UAV-based Powerline Inspection

Lebanese American University

Computer and Electrical Engineering Department

Capstone Design Project II:

COE/MCE 596

Project II Project Report

Supervised by Dr. Wissam Fawaz

Christy Daou

Jana Othman

Ryan Sam Bedran

Jad El Fata

Abstract – Powerline inspections have always been a challenge due to its cost and safety issues. As a result, it has been crucial to create new methods/mechanisms for inspecting the powerlines. Moreover, existing solutions typically highlighted the inspection of only one of the components of a powerline at a time. Therefore, in this capstone design project we design and develop a powerline inspecting drone that can inspect multiple components of the powerline. This report discusses mainly the hardware and software architectures of the aforementioned drone-based inspection system. In the process, our choices of the different building blocks that make up the proposed design are thoroughly discussed and justified while delineating the main challenges encountered during the process of building our drone-based system.

1 TABLE OF CONTENTS

2 TABLE OF FIGURES

3 TABLE OF TABLES

4 INTRODUCTION

Drones, also known as Unmanned Aerial Vehicles (UAVs), are proven to be very useful tools in many areas of life. They have been improved and developed to be equipped with sensors and cameras and thus to perform autonomous tasks. Drones are used in various fields including inspection tasks. They can complete hazardous tasks that humans cannot perform. In addition, they can save time, effort, lives, and money compared to other inspection tools.

Powerline are used to transmit electricity from supplier to consumer. They are composed of many components such as ground wire, clamp, insulators, and spacers. Electricity demand is increasing with the technological development that is witnessed day by day. This necessitates expanding the capabilities of transmission lines to meet this increase in the demands [1]. Moreover, to ensure faster electricity distribution, powerline operations should be regularly checked and inspected. The powerline infrastructure is the backbone for supplying the electricity to administrative, industrial, and private sectors. Consequently, to maintain a healthy powerline status, regular inspections are required.

The traditional ways of the inspection where specialized teams use manned helicopters and ropes to access the powerline are time-consuming since they require a long time of training and put humans' lives at risk. These methods proved their inefficiency [2]. Thus, a more efficient way for inspection is needed. One of the most recent and accurate ways nowadays consists of using a UAV. Along with a high range of sensors' modalities including electrical, positioning, and motion-related sensors, and a thermal camera, a drone can be used to inspect the powerline, automatically. As a matter of fact, the traditional ways of inspection depend on visual observations of the powerline which are then later analyzed based on the captured data. Alternatively, drones allow for more accurate inspection results relative to the other inspection tools. They can quickly and efficiently map out the area of interest, inspect it, and send the data to the ground station for analysis.

In this project, a drone equipped with a camera and high range sensors including LiDAR sensor, thermal sensor, accelerometer sensor, gyroscope sensor, barometric/altitude sensors, and other electrical sensors is used to detect and inspect powerlines. Inspection of a powerline is not a trivial task since there is more than one component that is mounted onto the powerline such as insulators and clamps, to cite a few, and all of these components need to be inspected. In this

project, a drone is tasked with inspecting the powerlines by performing the following tasks: wire inspection, insulator inspection, clamp inspection, and Ultra Violet (UV) defect detection. Insulators are detected and inspected through the high-resolution images captured by the camera used by the drone. Clamps defections are detected and analyzed using a thermal sensor. The other discharges that may occur at the high voltage powerline are detected using a sensor that helps capture these discharges. All of these inspection tasks are carried out with the help of machine learning and specifically deep learning. The data analysis is done offline using several methods that will be presented in greater details, in what follows.

Moreover, a web application is to be implemented with an easy frontend so that the non-expert users can use it easily and a backend that is going to be connected to the database storing the data collected by the drone. This application allows the user to view a live streaming of the powerline while the drone is inspecting it. It contains buckets that holds inspection information and based on the content of these buckets, a report of the available data is automatically generated for user perusal.

In short, the system is designed to:

- go to the towers automatically by using a path-planning algorithm explained in details in subsequent sections,
- inspect them one by one using specified machine learning algorithms for each powerline component to be inspected, and
- finally send the data to the ground station that uses a non-relational database coupled with a web application. This way, the inspection data results are stored for future analysis and processing.

5 PROJECT CONSTRAINTS AND STANDARDS

This project has several constraints, which are delineated in the table given below.

