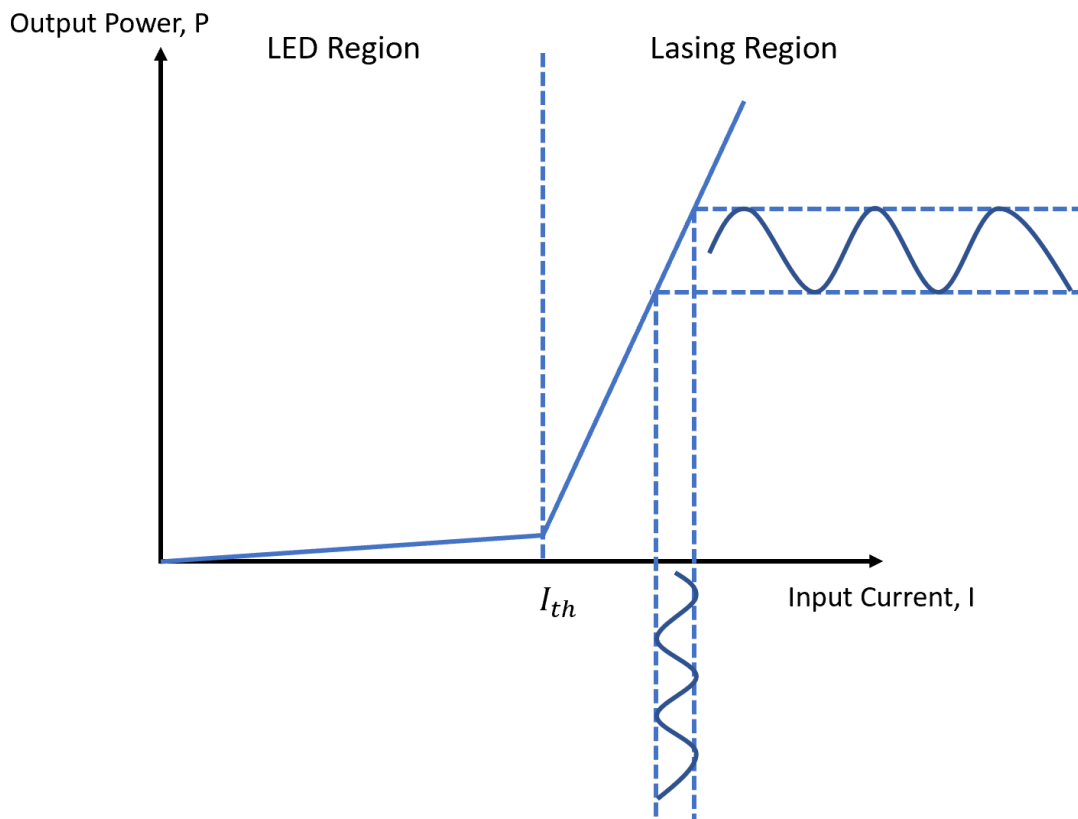

*Figure 5.5.11: Small signal direct modulation of a laser diod*e

With all of this in mind, the first method of direct modulation can be discussed. Small signal modulation is the method where the current modulates between two levels that are well above the threshold. The basic current and power behaviors of this method are shown in Figure 5.5.11 The difference between these two currents is not very large. Unfortunately, this means that the extinction ratio is rather poor, so it would be quite difficult to read the signal. Also, the power consumption is relatively large, since the average current is so high relative to the other methods.
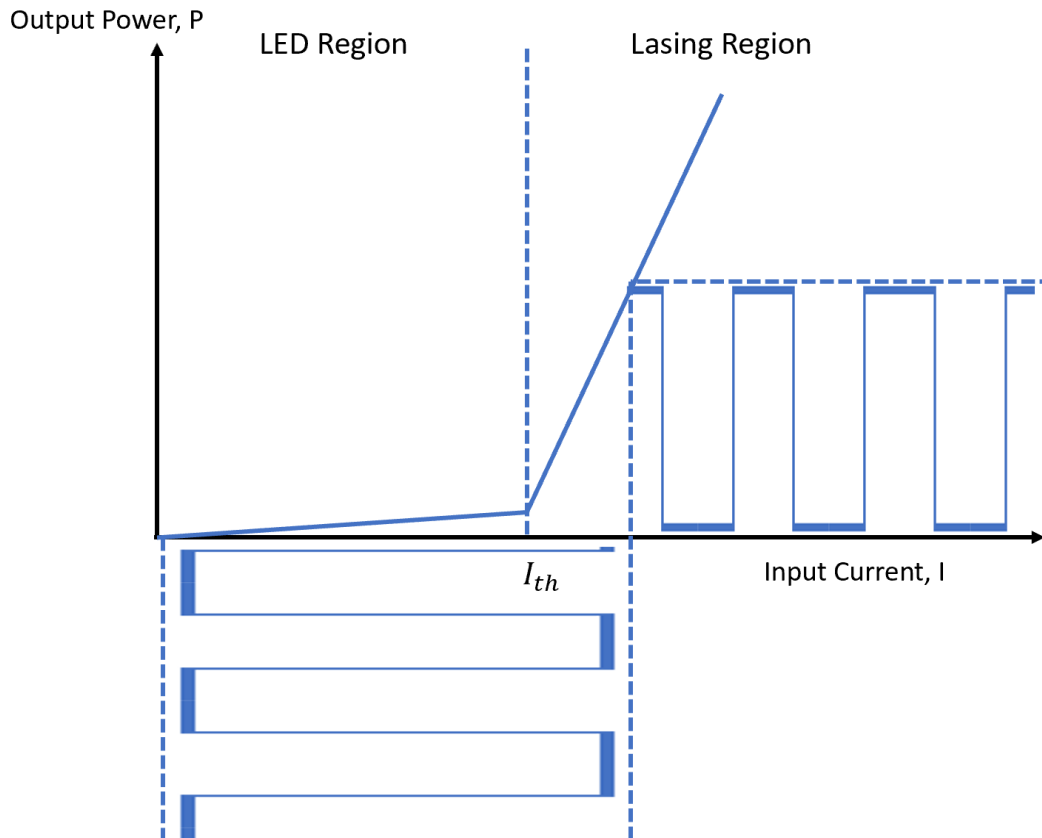
Figure 5.5.12: Large signal direct modulation of a laser diode

      The next method is large signal direct modulation. This method, as demonstrated in Figure 5.5.12, modulates the current of the laser diode from 0 to just above the threshold. This method results in a very high extinction ratio. The output power will modulate between 0 and another maximum. Naturally, this large difference makes any signal sent this way can be more easily decoded, with environmental noise affecting it less when compared to some of the other other methods. The average power consumption is relatively low since it switches between the lowest current of 0 and a maximum current that is only slightly higher than it needs to be for lasing. So, it will draw the least amount of power compared to the other methods. However, this method also has a significant problem when it comes to relaxation oscillations. The constant switching of the laser on and off means that the modulation rate is heavily determined by the relaxation oscillation of the diode.
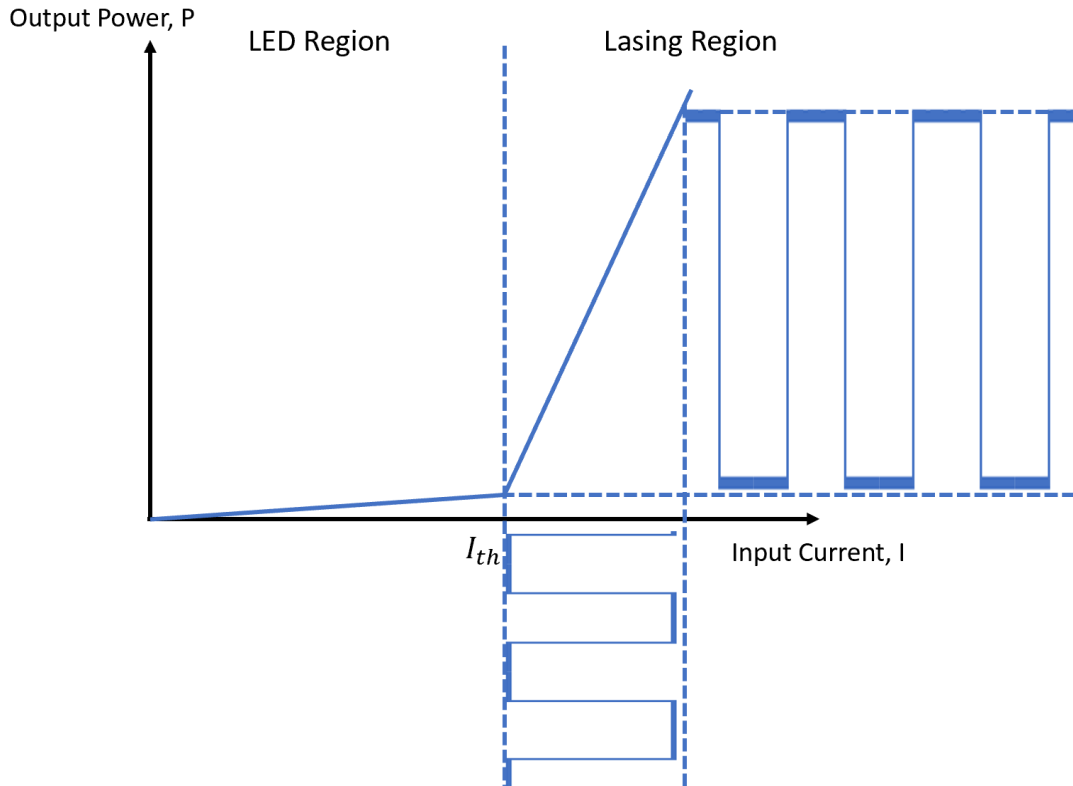
Figure 5.5.13: Pulse code direct modulation of a laser diode

Figure 5.5.13 demonstrates pulse code modulation. In this method, the bottom, or zero power level, is set at the threshold current, and it is modulated to a level significantly higher. The current is always within the lasing region, and it only grazes the edge of the LED region. This is a decent middle ground between the other methods mentioned so far. There is some slight concern with a relaxation oscillation, but not close to the level of large signal modulation. Since the laser is never technically off at any given moment.The power consumption is alright, with the average current level that is typically below that of the small signal method, but a bit higher than that of the large signal method. The extinction ratio from this modulation is decent, but not amazing. It is not as poor as small signal modulation, but certainly not as good as large signal modulation. With all of these factors in mind, this may be the best option for modulation of a laser diode in this particular system.

5.5.4 External Modulation

Another alternative method of modulating the transmission signal is through a process of external modulation. This method prevents the signal from. It also allows for higher driving currents. As shown in Figure 5.5.13, a continuous current is fed into the laser diode, which creates a constant output power. Then , this constant output power is then modulated separately in a device called an electro-optic modulator. There are several types of these devices that achieve this goal with different methods, but they all achieve the same goal. They take a

continuous light signal, and then alter it such that it has the same modulation as a particular input electrical signal.
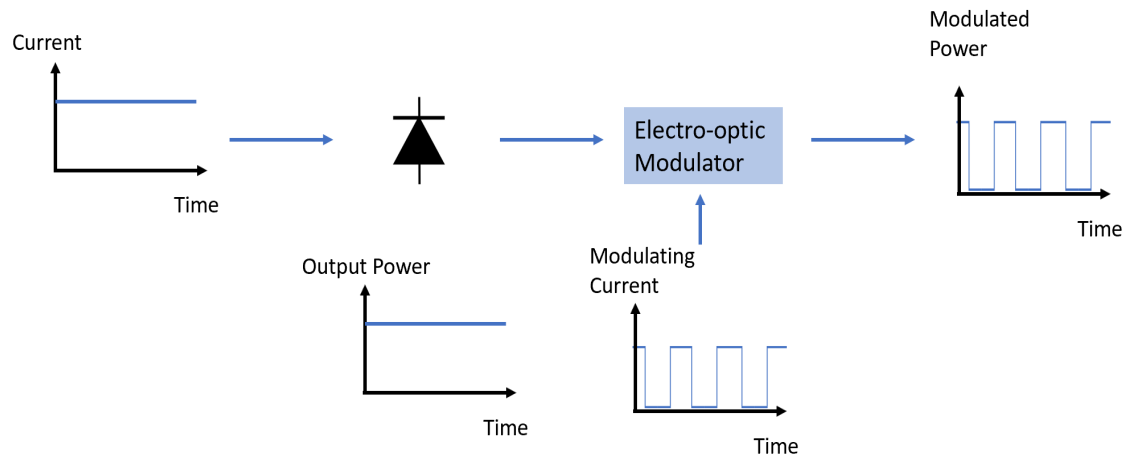

Figure 5.5.14: Simple external modulation setup

This type of modulation requires a type of interferometer, a Mach-Zehnder interferometer. The phase difference (ɸ) between the waves in both paths in a Mach-Zehnder interferometer with the path length (L) and laser wavelength (λ) is estimated by Equation 5.5.11

$$\phi \; = \; \frac{2\pi n L}{\lambda} \qquad\qquad (5.5.11)$$

In order to change the phase difference, one of the variables present on the right hand side of the equation must be manipulated. There are three possible characteristics of the system that can be varied: the length of the interferometer, the wavelength, and the refractive index. Naturally, the wavelength cannot be changed. It must remain a constant value throughout the system, or else the signal quality, outgoing power, and measured power may all be compromised. On the other hand, the path length of the system is not able to be changed in the interferometer setup directly without an entire system of moving parts. This is rather impractical, especially on the small scales that are required by the overall LiDAR system. Therefore, the variable that must be modulated is the refractive index. This is possible with an electro-optic modulator. These devices make use of piezo materials (not dissimilar to those that may be found in a microphone or speaker system) whose size and refractive index can be changed extremely rapidly. As an electrical signal is sent across these devices, the index will change, and then the input beam will experience a delay relative to a second beam that is passed through a second arm of the interferometer that is left unchanged. Using an electro-optic material whose refractive index can be changed with a voltage could allow for this setup to induce a phase shift. The resulting phase difference between the waves will cause an interference pattern. This patterned light will then become the outgoing signal that can be reflected back to the sensor and interpreted. This method can

be driven quite fast, and it prevents a phenomenon called *frequency chirp,* where the constant switching of a laser on and off causes a buildup of charge carriers in the laser diode. This causes sudden releases of energy that are significantly higher than those that are present throughout the continuous wave state of the laser. This will ultimately lead to some slight shifting in the wavelength of the laser. In some cases, this may not be too big of a problem, and the team took these effects into great consideration when deciding which method of laser modulation that they wanted to use.

After thorough consideration, the team decided that external modulation, while it is very efficient and can prevent frequency chirp, is not a good option for this. The optics to make such a setup too large and are quite sensitive. Additionally, after doing further research, the team decided that due to the often extreme prices for consumer electro-optic modulators, this external modulation method was not cost effective enough to justify using it in the final system

With all of this in mind a theoretical laser setup could have the layout shown in Figure 5.5.15. This laser rangefinder would use direct time of flight, and it would continually send out light pulses at a consistent rate. These would then reflect back toward This setup includes a lens in order to focus the incoming reflected laser light onto a single IR sensor. This would allow for fewer of these photodiodes required for the system.
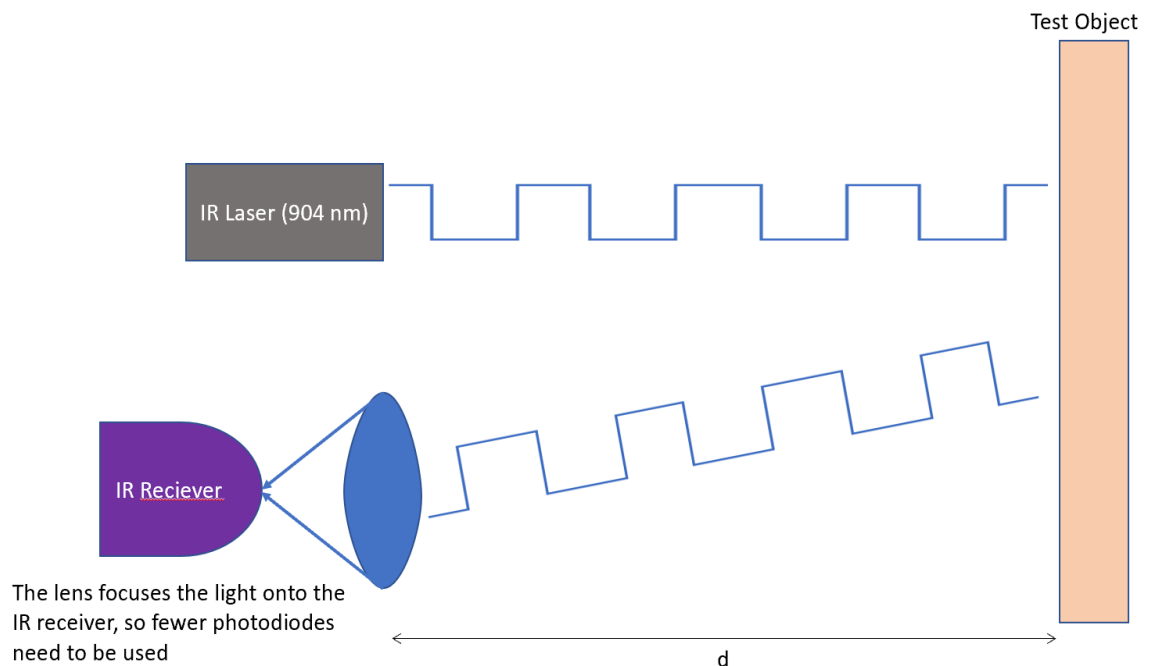

Figure 5.5.15: Potential setup for a laser rangefinder

Ideally, this laser ranging system would be implemented into a full 360 degree LiDAR scanner. One or two of these rangefinders would be mounted onto

a rotating stage, the distance would be calculated constantly as the sensors rotated. This would lead to the system creating a continuous circle of distances around the drone at all times. The circle of points with relative distance information is called the 'point cloud' of the LiDAR system. Such a setup is shown in Figure 5.5.16.
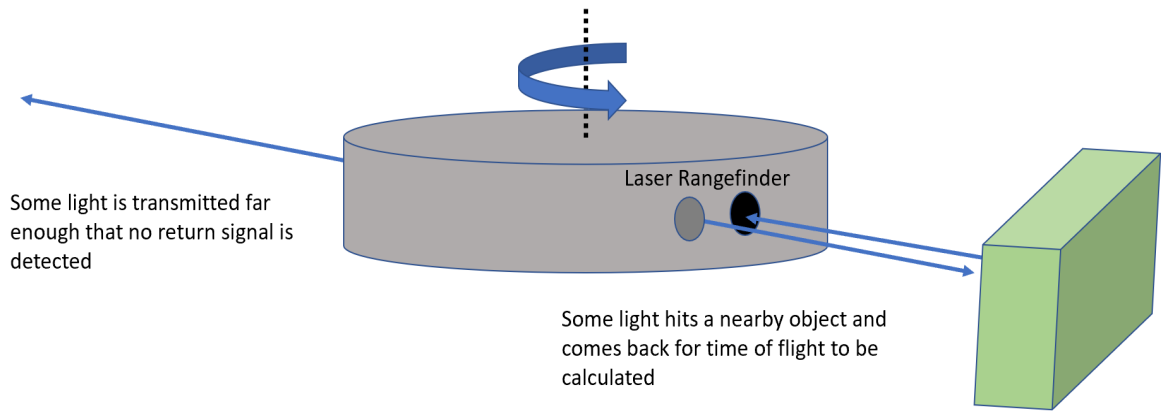

Some light is transmitted far enough that no return signal is detected

Laser Rangefinder

Some light hits a nearby object and comes back for time of flight to be calculated

Figure 5.5.16: Potential setup for a 360 degree LiDAR system

No return signal is detected

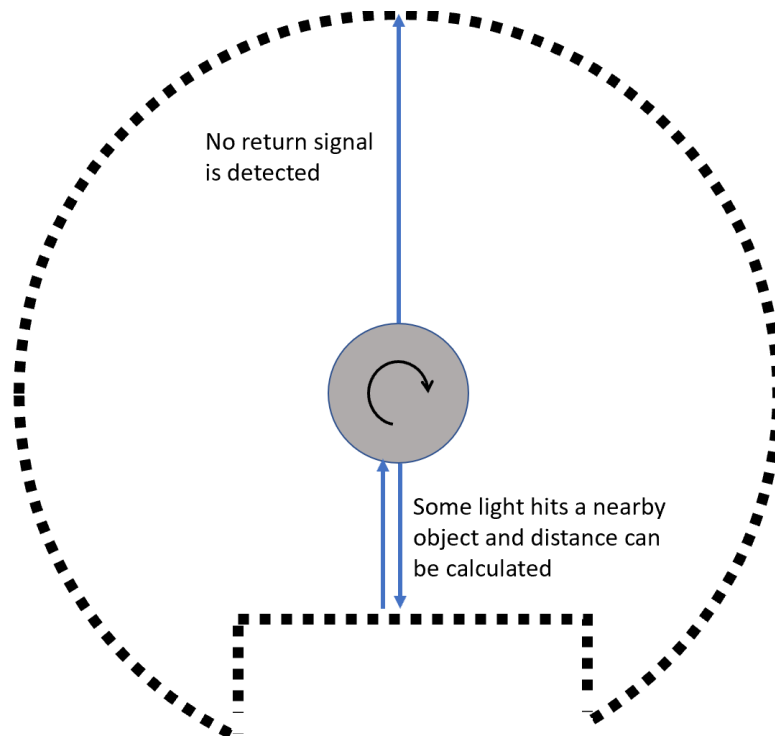Some light hits a nearby object and distance can be calculated

Figure 5.5.17: Nearby object creating a bump in the point cloud

Figure 5.5.17 shows that if there are no objects in the immediate area of the drone, then the values of the point cloud will automatically default to the

maximum measurable distance. So, if there are no objects at all, then a perfect circle with a radius of the maximum measurement distance is made around the drone. However, if an object is placed within a detectable range, then the outline of that object will appear as an interruption of the point cloud. This creates the final two dimensional map of the local area. Group members have speculated that thus information about the environment could do more than provide a warning for incoming objects. This could even be used to improve the image detection and navigation algorithm. If the system had a long enough range, and the software was able to correctly correlate objects in the point cloud with detected objects of interest in the camera, then the whole system could tell exactly how far away the target object is, enabling many other subsystems, like an estimated completion time. Additionally, if the drone had the ability to map an entire area, then it could feasibly be able to create a path that led to both the fastest completion time as well as the safest path, avoiding all obstacles with ease. All of these potential benefits allow the drone to make better decisions on how to navigate its environment. It is for these reasons that the group is heavily invested into the idea of making this LiDAR scanner a reality. It would provide numerous benefits that could dramatically improve the efficiency of the entire system.

This subsystem would likely need to be mounted on some kind of stabilization device. As a drone moves, it will naturally tilt in any given direction. This would naturally change the angle at which the distances are found. The largest issue with this is that the scanner would then be measuring off of the axis we want. This could even be so extreme that the scanner would completely miss some possibly hazardous objects next to the drone. To prevent this from causing significant errors, the system must be able to compensate for this. Perhaps a type of gyroscope system could be used, or maybe even a series of servos. Since the creation of a full LiDAR system seems very likely at this point in development, the team has chosen to put time aside for further investigation of solutions to these issues.

The primary goal of the combination of all of these systems is to create a two dimensional LiDAR scanner that is cheaper than the nearest commercial alternative. A similar system to the one shown in Figure **.16 will typically cost a consumer around $100 or more. Given what the team has been able to calculate for the electronic components and shell of the design (that would likely be 3D printed) they are confident that they can create a device that is well below that cost. If further tests are done to mature the rangefinding technology, the team hopes that they can accomplish this stretch goal by the time all of the systems must be integrated together

5.5.5 Final Circuitry

The final set of distance detection methods involved two primary systems

Signal Sending and Reception
Amplification
Schmitt Trigger
Comparison
Low Pass Filter/ Averaging

Received Signal
Photodiode
980 nm
Laser Diode
Modulation Signal

Ensure both waves have similar amplitude

Make both waves are square waves

Received Signal
XOR Gate
Sent Signal
Tdelay

Compare signals using XOR gate

Tdelay
Out

Output is a lower frequency signal that can be directly analyzed and averaged
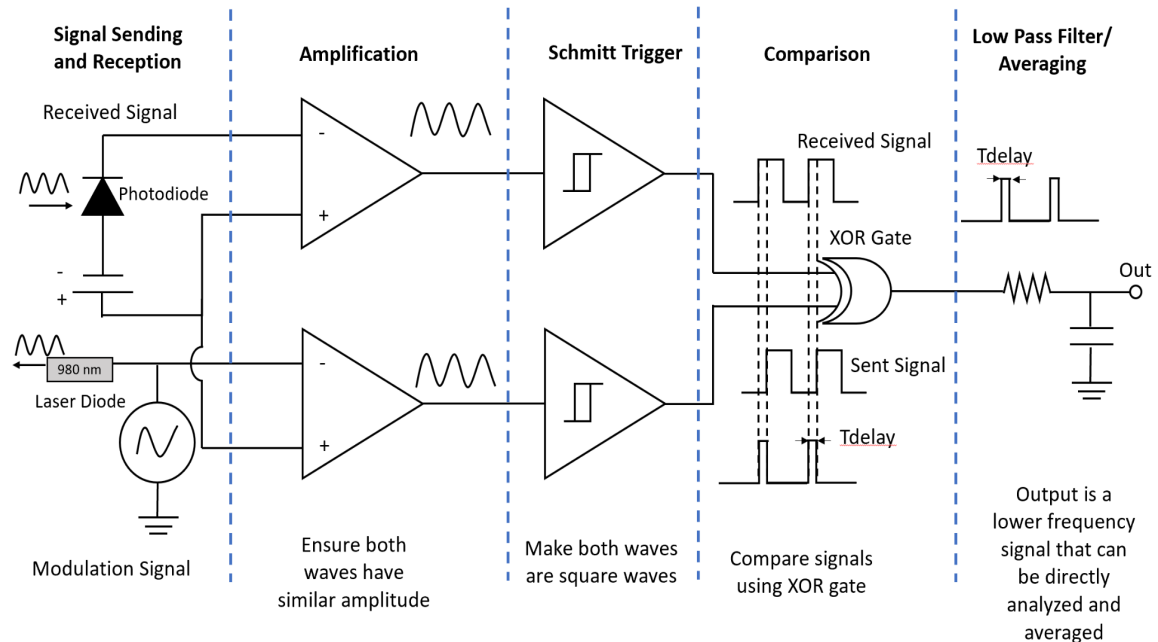
Figure 5.5.*18: Block diagram of driving circuitry for time of flight circuit*

      The time of flight circuit that was developed contains several primary elements. Since light travels very fast, taking a nanosecond to travel a foot, analyzing time of flight with a digital signal directly would require an incredibly fast processor. Instead, this circuit calculates the time of flight using analog signals. It starts by amplifying both the sent and received signals, ensuring they both have the same amplitude. Then, they are passed through a set of Schmitt Trigger circuits. This ensures that both of the waves are square so they can be more easily compared. Then, the two signals are compared using an exclusive OR gate. This will result in a series of pulses with the same period as the sent signal. The width of this signal is the time of flight. Since this cannot be analyzed with our microcontroller, this pulse train is then averaged using a low pass filter circuit . This gives a single DC voltage signal. The higher the DC value, the farther away the pulse. By inputting this signal into one of the Raspberry Pi Pico's analog to digital converters, we can calculate the distance to any given object that the laser and photodiode array is pointing towards.