5.1 PROJECT CONSTRAINTS

Table 1: Project Constraints

Economic Constraints	
Cost	The proposed solution needs to satisfy our requirements at a reasonable and affordable cost, especially during the prevailing Lebanese economic crisis. Our final cost should be constrained by the cost of our essential building blocks: 1. Raspberry Pi 4 4GB: the SBC is already available at our disposal. 2. Hardware Components: these components will be the core components of the drone. The following includes a controller, telemetry modules, brushless motors, drone body, camera, thermal imaging, LIDAR sensor. The price of these components will add up to around $550 3. Ground station component such as a joystick controller is needed. The estimate price will be $50
Functional Constraints	
Weight	The drone should not exceed a weight of 2kg, to maximize flight time and agility
Battery	Battery should be powerful enough to handle a flight time of 20 minutes before returning to the base station
Response time	Power grid analysis should take between 2 to 4 minutes. The drone needs to analyze as many grids as it can
Telemetry	The drone and the ground station should have a long range fast and stable connection with low latency, to be able to reliably transfer footage such as thermal images and live video feed
Safety Constraints	
Proximity	To prevent the drone from colliding with objects and cause it to crash, a LIDAR sensor will be installed to detect and avoid object at a certain proximity, from all angles
Connection	The connection between the drone and the ground station should be secure.

5.2 PROJECT STANDARDS

Table 2: Project Standards

Standard	Standard Number	Description
Bluetooth	IEEE 802.15.1	Open wireless standard used to create a personal network between devices. It uses the 2.4GHz spectrum. It is mainly used to stream audio to nearby devices such as headphones. In this project, Bluetooth will be used to pair a wireless controller to the ground station
Cellular Networks	IEEE 802.15.4	Standard that was developed to provide a framework and the lower layers in the OSI model for low cost, low power wireless connectivity networks. LTE (Long-Term Evolution) will be used to communicate and exchange data between the ground station and the drone.
Global Positioning System (GPS)	IEEE 802.22	A System that allows to determine the location of any object using satellites orbiting around the earth. It is mostly used in navigation and tracking. GPS will help us locate the drone, and give us additional information such as the altitude.
Wi-Fi	IEEE 802.11	A family of specifications developed by the IEEE for wireless LAN (WLAN) technology. WiFi may be used to connect the ground station to a router with an internet connection to be able to communicate with the drone.
Radio Telemetry	IRIG 106	A comprehensive telemetry standard to ensure interoperability in aeronautical telemetry application. It will be used to communicate with the drone as a backup in the event of a malfunction with the primary communication standard.

6 BACKGROUND

Powerline inspection is a risky task to be performed by electricians even when the proper safety precautions are observed. The motivation behind this project was to decrease the risk on human lives as well as to meet the increased demand for electricity by continuously checking the powerline towers and maintaining their safety and operation. The risks that the powerline workers experience could be mitigated using a safer, more accurate, time-efficient, and cost-effective approach through the means of a drone-based inspection instead of the traditional inspection methods. Moreover, drones DJI (Da-Jiang Innovations) has already created a drone, Matrice 300 RTK (Real-Time Kinematics), that has all the features and components we are using in our drone, but it costs more than $13,000 [3]. This price is not affordable, not to mention it violates our cost-related constraints, especially with the current situation of the Lebanese currency, and the current financial situation of the country where cost-effective tools are much needed. Our goal is to build such a drone with less cost and to make it to the extent possible as efficient as the aforementioned DJI drone.

Given the lack of technological aspects in the electricity field in Lebanon and given that there are no methods used in Lebanon that regularly keep on checking and identifying the powerlines operation, there are many disastrous electrical problems. Therefore, this project aims to help in solving many issues related to the electricity sector in Lebanon and thus helping in solving the problem of electricity shortage. This is particularly true since the current electricity situation in Lebanon is extremely bad. Regular inspection of powerline has the ability to decrease the need for cable maintenance and thus reduce the financial burden on the Lebanese government.

Under these circumstances, a solution like the one described in this report becomes necessary. Particularly, this project aims to realize drone-based inspection by both building a drone with reduced and affordable cost and then deploying it to inspect powerline in an efficient manner. This work complements the continuous worldwide endeavor looking into powerline inspection using a drone system and which has recently attracted a lot of attention.

7 EXISTING SOLUTIONS

Powerline maintenance has always been an area of concern, and thus many researchers have dedicated a lot of time and effort to providing the most efficient and economically viable solutions to this problem. However, powerline maintenance is complex in the sense that there are several aspects to be evaluated from path calculation, faulty wire detection, and insulation inspection to many other problems. As a result, in this section of the report, all of these aspects will be thoroughly discussed to help develop a realistic and useful design.