5.5.6 Camera

      A full color camera is used as the eyes of the system. The drone will take visual data from this and use it to detect specific objects that the voice commands tell it to navigate around.The primary considerations for this device is the resolution and size. There are many reasonably sized and priced options for a color camera online that would work with the system. The resolution needs to be high enough to detect objects at a certain distance away, but it also must be low enough to be able to store and analyze the images efficiently. The onboard computer has a limit on processing power, and in order to navigate efficiently, it

needs to be able to process both the commands and the images. If the resolution of the camera is too high, then it may overload the system, causing the processing to slow down and increasing completion time. With all of this in mind, a standard high definition 720p or 1080p camera would be used, depending on the graphical processing capability of the chosen algorithm and the capabilities of the chosen microprocessor. We will be using 480p though, in hopes of saving on performance.
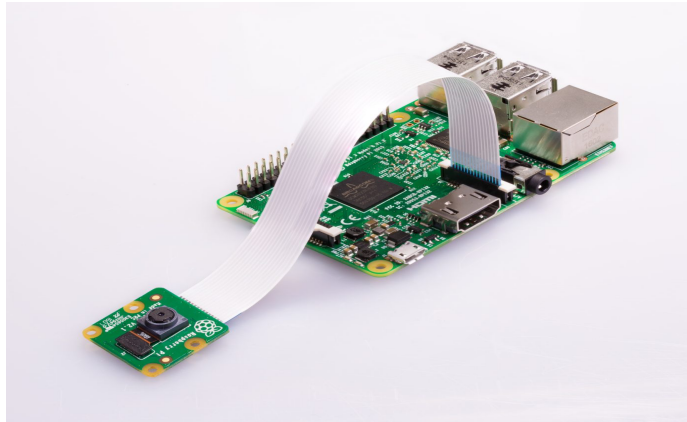

Figure 5.5.19: Photo of a Raspberry Pi Camera Module

The camera must also be compatible with the team's chosen processing unit. Often, small sensors can only be used easily with certain processors. There is a limited number of usable USB ports available on the unit, so it is advantageous to choose a unit that is able to connect directly with the microcontroller.

At least two of these cameras must be in use simultaneously, since the team needs the drone to have as wide of a field of view as possible. Only having a single camera may be limiting, and if it cannot see an object from one side, it may be required to rotate. This would call for extra subroutines and a greater number of areas of failure. To reduce this, the total field of view is increased with the two cameras. However, there are issues with placing too many cameras. Placing more than two cameras on the system could cause significant problems. After a certain point, it would negatively affect the weight of the system, possibly making the drone harder to fly. Also, the overlapping sections of the camera's view may be more difficult to splice together, leaving more room for bugs and errors to prevent successful operation. Moreover, more cameras means more data to process. One of the desired outcomes is to make the system reasonably fast once given instructions. The unit's computer has a limited amount of processing power to dedicate to this task, so it must be conserved in other areas where possible.

5.6 Decision Matrices

There are many factors that may influence what the group is able to do for the final system. The following charts are one of the ways that the group chose to help narrow down what to do in situations where the path toward a goal was not entirely clear. These decision matrices were created as a way to help the group make choices on key aspects of the system: what programming language to code in, what the obstacle avoidance system should be like, etc.. These charts proved to be an essential tool, especially early on in the development process where they were quite uncertain on how to tackle the more fundamental aspects of the final system.

In order to help focus their time and effort, the group created several decision matrices to help them decide on certain key aspects of the drone system. For each decision, they determined several aspects that are the most important to them including but not limited to: how difficult the option is, how familiar the team is with the given option, their excitement to do the option, and the cost of the option. Each of these specific characteristics were given an integer score between 1 (the worst) and 5 (the best). So, in the end, the option with the highest score would be the preferred one. It should be noted that even though an option may "win" in the chart by having the highest score that does not necessarily mean it is the chosen path. Several factors could come up in the design and construction process that the team did not foresee, causing them to change course on an idea.

Table 5.6.1: Programming Language Selection

1 = worst 5 = best	Difficulty	Familiarity with Language	Motivation / Excitement	Score	Comments
Python	4	5	5	14	The machine learning software primarily being used is 'PyTorch,' which is based in python
C++	2	3	3	8	Somewhat familiar, and has more compatibility/similarity with many circuit boards
Java	1	2	2	5	The team is not very familiar with this software, but wanted to keep their options open

The first chart, Table 5.6.1, weighs the different options for a primary programming language that the team could choose. The primary options were between Python, C++, and Java. After considerations, the team decided that they should primarily code in Python. This is the software they all feel the most comfortable using. Also, they are confident that any gaps in their knowledge could be easily filled by the wide variety of online tutorials and code repositories. Furthermore, Python is also a language that is often used quite often in the field of machine learning.

Python will also work with our determined flight controller, Pihawk. This is the flight controller that is being used by at home drone builders to implement machine learning algorithms. This method is integrated by a raspberryPi. These three components again, are widely being used for at home drone building which ensures that there is sufficient data for us to be able to perform our project efficiently and effectively.

Python is also a widely used and user friendly coding language. The four of us on the team do have experience with coding with python. Python is also used within our classes and industry. This gives us the opportunity to learn more about coding with this software and using this knowledge that we are gaining for after graduation. This knowledge is used for innovations within the areas of study we each decide to work in.

The wide usage also ensures that there is enough studies, open source codes, troubleshooting articles on the web in order to solve any issues we may face. It also ensures that other faculty members at the university will easily be able to provide feedback on how we can improve our machine learning algorithms within the software. Below is a reference figure on how to properly connect the pihawk flight controller with a raspberryPi for efficient usage with python coding. This is important because it shows the pins that we are to use.

The next chart, Table 5.6.2, describes the different options for an object detection and obstacle avoidance system. After comparing the options, it was still unclear what to do. The team eventually decided that a simple object detection system would be their core goal, and they would create an advanced goal to have a series of distance detection sensors.

Finally, their stretch goal would be to add a full 360 degree LiDAR scanner that would be able to create 1 2D map of the drone's environment. This system will also detect the designated safe landing zone for the drone. The landing zone is a white paper with a large black X. The drone is to avoid the balloon obstacles and then to land on this designated zone. Therefore, there must be two cameras for the detection system, one for viewing its horizontal surroundings and one for viewing the below surroundings to determine where the safe landing zone has been placed

Table 5.6.2: Drone Flight Object Detection System

Drone Flight Object Detection System						
1 = Worst 5 = Best	Cost	Difficulty	Familiarity with Technology	Motivation/ Excitement	Score	Comments
Simple Laser Distance Detection	4	4	4	1	13	Only detects how far the drone is from an object in front of it. No mapping capability, but would be cheap and easy to implement
360 Degree LiDAR Scanner	1	3	4	5	13	More expensive, but allows us to create a comprehensive map of the area the drone is moving around. Would lead to a higher success rate.
Simple Object Detection	5	2	5	1	13	Very cheap, and it can be easily integrated into multiple parts of the drone, but cannot map the environment. Limited overall
Camera Image Analysis	3	3	4	2	12	More difficult mapping. Also requires more AI training to be able to "see" when objects are getting larger and therefore closer. Would be harder to code for and needs more processing power

5.7 House of Quality Analysis

Below is the house of quality table that is used as needed for this project:

Customer Requirements	PRODUCT COST	INSTALL TIME	FIELD OF VIEW	DIMENSIONS	BREADBOARD
COST	-	-	↑	-	↑
EASY INSTALLATION	↑	↑↑	-	↓	-
VIDEO QUALITY	↑	-	↑	-	↑
ACCURACY	↓	↓	↑	×	↑↑

Analyzing the above diagram, there has been a general baseline of requirements established for the sensor system. Due to the complexity of the design of the drone, an easy installation and minimized time to install is necessary. These benefits may come with an increased cost as defined by the table, but having some funding provided by the sponsor, cost isn't as important as long as it's within reasonable range. Also, with increased cost also comes an optimized video quality and accuracy of detecting other objects.

In terms of the dimensions of the system, that would be the least important aspect. It has minimal impact on the other requirements other than potentially a bigger system providing better functionality. Finally, the FOV is also very important as it can increase the width of the detection zone, which may in turn

make it easier to detect objects. A wider range shall result in a higher accuracy for the drone to more effectively make decisions.

Overall, it can be concluded that the primary requirement that needs to be satisfied is accuracy, since without proper object detection, the drone will not be able to successfully complete the test. Therefore, according to the above diagram, the following aspects shall be met in an order of significance when choosing the best sensing system as described in Table 8.1.

Table 5.7.1: Describes the different engineering requirements of the system and their relative importance

Primary	Field of View - Wider range will produce better distinguishment
Secondary	Cost - Lower cost will provide more flexibility for other components
Tertiary	Install Time - More complex systems shall require longer installation time
Least Important	Dimensions - More complex systems does not necessarily mean larger dimensions

5.8 Breadboard Testing

The team needed to test some of the components on a breadboard as a proof of concept. This section contains an image of the object detection system made for the Senior Design discussion's midterm demonstration. This system is able to detect whether an object is within a range of about 30 cm, and it can send a signal to a hypothetical microcontroller to act as a warning system to stop the drone from flying in a specific direction.

It contains a potentiometer which is able to fine tune the exact distance an object must be away to be detected. An improved version of this system was created by the photonics engineer for their final demo in Senior Design 1. They still believed that there were some final improvements to a distance detection system. In fact, a more mature system was demonstrated in the final demonstration. A more advanced system would likely be printed onto a PCB for the final system. This would miniaturize it, making it easier to implement into the final product.
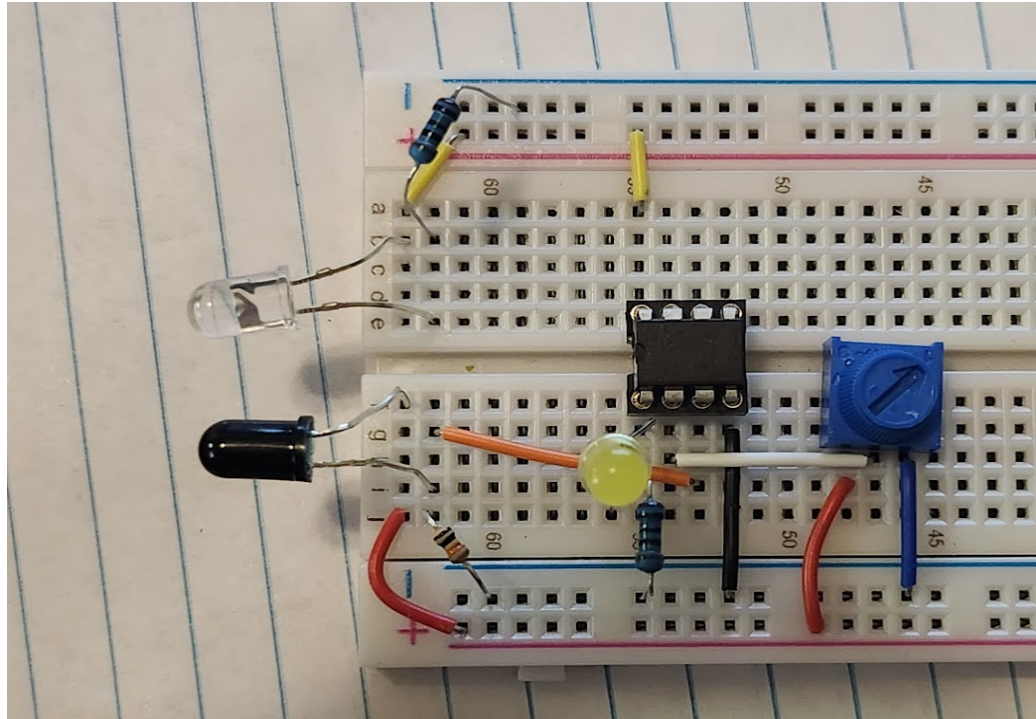
Figure 5.8.1: A breadboard model of an object detection system consisting of a 940 nm LED (top), an LM358 op amp, and a NIR photodiode (bottom)

5.9 PCB Planning

One of the requirements for the senior design project is for the team to create a custom printed circuit board (PCB). This design should be original and unique, and it must make significant contributions to the functionality of the device. This section describes the basic requirements of the PCB, and how the group will approach them in addition to making a decision on what part of the drone's system is put onto such a component.

The PCB has a microcontroller unit chip on board. For data intensive jobs. For some tasks that require heavy analysis and computation, the professors of the class recommend that two different microcontroller units are used. The first is used for the primary tasks, like power controlling and basic movements. The second processing unit would then be used solely for the more computationally heavy elements, like analyzing a point cloud. This type of system prevents resources from being diverted away from other core tasks, like power control and movement instructions.In the case of this project, these errors could look like a delay in processing time, or lag. When it comes to an obstacle avoidance system, a delay could mean the difference between colliding with an object and crashing,

According to the senior design requirements, the PCB must be integrated, meaning that all of the components (like the sensors or output controls) that the

microcontroller controls must use power from the same supply. Also, the input and outputs of the controller should be clearly defined, partially to make it easier to explain the PCBs purpose to judges in the final demonstration.

The professors defined that part of a significant PCB design in the context of this class is to have an 'amateur approach.' This means that the group will not be creating the PCB entirely from scratch. They are using existing software available online and hardware available from the major PCB manufacturers to create their design. Anything done from scratch requires prior approval from the professors before it is made. Given the group's rather limited experience with PCBs, and through the professors' recommendations they decided that trying to make any of these parts from scratch would be far too complex and not worth the effort.

The group has made many considerations as to which aspect of the system would be good for the PCB. There were talks of making the image processing and drone control onto a PCB, but due to the very heavy computational requirements, the group explored other options such as buying a graphics processing unit. While it is still an option for next semester, the group agrees that further deliberation is required before the final assessment on the matter is made.

The other component that could be put onto a PCB is the optical detection system. As it stands now, the team appears to be on track to being able to create the full 360 degree LiDAR scanner. This would require a significant amount of power control across the multiple components. A battery power source must be split into the laser diode, the detection system, the rotation mechanism among, etc.. This requires a fair amount of computation, and combined with the other software computation needed to construct a point cloud of the gathered data, some type of processing unit is required for this system to function.

Furthermore, a PCB would be able to have connections that could connect more directly with the RaspberryPi than something like an Arduino. Making a PCB for these components could also enable the group to use solid state electronics, which have the potential for much higher driving frequencies, an essential component of the rangefinding system. One of the comments made to the group during both the midterm and final senior design demonstrations was that the circuits would likely perform much better with solid state electronics on a PCB. Furthermore, the system has very straightforward goals, with clearly defined input and output requirements. Not only does this help them more easily explain the significance of the PCB to the judges during the final presentation, but it also simplifies the circuit itself. Too many inputs and outputs or moving parts could make the design messy and complicated very quickly. The group wants to avoid this confusion wherever possible. None of the group members have had experience designing or constructing a PCB design prior to this class. With all of this in mind, the group is confident that the LiDAR object detection system would be the perfect candidate for a PCB assembly.

The group had the choice of ordering a PCB from multiple companies. Most of these companies offer the service of assembling the PCB with the components specified for an extra charge. The group talked over the benefits and downsides of each. Pre-assembly would free up more time for the group to divert toward other tasks; however, the group also wished to minimize the overall costs. After all, especially with the object detection system, one of the primary goals is to create something that has a competitive price when compared to all of its commercially available alternatives.

Soldering the components to the PCBs themselves would save the group money; however, soldering PCB components can prove to be quite a challenging and tedious task. The parts that make up a PCB are far smaller than what are found in most DIY circuits, and it requires a high level of precision and skill. Not only that, but more fine soldering tips are needed to effectively put the parts onto the board.

With both methods of PCB assembly in mind, the group came to a decision. They have a member of the team that is quite proficient with soldering components. This member has had a lot of experience soldering different electronic components together, and after researching different methods on how to do so with PCB components, they are confident they have the ability to assemble it themselves.

While there are some aspects related to this that are subject to change in the next semester depending on time constraints for the final drone construction, the group is confident that they possess the capabilities to assemble the PCB themselves rather than pay a premium to have the manufacturer do it instead. Not only that, but they are confident with their tentative decision to use the optical detection subsystem as the chosen electronic component to base their PCB assembly off of.
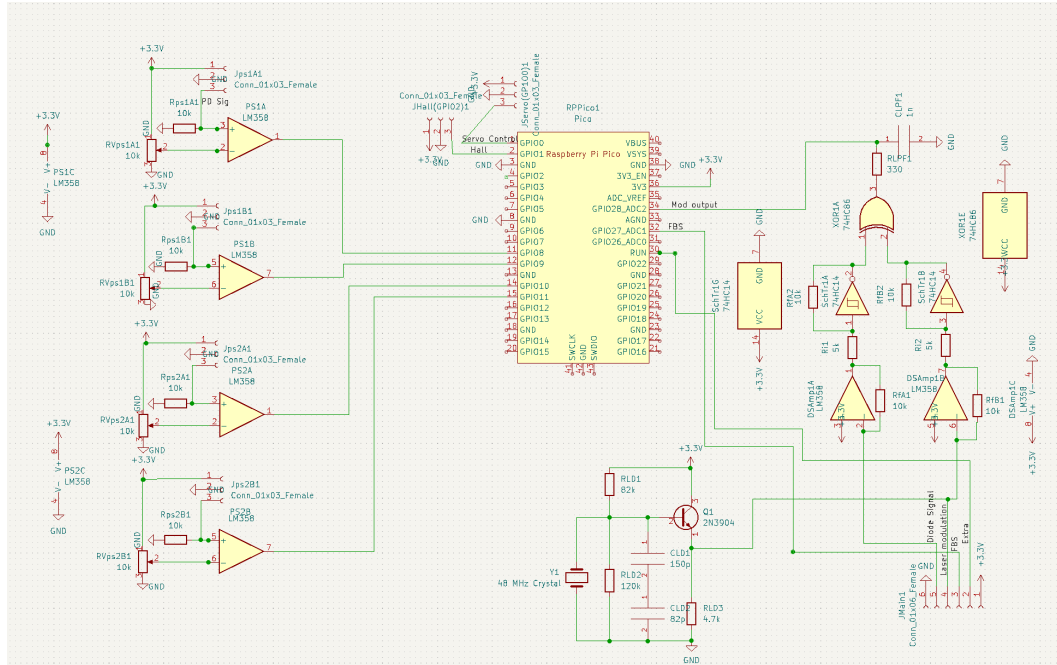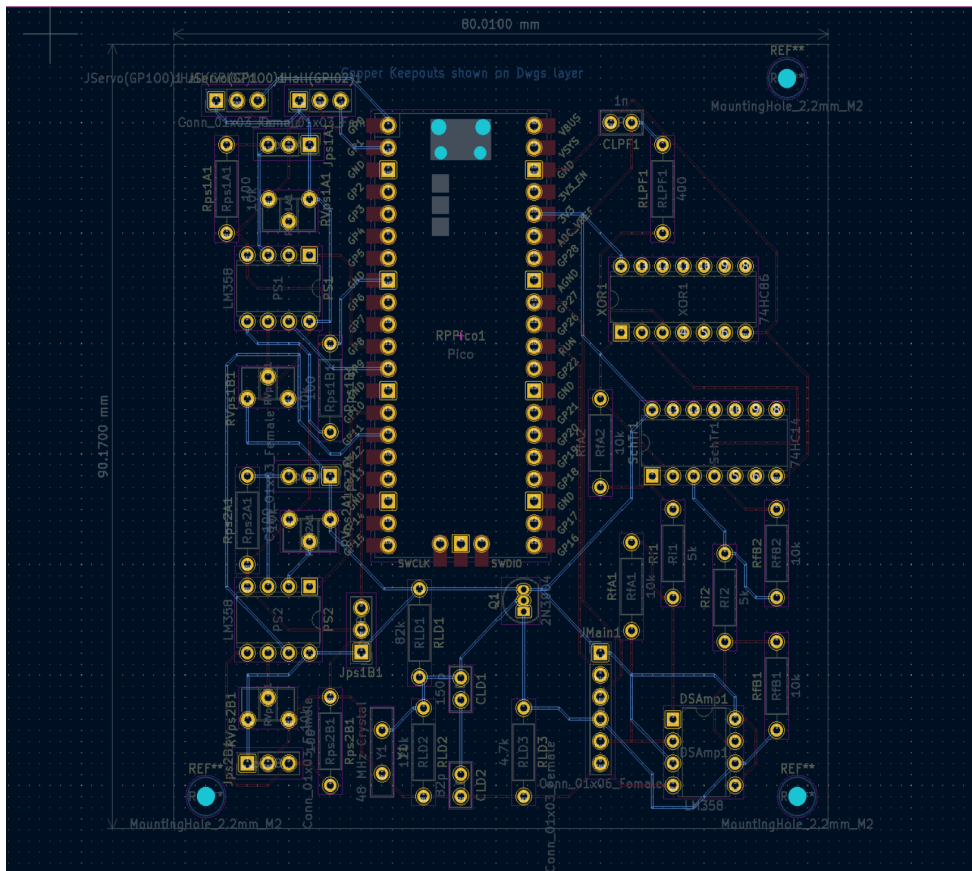
Figure 5.8.3: Final PCB Circuit

Figure 5.8.3: Final PCB Design

The final PCB design, as shown above in Figure 5.8.2, consists of three main sections: proximity sensing circuits, the modulator, and time of flight detection. The left side of the board contains several amplifier circuits that will receive the signals from the four proximity sensors. Whenever they detect a nearby object, a high signal is sent to the processor. Next, on the bottom of the drone, there is a Colpitts oscillator circuit. This uses a 48 MHz crystal oscillator to generate a continual oscillation signal that is sent both to the 980 nm infrared laser and the time of flight circuit. Finally, the right side of the board holds the time of flight circuit, where the received and sent signal can be directly compared. In this project, it is an RP 2040 Microcontroller unit that is on a Raspberry Pi Pico. This is the main unit that interprets all of the input signals and gives instructions for the behavior of other components like the servos and LED signals It also provides the output signal for the object detection system on the drone. Additionally, there are three separate mounting holes for the board to screw into the LiDAR assembly. The final PCB will be put within the final assembly since its primary functions involve controlling the different aspects.

Table 5.9.1: A basic comparison between different PCB manufacturers

Manufacturer	Price	Consumer Perception (according to Google Reviews)	Comments
PCBWay	Low	High	May have longer lead times and higher shipping rates. Centered in China
Camptech	Average	Great	Somewhat delayed shipping. Based in Canada
Sunstone Circuits	Moderate	Average	Centered in US, shorter lead times and lower shipping costs

5.10 Reference Design Material

The following section describes the basics of how the drone and its components shall be built, as well as visual representations of the functionality of the device once completed. These components include the propellers, the UAV system, and the PiHawk and PiCam systems. Each component is very important for object detection and the below references will help get an understanding of how they coexist with each other to create an efficient detection system.

5.10.1 UAV System Components

The UAV system is crucial to enabling autopilot in which software will control the flight of the drone, rather than an independent remote control. This system is comprised of the following procedure:

- Bottom Mounting Board
- Autopilot Feature
- Motors
- Top Mounting Board
- Propellers

This system is built in a layered method to ensure proper security when installed on the drone. The process of constructing this is simple, but there are some precautions to take, such as ensuring the propellers are all aligned in the same orientation to prevent misdirection during flight.
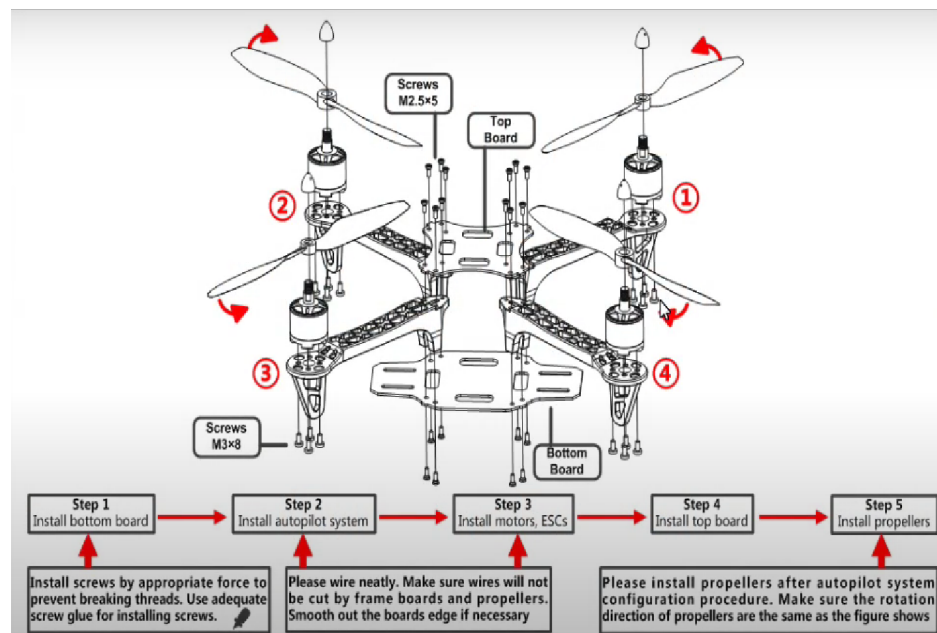

Figure 5.10.1:   Step by Step diagram showing the basic steps of how to assemble a UAV.

The above figure is from the drone building kit that the team is looking at purchasing. It is important to our team because it shows how the drone's main frame will look and the steps in building the drone. All team members are novice in building robots and building the drone will take a great deal of care with precision, and wiring. All electrical components of the drone must be calibrated prior to turning on the drone. Electrical components that do not have calibration can result in an electrical fire, which is to be avoided for safety measures and the component reliability.

It is impeccable that we do our best and uphold the highest safety measures when building the drone, since time and resources are now limited. Burning an electrical component can and replacing and severely impact our deadline with current wait times on parts delivery. For this reason, the team is looking to purchase a drone kit with instructional videos and spare parts in order to ensure that we are able to meet the senior design project deadlines within the spring semester.

Below provides images of components that are within the drone kit our team will purchase over the winter break. Solid work drawings of the drone frame were attempted by Jazmine but found to be incomplete by project submission due to novice use and continuous crashing of the university provided software. It is imperative to note that our drone is a standard quadcopter drone and the innovation and focal of this project is the vision navigation and machine learning components of the drone. Not the actual drone design itself.

5.10.2 PiHawk and PiCam Components

The PiHawk Flight Controller is crucial for sending data to the camera, which is connected to the Raspberry Pi motherboard. This allows adjustments to the field of view, the frame angle, and also liftoff thresholds. This camera provides high efficiency and is currently known as the best flight controller to be implemented for a civilian used vision navigation drone.

The PiHawk will allow user interfacing in which the region of interest can be chosen via the camera mapping provided by the PiCam. As described by the figure below, there are in fact multiple different function calls which are used to set values for altitude, takeoff markers, radian to degree conversion and vice versa, as well as simple device connection.
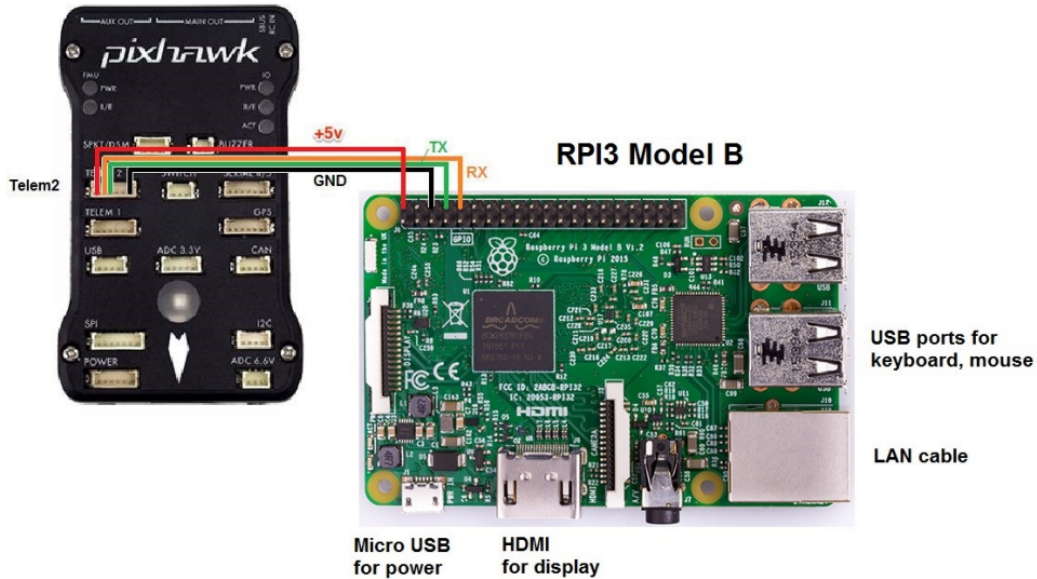
The above figure describes how to connect the RaspberryPi to the Pihawk controller which is essential in ensuring that the PiCamera module will integrate properly with the controller of the system. This figure is essential as a reference for the team and for the judges to be able to provide feedback on our specifications.
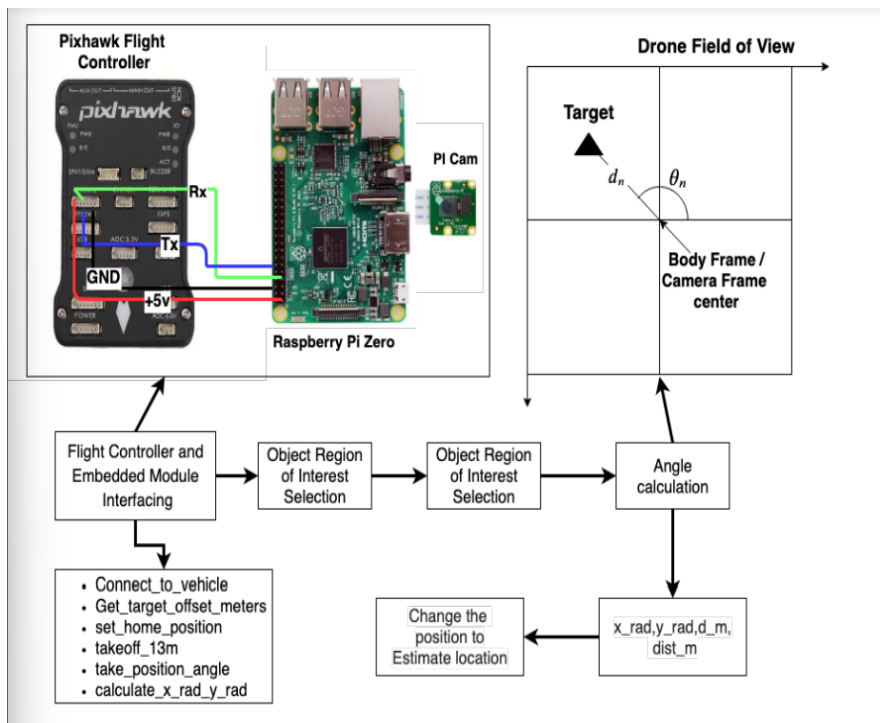

Figure 5.10.2: Diagram showing the Pihawk flight controller and Picam module positioning and drone field view. Our design will include two cameras for complete view

Figure **5.**10.3 describes the procedure for mounting the PiModule camera onto the Raspberry Pi motherboard. While there are some optional components, only the primary parts such as the circuit board, dust cap, and adjustment rings are generally needed for this type of project. The dust cap is significantly important to prevent any blurred camera results, as well as proper layering since any error can cause the camera to not perform as expected.
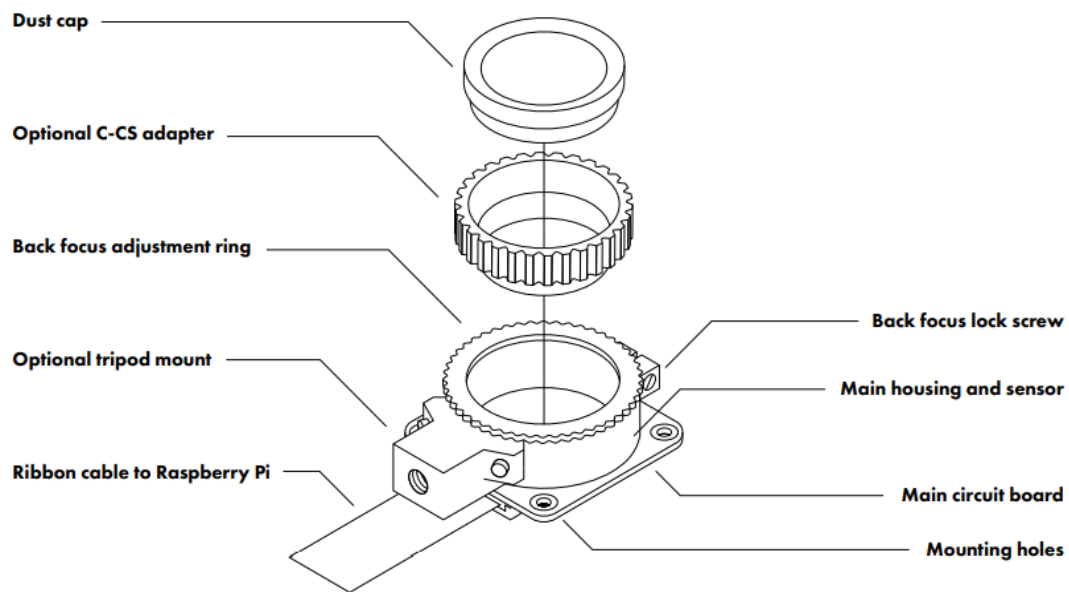

Figure 5.10.3 Mounting the Pimodule camera onto Raspberry Pi 4B

5.10.4 Reference Frames

Reference frames are also important in order to develop an accurate 3D environment for object detection. The ability of the drone to orient itself and other objects within a given environment is an essential skill that the system has in order to function properly. Figure 5.10.4 shows these references in perspective to the x, y, and z planes.
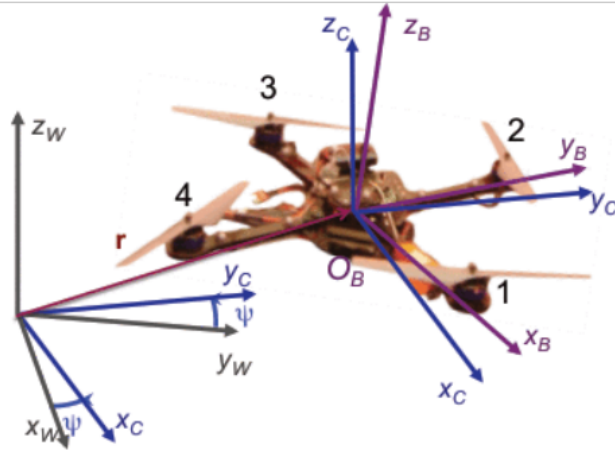
*Figure 5.*10.*4 The flat outputs and the reference frames needed to take in consideration for the software and the drone in a 3D environment.*

5.10.5 Camera Perception References

The camera of the drone has the capabilities to detect geometric objects, such as 90-degree angles, to determine a doorway, a window, or even a square object. With some fine tuning to the software, this feature becomes a very efficient method overall.

By detecting openings and obstacles using this perception technology, the drone can develop a motion primitive, in which it will project a path given the objects detected in order to prevent crashing into walls or obstacles or potentially flying in an unwanted direction.
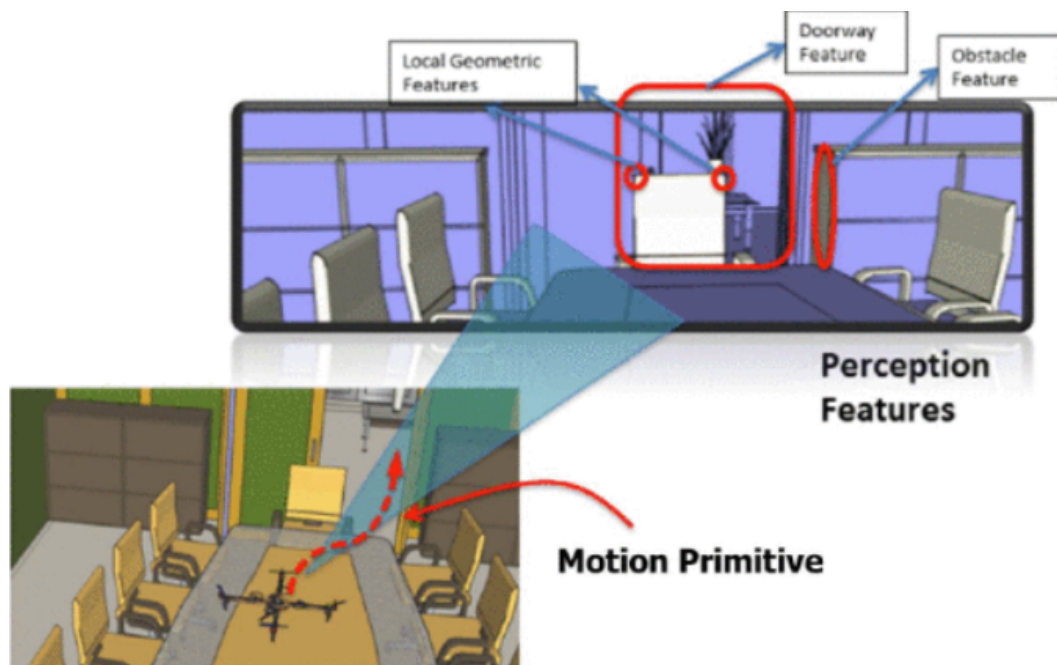
*Figure 5.*10.*5 A representation of a drone in motion in an indoor environment and detecting objects in its view.*

5.10.6 Navigation References

Beyond the camera features, there are three other features that are crucial to building a successful drone that has visual navigation abilities. These include awareness, basic navigation, and expanded navigation.

With awareness, the drone needs to have a reference to the edges of the testing environment, as without any insight on that, crashes are much more likely to occur. Adding to the simple object detection for the shapes hung from the top of the mesh cage for testing, the drone shall be able to detect the walls using the same mechanisms.

Basic and expanded navigation are simple to integrate, and for this type of project, drastic altitude changes aren't necessary. For expanded navigation such as "flips", that kind of movement would be rather detrimental for the drone especially in terms of detecting objects.
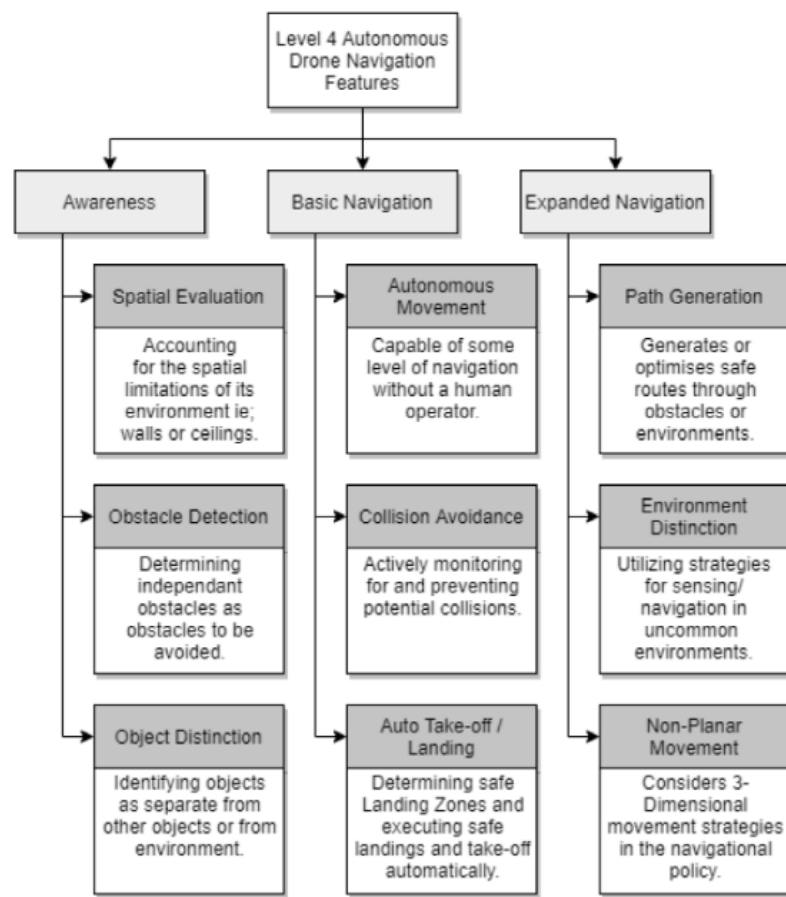

Figure 5.10.6 A flow chart showing the three main features the team is trying to accomplish with our drone.

5.10.7 Drone Tilting Phenomenon

In all types of flight, there are many natural occurrences that can easily cause the vehicle to tilt in multiple directions. Steady wind speeds, random wind gusts, or any number of other forms of environmental turbulence can influence the state of the drone's flight at all times. The infographic describes how the blade direction can affect speed and altitude based on the tilt angle at any given time. If the blade of the propeller is traveling in a climbing direction, such as where the blades are exerting upward force, the drone shall increase in altitude within the same direction as the tilt. The same theory also applies to the retracting blade direction. Understanding these concepts is crucial to fine tuning software to offset any unwanted tilt, as well as to correctly fly in the case where tilt is needed to reach a given object.
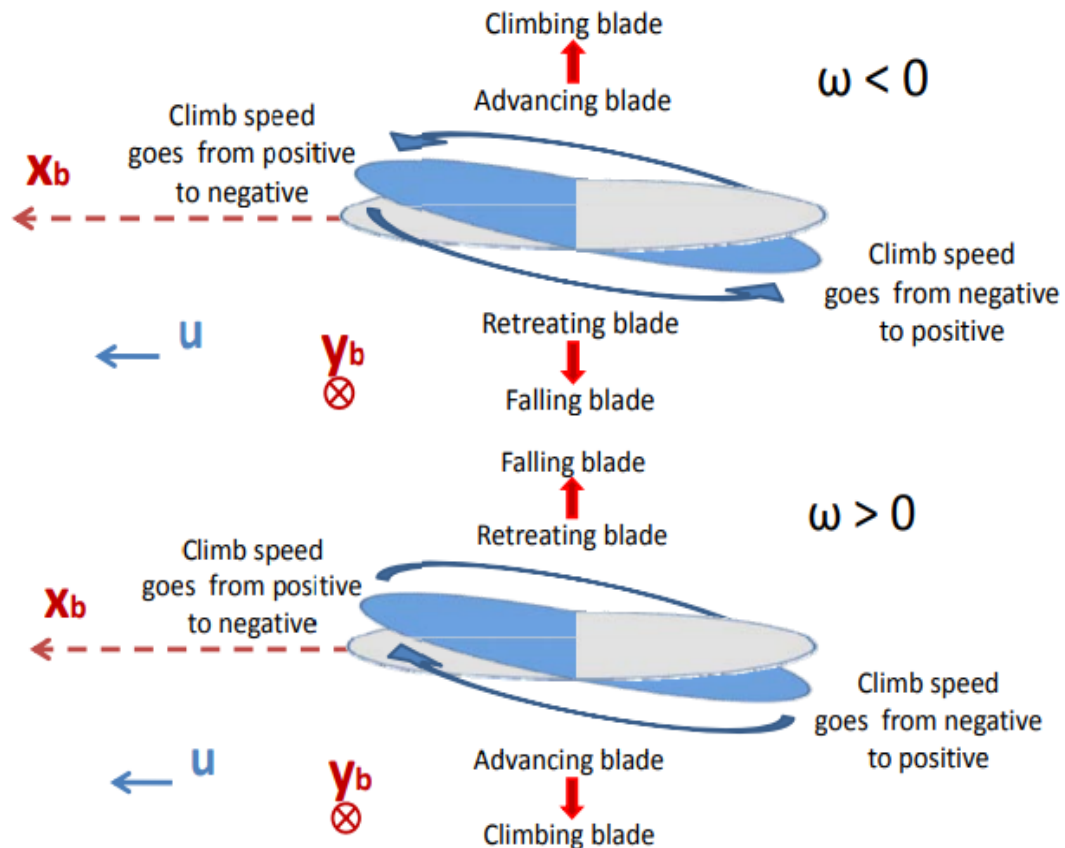

Figure 5.10.7: The tilt phenomenon needed to understand in order to build and fly the drone correctly
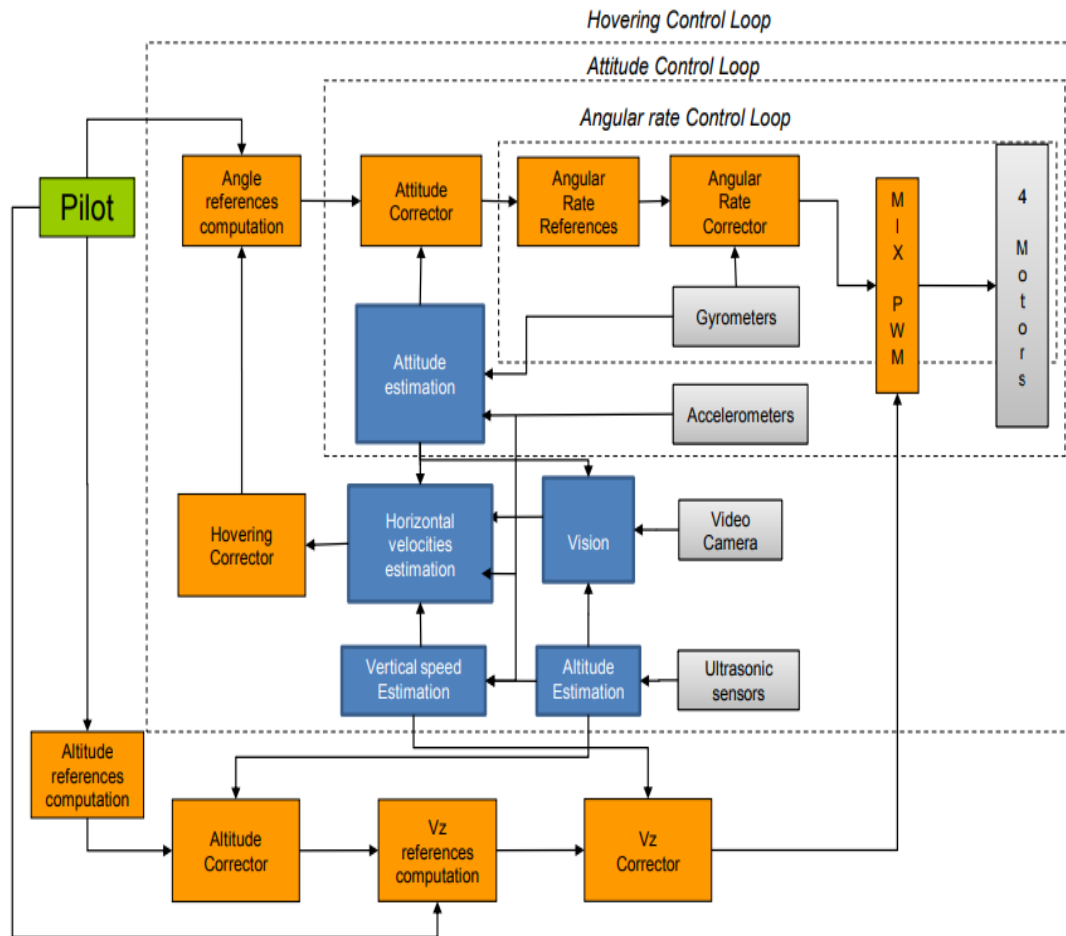
5.10.8 Control Principles and Loops

Figure 5.9.8: The architecture of the control principle for the three main controls we are aiming to achieve, hovering control, altitude control and angular rate control.

Understanding how the types of controls work is important when controlling the drone in different environments. Fully understanding these controls is essential to the steady flight of the drone. There are three main classifications of controls and loops they pertain to:

Hovering control loop: this requires altitude and speed estimations while also being able to correct any issues that may occur while hovering over a single position without any horizontal movement. This is also where the camera has a responsibility in determining the height of the drone at any given time in order to help with troubleshooting mid-flight.

Altitude control loop: This segment is crucial to avoiding obstacles in the positive and negative y-plane. By maintaining a desired altitude, objects that are above and below the drone can be ignored. The loop requires accelerometers

which is important in maintaining height given the natural force of gravity acting upon the drone.

Angular rate loop: This utilizes gyrometers to help maintain a stable flight angle, which also refers to the tilting phenomenon, by determining the angular rate at any given time and position.

5.10.9 Image Extraction and Feature Detection Reference

The camera installed on the drone will provide regional estimates for detected objects in order to determine distance, height, and width, given the PiCam's 3D scanning capabilities. These capabilities must be accurately assessed . Both Figure 5.10.9 and Figure 5.10.10 provide the logical description of a hypothetical image recognition software that the system would use to identify objects. These logical structures are necessary to successfully navigate the environment.
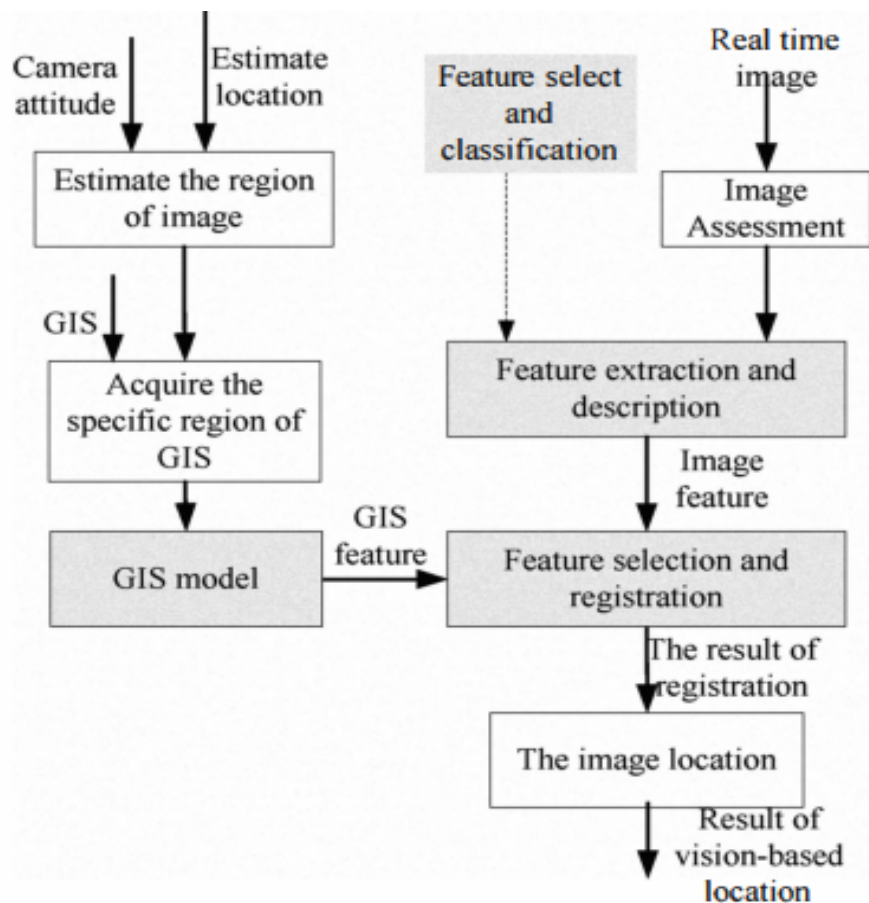

Figure 5.10.9: An example of a profile of a vision-aided navigation system with a geographic information system (GIS)
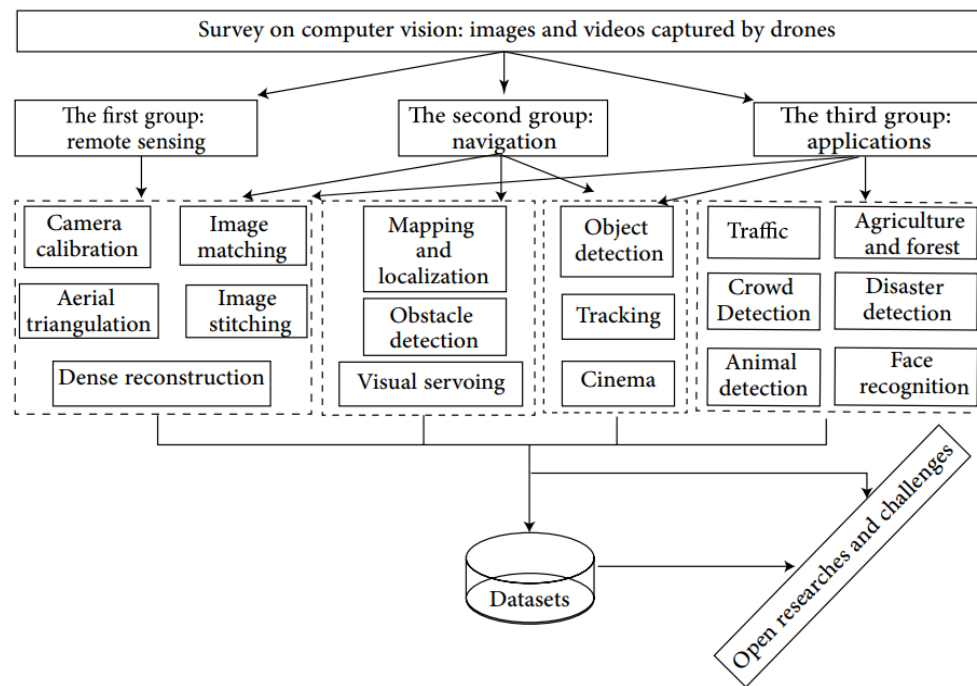
Figure 5.10.10: An example of an overview of a survey structure for vision navigation

Overall, the team has decided that one of the most important aspects of this project as a whole is the planning that goes into the image detection algorithm. Given that this software is one of the fundamental building blocks of the entire system, it seems to be an apt label. As with most computer programs, having the skills required to properly plan out and visualize the logic and flow of the code is just as important as being able to write the code itself. With the image detection algorithm and potential for subsequent navigation instructions this is especially true.

The algorithm needs to follow a clear structure, or else the subsequent code is just as tangled and messy. If one puts garbage into it, garbage will come out. With this report, the team has finalized a structure to the code, paving the way for them to write and construct the software in a rigorous, structured, and logical manner. This will put them ahead in the second semester since less time is spent worrying about the flow, and more time fixing minor bugs. The team believes that with this structure, major flaws and logical missteps can be avoided, which may prevent countless hours of time across the whole group.

The programmers and hardware focused members of the team all agree on how the drone will think, and they will strive to work in tandem to develop and transform these logical flowcharts into a real, working system. This agreement and stewardship towards each other will ensure a successful senior design completion.

6 Administration

6.1 Project Milestones and Deadlines

Throughout the course of Senior Design, which takes place between August of 2022 and is expected to finish in May of 2023, there shall be core milestones established within the section that are crucial to the success of this project. There are eight key deadlines that are to be met by the group.

The **first stage** of deadlines is between September and December of 2022. This is the stage where the documentation for the project must be completed and submitted. The primary goal of this stage is to complete a descriptive, 120-page paper that shows all the key components and methodology of the project, which is all described in the above sections.

The **second stage** of deadlines is between January and May of 2023. This is the testing and development stage of the project where all the components of the drone are to be ordered and assembled to create a finished product. Additionally, the testing environment shall be built within the stage so that thorough testing can be conducted.

There is a period in between the first and second stages that will primarily be used for more planning as well as part ordering and shipping. During this time, the team plans to go over this document and make any necessary changes should any revisions come up. This could be due to the failed delivery of a component, an out-of-stock component, or even an outright modification to the use of a component.

Listed below are the eight stages of deadlines which must be met by the end of each month in order to remain efficient:

September 2022 - At the beginning of the semester, the group must become familiar with methods of training a drone and come to a conclusion as to which tactics would be optimal for completing the project goal. The group must decide on specific aspects of the project to build on: which constraints, specifications, software, and machine learning algorithms to be used. They must use this information to complete the initial project documentation.

October 2022 - By the middle of the first semester in October, the sensing and imaging systems should be in the prototype stage, enough for the mid semester demonstration of its optical components. This is a required task for the photonic science and engineering student in the group. An in-person demonstration must be done to show the work they have done so far in the semester. At this time, the team should also have purchased a drone kit to study and use to adapt their own drone design. Initial designs of the drone should soon be completed.

November 2022 - Another in-person demo is required of the photonic science and engineering student in the group. This in-person demo will contain a more mature prototype of the systems demonstrated previously, and possibly another new system if one is created or ready to present. At this stage, the finalization of the parts list should be nearing completion. A proposal to the sponsor should be crafted to explain the necessary parts and final total cost. Also, the team should begin constructing the safety cage that the drone is tested in. This cage is located inside of a lab that the sponsor has provided them.

December 2022 - After the first semester is complete, the optical system should be prototyped and be functional enough to formally begin the training of the AI. The package for the convolutional neural network must be selected. Once selected, the CNN can be altered to fit our project better. The image recognition training can then be worked on. Also, all of the parts necessary for the drone to fly should be identified, and a proposal should have already been submitted to the sponsor to ensure delivery of most of the parts before the next semester.

Supply chain disruptions have been quite common in recent years, so the team should plan accordingly. They must act as if all of their parts may be delayed for a long time, so parts should be chosen well before they are needed. The group must begin assembling the drone no later than January 2023. There must be ample time to receive the parts and construct the drone to ensure there is plenty of time for the AI to be trained and integrated.

January 2023 - The final designs of the drone are reviewed and drone assembly should begin immediately with the parts that should have already arrived. As the hardware is being constructed, steps should be taken in order to ensure that all of the constituent parts of the drone can integrate together properly. Some group members have expressed concern due to experiences with past projects. Often, while individual pieces of the system may work well individually, when they are all brought together into the same system for the first time, they will likely not work. The group believes that consistent and clear communication among all of the team members when they are creating their individual parts is an essential part of ensuring significant integration issues do not arise.

The beginning of the on-board AI should also begin development using the imaging systems that had been prototyped in the previous semester. Once the drone is built, the simulation software can be set up and work alongside the CNN to train with live camera feed, rather than with a dataset of images as done previously. Any last minute plans should be drawn up now. After this point, the group may not be able to afford the time for any more additions to be made to the final drone's hardware plans, and the AI won't have time to train if there are more iterations that need to be done.

The PCB design should also be finalized around this time. The group needs ample time for the order to be constructed (and possibly assembled) by their PCB manufacturer of choice. Due to this component's importance in terms

of the final evaluation, it is absolutely essential that they get it working as soon as possible.

March 2023 - By the middle of the second semester, the drone should be assembled fully. Most, if not all of the hardware components should have been tested and quality checked. Importantly, the PCB that has been created should have arrived and be fully integrated into the hardware of the system. This is a key feature since the PCB is a required component of the final design in order to pass the class. At this point, the drone should be fully capable of hovering and maneuvering on its own without any external supports (i.e. ropes and/or supports). Given that autonomous flight is one of the core goals of the project, it should have reached this stage of operation. The autonomous flight cannot be tested if the drone cannot hover and do basic movements without assistance.

The neural network should have been well underway with its training as well, and steps should be taken to ensure everything can be integrated together. Drone should be able to identify objects within the view of the camera and classify them accordingly. If a LiDAR scanner has been created, then it should be fully constructed and integrated with the drone's commands.

May 2023 - By the end of the second semester, the onboard artificial intelligence should be fully trained using provided machine learning algorithms, and it should be fully integrated onto the drone. The drone should be able to easily recognize its predetermined objects, and take a string of commands and translate it into instructions that the drone can then form a route and execute on it. A full demonstration should be ready for the senior design competition.

First Progress Update (December 2022):

As of the date of submission (December 5, 2022), the group has made satisfactory progress toward each of the goals outlined in the previous sections of the milestones. The midterm demo allowed them to present a simple object detection system. This could detect whether an object has come too close to the drone system. It would send a signal to the completed system, and there would be an override of the controls, forcing the drone to stop. This system would satisfy their core optical goals.

In the fall semester, a second demonstration was done of the now updated optical components. This device was able to detect the exact distance away an object is as opposed to the presence of the object. This will give them more accurate location data. Furthermore, if it was sufficiently miniaturized, it could also be used to make the full 360 degree LiDAR scanner. They have also decided that this component is the one that will fulfill the PCB design requirements. So, the group is maintaining a rather optimistic attitude about the potential of the final optical system.

Furthermore, the group has finalized their plans for the drone construction and begun to order the necessary parts for the final system. After writing the report, the team has finalized their part selection for the drone with high confidence. The group has no reason to believe that construction of the drone will not be proceeding as scheduled above. The software development process has also been laid out completely, and in a manner that the group is satisfied with. The programmers of the team believe that they are able to create the image detection algorithm on schedule.

The software integration will follow the construction of the drone once a full flight test is completed. For this type of planning, a Gantt Chart may be used to determine which parts of the software need to be completed, as well as a dedicated timeline for each component of the software. By completing this gantt chart, the software team has a sufficient method of establishing a timeline for completing the learning algorithm for the drone as well as allowing enough time for testing the software.

Final Progress Update (April 25, 2023)

The group demonstrated their drone in front of their faculty review committee and was given approval for a successful project. While the team was not able to complete the autonomous drone navigation, they were satisfied with the overall project. The object detection algorithm worked exceedingly well. Additionally, the group was able to lower the processing time for object detection by integrating the LiDAR scanner with the system. By only running the image recognition software whenever the scanner detected an object near the drone, the drone would not waste any time or processing power detecting objects that were obviously not there in the first place. Furthermore, the drone was able to achieve steady flight, and though there was some complications flying the drone with the LiDAR scanner due to weight balance issues, it still functioned well on its own.

Perhaps in future works, advanced capabilities such as autonomous navigation, an extended ranging and mapping system, or a processor with integrated graphics installed to speed up detection time could be added. Although, for the time, budget, and skill restraints of this project, the group consider themselves successful.

6.2 Budget and Financing

After looking through several different kits for inspiration on how to build their own autonomous drone, the team identified some basic components that they would need in order to build the final system. These include but are only limited to: a lightweight frame, four brushless motors, four propellers, an electric speed controller, a power distribution board, a flight controller, and a rechargeable lithium ion battery of sufficient capacity. There will also need to be several other pieces of equipment that are used for the drone's sensing and

computing abilities. These other pieces include a camera (for object identification), some type of video transmitter (to send the video data back to a computer on the ground), and an on-board processing unit that can collect data from all the devices, and be able to store that data as well as the AI's navigation program.

The team estimated the cost of their own drone to be similar to that of a mid range consumer drone kit, which can be several hundred dollars. If the group were to reach its stretch goal of having a 360 degree LiDAR scanner, they want to have such components be cheaper than whatever may be commercially available.

The team has also analyzed lower budget mapping LiDAR scanners, which cost $100 or more. So, the components to such a scanner should be significantly cheaper. This is one of the primary goals of optical design. Creating a product that is cheaper than any reasonable commercial alternative is the primary way to justify creating these parts as opposed to simply buying several units off of a consumer website.

With all of this in mind and with consultation from the sponsor, the team came up with a total estimated budget of less than $1400. The team's generous sponsor, the UCF ECE department, has confirmed that they will completely finance any of the group's hardware needs within reason, They have defined a specific process to acquire any funding they may need. According to the sponsor's terms, in order to get funding for specific parts, the group must first submit a proposal to the sponsor.

This should include the price of the part, the date that the part is needed by, and most importantly, detailed reasoning and justifications as to why the team needs the part specified. The sponsor will then review the proposal and promptly respond to the team with whether or not they will choose to allow or deny funding for the part.

The team has decided that no further research into additional sponsorships is needed. They are receiving plenty of funding to complete the project. Given the estimated cost of all the components needed to build a complete drone, this budget is more than reasonable and the funding provided is more than sufficient. If any of the total cost goes over the proposed budget from the sponsor, then the team shall pay the remainder out-of-pocket.

The below table describes each component as well as the required cost for each. Based on the above budget for the project as determined by the sponsor, the total estimated cost for the project as a whole, not including extra parts to be added if necessary in the late development stages, comes out to approximately **$1350 USD**.

This number may be deemed higher than initially expected, but the group expects the total cost to be much lower as the price estimates are just overall

averages. Through the generous financial support provided by the ECE department, these costs is alleviated After considering the components explicitly described in Table 6.2.1 on the following page and considerations of any other miscellaneous components they may have to purchase, the team is rather satisfied with their ability to stay within any budgetary restrictions given to them. Please note that the below table lists market prices, and that the costs are subject to change at any given time.

Table 6.2.1: A breakdown of major parts and current market prices.

Item	Part	Cost
6.2.1	Drone Frame	$20
6.2.2	Four Brushless Motors	$15 each
6.2.3	Eight Propellers (four for backup)	$8 each set
6.2.4	Propeller protector rings	$11
6.2.5	Four Electric Speed Controller	$41
6.2.6	Power Distribution Board	$40
6.2.7	Flight Controller - Pihawk	$195
6.2.8	Battery	$16
6.2.9	Battery Charger	$15
6.2.10	Battery Fire-proof case	$15
6.2.11	SD card and SD Reader	$15
6.2.12	Pi module Camera (2)	$180
6.2.13	RC Video transmitter	$25
6.2.14	Sensors (height and lidar sensors)	$500
6.2.15	Raspberry Pi 4B	$225
6.2.16	Vibration Dampening Plate	$15
6.2.17	GPS and Compass Module	$23
6.2.18	Infrared Laser	$20

6.3 Conclusion

After full completion of this document, the group came to a consensus that all the work shown in this document has been of their own original work, and that all uses of outside sources have been referenced in the below section. Also, the team went through for white space and line spacing errors, as well as correcting any grammatical and numbering eros that may have risen during the final stretch. This document shows the full stages of development and while some necessary modifications may arise while the development stage commences, all crucial information has been covered.

There was an initial document that was submitted as a rough draft back in September2022. The constraints of this project have changed significantly since then, and this space is an area to acknowledge this. The original document has a different purpose,

Unfortunately, after the group's original sponsor retired from UCF, the group had to re-evaluate their project. Initially, the drone was meant to be able to interpret oral commands from a person standing away from the drone. This resulted in the largest change in the final product. The initial plan was to create a drone capable of visual language navigation, where a microphone would be added to the drone and it would navigate using the user's specific commands, rather than software based solely on the visual navigation capabilities of the drone. Regardless, with the original sponsor leaving, who had insight on this kind of navigation and machine learning, it was subsequently deemed to be too difficult for the group to complete especially since a functionality revision was made in the middle of the semester. This lack of guidance forced the group to abandon one of the oral interpretations using a machine learning goal. Thankfully, the UCF ECE department stepped in to act as a source of funding. This has alleviated the group's fears of not being able to afford the final project, which was a great amount of pressure on the team in terms of financials for a short time. Regardless, they are extremely grateful for the ECE department for helping them out in such an uncertain time.

The plans for the construction of the visual navigation drone show incredible promise. Following extensive research, the group has discovered several methods of developing an artificial intelligence with the capabilities to recognize specific objects in an area. They have also developed plans for ensuring the safety of the drone and

The group also sees a great potential if it were to be used in a more commercial setting, where the scope, scale, and budget would be far greater. With significantly more time and processing power available to the team, the image recognition AI could be trained to recognize more objects, with a greater efficiency. In such a setting, the drone could recognize more everyday objects rather than specific targets. It would be able to recognize trees, buildings, or even people. This could be combined with more mature instructions, where the drone

can fly by also changing its height, and not just flying in a rigid two dimensional plane.

Eventually, this drone could be integrated with voice commands, and the drone would be able to carry out specific tasks spoken to it by a human. Such a mechanism would allow for the drone to become fully autonomous, and it could navigate through an environment with minimal human effort.

Also, with higher quality parts, the object detection system could be made to be far more mature. With the budget for a commercial application, the LiDAR system could become able to create a three dimensional map of the environment. This would require much more expensive parts, and circuit components of a higher quality. The scanning system would have to rotate in three dimensions, and there would have to be enough processing power to accommodate this exponential increase in data. If the technology would be able to mature to this stage, it could be combined with the image acquisition algorithm. The system would then be able to tell exactly where an object is in its environment, its distance to it. From here, it could make calculations for the expected flight time, analyzing possible flight patterns and speeds to know exactly when it will arrive at the target. This fully matured three dimensional map could be used for very precise flight paths to specific objects, avoiding all objects and people efficiently. A more efficient flight path can have shorter completion times, and more satisfied customers for a potential business.

Unfortunately, the group would not be able to achieve such goals with the limitations facing them in a senior design project. Muscles are not grown stronger without strain, and knowledge cannot be grown without thought provoking questions. In the end, the group is satisfied with the current plans for the drone, and they see a great amount of potential in the applications of the technology if it were to be extrapolated into an industrial setting. They are and will remain fully committed to developing the best technology with the resources and time available to them.

In terms of the goals and timelines that were established at the beginning of the semester, the group has strictly followed these deadlines to get each specified section of this document done for rough draft submission. One major setback in terms of reaching the page limit for this final document was line spacing due to a software issue within Google Docs. But, with quick remediation, the document was reorganized and revised to fit all the contents in a correct manner. Beyond that restriction, all goals have been met as a prerequisite to the construction of the drone, with the exception of ordering parts as needed.

In further discussion, the part selection section of this document fully describes the comparison tables as well as the decision matrices that went into picking the right component with two main attributes in mind: efficiency and accuracy. The document describes how each specification of the component had an effect on the success of the project as well as how the component cost correlates to processing power, camera quality, and other key specs.

As discussed in the previous section, the budget and financing aspect of this project is almost entirely covered by the ECE department. But, since this is an engineering project, all specifications and constraints must be explained as all projects have a financial component to it. These explanations must be in depth and reasonably justified by the group before further action is taken. The team understands these requirements and are willing to work around any further needed requirements set forth by the sponsor.

Overall, the team feels as though this project as a whole will help them adjust to the thought processes and procedures present when creating a product in real-life applications associated with engineering. They understand that they are able to use every skill they learned about technology investigation, design, and construction from this project in order to develop numerous other creative solutions to unique problems in the future. This entire process required a great deal of investigation and planning, and the physical and software development of the final drone in the upcoming semester provides even more challenges that the team welcomes enthusiastically. The group remains strongly optimistic and confident heading into the next year. They will take all of the lessons they have learned about the hardware, software, and teamwork and carry it with them directly into the next semester and beyond as they reinforce their knowledge not just in university, but into industry as well. They will carry these lessons with them. For now, they will continue to put forth their best effort into this project. The final product at the end of Spring 2023 is their autonomous visual navigation drone, and they will make it the best it can be.

7 References

1. Qi, Yuankai, et al. "Object-and-action aware model for visual language navigation." *European Conference on Computer Vision*. Springer, Cham, 2020 https://link.springer.com/chapter/10.1007/978-3-030-58607-2_18#citeas.

2. Anderson, Peter, et al. "Sim-to-real transfer for vision-and-language navigation." *Conference on Robot Learning*. PMLR, 2021 https://arxiv.org/pdf/1904.04195.pdf.

3. Tan, Hao, Licheng Yu, and Mohit Bansal. "Learning to navigate unseen environments: Back translation with environmental dropout." *arXiv preprint arXiv:1904.04195* (2019) https://arxiv.org/pdf/1904.04195.pdf.

4. Zhu, Wanrong, et al. "Diagnosing vision-and-language navigation: What really matters." *arXiv preprint arXiv:2103.16561* (2021) https://arxiv.org/abs/2103.16561.

5. Wang, Hanqing, et al. "Active visual information gathering for vision-language navigation." *European Conference on Computer Vision*. Springer, Cham, 2020 https://link.springer.com/chapter/10.1007/978-3-030-58542-6_19.

6. US Department of Commerce, National Oceanic and Atmospheric Association (2012, October 1). *What is Lidar*. NOAA's National Ocean Service. https://oceanservice.noaa.gov/facts/lidar.html

7. *Light detection and ranging (LIDAR)*. Newport, Light Detection and Ranging (LiDAR) System Design. (n.d.). https://www.newport.com/n/lidar

8. Whyte, R. (2020, November 23). *Phase wrapping and its solution in time-of-flight depth sensing*. Medium. Retrieved November 4, 2022, from https://medium.com/chronoptics-time-of-flight/phase-wrapping-and-its-solution-in-time-of-flight-depth-sensing-493aa8b21c42

9. Oliver. (2021, April 19). Understanding indirect tof depth sensing. Azure Depth Platform. Retrieved November 4, 2022, from https://devblogs.microsoft.com/azure-depth-platform/understanding-indirect-tof-depth-sensing/

10. N. Yamamoto and N. Uchida, "Improvement of Image Processing for a Collaborative Security Flight Control System with Multiple Drones," 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA), 2018, pp. 199-202, doi: 10.1109/WAINA.2018.00087.

11. F. Schilling, J. Lecoeur, F. Schiano and D. Floreano, "Learning Vision-Based Flight in Drone Swarms by Imitation," in IEEE Robotics and

Automation Letters, vol. 4, no. 4, pp. 4523-4530, Oct. 2019, doi: 10.1109/LRA.2019.2935377.

12. A. Parker, F. Gonzalez and P. Trotter, "Live Detection of Foreign Object Debris on Runways Detection using Drones and AI," 2022 IEEE Aerospace Conference (AERO), 2022, pp. 1-13, doi: 10.1109/AERO53065.2022.9843697.

13. Ramarao, P. (2022, August 21). *Cuda 11 features revealed*. NVIDIA Technical Blog. Retrieved November 4, 2022, from https://developer.nvidia.com/blog/cuda-11-features-revealed/

14. *# Gazebo Simulation*. Gazebo Simulation | PX4 User Guide. (n.d.). Retrieved November 4, 2022, from https://docs.px4.io/main/en/simulation/gazebo.html

15. Raspberry Pi Trading Ltd. "Raspberry Pi High Quality Camera." Raspberry Pi, Apr. 2020, datasheets.raspberrypi.com/hq-camera/hq-camera-getting-started.pdf.

16. D. -Y. Gu, C. -F. Zhu, J. Guo, S. -X. Li and H. -X. Chang, "Vision-aided UAV navigation using GIS data," Proceedings of 2010 IEEE International Conference on Vehicular Electronics and Safety, 2010, pp. 78-82, doi: 10.1109/ICVES.2010.5550944.

17. DNDrone Nodes is an online communication platform that brings together experts and enthusiasts in drone research, start-ups, businesses, and educates about the newest technologies in the drone and FPV market. "Power Distribution Board PDB." Drone Nodes, dronenodes.com/pdb-power-distribution-board. Accessed 4 Nov. 2022.

18. Wikipedia contributors. "Elmer Ambrose Sperry." Wikipedia, 16 Sept. 2022, en.wikipedia.org/wiki/Elmer_Ambrose_Sperry.

19. Khofiyah, Nida An, and Rakhham Ardiansyah. Global Business Strategy for Commercializing a Technology of Drone: A Lesson Learned From DJI Drones and Parrot Drones. Detroit, Michigan, us, IEOM Society International, 2020, www.ieomsociety.org/detroit2020/papers/238.pdf.

20. Loianno, Giuseppe, and Davide Scaramuzza. "Special Issue on Future Challenges and Opportunities in Vision‐based Drone Navigation." Journal of Field Robotics, vol. 37, no. 4, Wiley, May 2020, pp. 495–96. https://doi.org/10.1002/rob.21962.

21. Akbari, Y., Almaadeed, N., Al-maadeed, S. et al. Applications, databases and open computer vision research from drone videos and images: a survey. Artif Intell Rev 54, 3887–3938 (2021). https://doi.org/10.1007/s10462-020-09943-1

22. Lee T, Mckeever S, Courtney J. Flying Free: A Research Overview of Deep Learning in Drone Navigation Autonomy. Drones. 2021; 5(2):52. https://doi.org/10.3390/drones5020052

23. S. Paschall and J. Rose, "Fast, lightweight autonomy through an unknown cluttered environment: Distribution statement: A — Approved for public release; distribution unlimited," 2017 IEEE Aerospace Conference, 2017, pp. 1-8, doi: 10.1109/AERO.2017.7943617.

24. M. Beul, D. Droeschel, M. Nieuwenhuisen, J. Quenzel, S. Houben and S. Behnke, "Fast Autonomous Flight in Warehouses for Inventory Applications," in IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 3121-3128, Oct. 2018, doi: 10.1109/LRA.2018.2849833.

25. Mohta, Kartik, et al. "Fast, Autonomous Flight in GPS-denied and Cluttered Environments." Journal of Field Robotics, vol. 35, no. 1, Wiley, Dec. 2017, pp. 101–20. https://doi.org/10.1002/rob.21774.

26. D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 2520-2525, doi: 10.1109/ICRA.2011.5980409.

27. M. Saska, T. Krajník, J. Faigl, V. Vonásek and L. Přeučil, "Low cost MAV platform AR-drone in experimental verifications of methods for vision based autonomous navigation," 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012, pp. 4808-4809, doi: 10.1109/IROS.2012.6386277.

28. Bristeau, Pierre-Jean, et al. The Navigation and Control Technology Inside the AR.Drone Micro UAV. Proceedings of the 18th World Congress, vol. Milano (Italy), The International Federation of Automatic Control, 2011, www.sciencedirect.com/science/article/pii/S1474667016438188?via%3Dihub.

29. Upadhyay J, Rawat A, Deb D. Multiple Drone Navigation and Formation Using Selective Target Tracking-Based Computer Vision. Electronics. 2021; 10(17):2125. https://doi.org/10.3390/electronics10172125

30. Brownlee, Jason. "A Gentle Introduction to the Rectified Linear Unit (ReLU)." *MachineLearningMastery.com*, 20 Aug. 2020, https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/

31. Gupta, Sparsh. "The 7 Most Common Machine Learning Loss Functions." *Built In*, 17 Apr. 2022, https://builtin.com/machine-learning/common-loss-functions

32. Kumawat, Dinesh. "7 Types of Activation Functions in Neural Network." *Analytics Steps*, 19 Aug. 2019, https://www.analyticssteps.com/blogs/7-types-activation-functions-neural-network

33. Mvs, Chamanth. "Activation Functions : Why 'Tanh' Outperforms 'Logistic Sigmoid'." *Medium*, Medium, 6 Nov. 2022, https://mvschamanth.medium.com/activation-functions-why-tanh-outperforms-logistic-sigmoid-3f26469ac0d1#:~:text=Both%20sigmoid%20and%20tanh%20are,lies%20between%201%20and%20%2D1.&text=Mean%20of%20sigmoid%2C%20tanh%2C%20and,set%20of%20numbers%20is%20considered)

34. Parmar, Ravindra. "Common Loss Functions in Machine Learning." *Towardsdatascience.com*, 2 Sept. 2018, https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23

35. Sharma, Pulkit. "Convolutional Neural Network Pytorch: CNN Using Pytorch." *Analytics Vidhya*, 10 May 2020, https://www.analyticsvidhya.com/blog/2019/10/building-image-classification-models-cnn-pytorch/

36. Wang, Benjamin. "Cross Entropy Loss in Pytorch." *Medium*, The Startup, 16 Jan. 2021, https://medium.com/swlh/cross-entropy-loss-in-pytorch-c010faf97bab

37. Gallagher, James. "How to Deploy a YOLOV8 Model to a Raspberry Pi." *Roboflow Blog*, Roboflow Blog, 30 Jan. 2023, https://blog.roboflow.com/how-to-deploy-a-yolov8-model-to-a-raspberry-pi/ .