7.1 DRONE FUNCTIONALITY

7.1.1 Path Construction

Route planning of the drone has been thus far so neglected that almost all of the previous studies have focused on the hardware design and flight control features of the drone problem but not on the inspection planning aspect of the problem [4]. However, the very few studies that do discuss route planning focus more on the robot/drone control, but not on the need to take into consideration an optimized or efficient route planning algorithm for the drone. As a result, taking into consideration routing algorithms for efficient inspection would be highly beneficial in extending the very limited drone battery life.

Moreover, a two-layer routing for high-voltage powerline inspection research effort [4] has been conducted previously which cooperated a ground vehicle and a drone. The purpose behind the study was to develop an optimized drone and ground vehicle route such that the drone would inspect the powerlines in a strategic path while the ground vehicle is also smartly placed to help recharge the drone. In other words, to avoid placing several charging stations, the authors of the paper decided to move the charging station as the drone is performing its inspection with a view to lowering the overall cost of their proposed solution. As a result, a two-layer point-arc routing problem was considered. Moreover, some of the restrictions and assumptions that were taken into account in the paper in question when defining the efficient routing algorithm include:

1. The power consumed by the drone in inspection mode is higher than that associated with a regular flight mode.

2. The edges of the powerline networks will be inspected by the drone, and every edge is visited only once.
3. The ground vehicle must return to its starting point, and it should also reach its next destination before the drone needs to be recharged.
4. The drone must head to the ground vehicle before the battery is fully depleted.
5. The route of both ground vehicle and drone must be successive.

Accordingly, to help split the problem into two simpler routing problems, the first heuristic used was the Cluster First, Route Second. This heuristic consists of dividing the routing problem into two simpler problems, a capacitated arc routing problem, CARP, and a traveling salesman problem, TSP. Using this approach, all powerlines were inspected by creating a set of drone routes taking into account the drone's endurance capacity and time relationship between the drone and ground vehicle. Afterwards, a cluster function is applied to merge the generated routes in order to end up with a single optimal route. Another routing heuristic studied was the Route First, Split Second, which consists of generating a giant nonrealistic drone route covering all the powerlines, which do not take into account the drone battery limitation. As such, the main giant route is then divided based on the battery limitation of the drone to help find the maximum possible route given the drone's battery endurance. Comparing both heuristics, the results suggested that for small to medium scale instances, the results provided by the Cluster First, Route Second algorithm were around 11% more efficient. However, when considering large-scale instances, the Route First, Split Second algorithm would be more efficient, by around 11% as well [4].

While the above-mentioned study shows promising results, it is highly dependent on having an already existing blueprint/map of the powerline system. Therefore, a more dynamic routing algorithm needs to be considered. If we are to place a LIDAR (Light Detection and Ranging) sensor in our system, then we could use it to generate our route in a dynamic fashion. As a result, using a point cloud classification system, the raw LIDAR data can be partitioned into subsets, which include five main classes: buildings, ground, conductors, vegetation, and pylons. Furthermore, the interesting aspect of using a LIDAR sensor would be that both unsupervised and supervised classifications can be used, providing us thus with less restrictions. While the heuristics discussed in [4] have shown promising results for unsupervised methods, once can investigate other alternative methods, which include hierarchical classification, Hough transform, cluster analysis, random sample consensus, and eigenvalue estimation. For the supervised methods, our

classifier options would include JointBoost, random forests, and support vector machines which are commonly used to extract the powerline components that are of interest to us [1].

7.1.2 Insulation Inspection

Insulators are very critical components in powerlines as they, as their name suggests, provide electrical insulations but they also at the same time help support the weight of the conductors on the pylons of the powerline. Insulators are generally ceramic in material and thus, some of the common damages include chipping caused by hail storms or heavy contamination. As a result, these insulator damages are physical and thus should be easily detected using a camera [5]. However, choosing how to take these images and how to analyze them is the challenging task and hence, two approaches will be discussed.

The easiest and simplest way would be taking pictures of all insulators on one side of the traverse and then taking pictures of all insulators from the second angle. While this is extremely beneficial when the flight time is short, this would be an effective method only when the powerlines are supported by a single circuit. Another option would be inspecting insulators one at a time. In this way, a picture is taken of the right side on the first traverse, the drone would then move left to take a picture of the second angle and then move to the middle to take a capture of the isolation element. In other words, we take a picture of the right, left, middle, and up angles of the insulator. However, even though this technique allows for inspecting the insulator in a semicircular manner, there is another way to inspect the insulator. For instance, the drone, similarly to the previous method, would be taking a picture on the right and left sides of the insulator, but instead of moving to the middle, the drone would go back to the right and then from the right side traverse to the middle. The first method includes a combination of the above mentioned options to help inspect these insulators. Although this allows us to have a detailed inspection, it is nonetheless time consuming, taking around 12 hours for inspecting 11 towers, and also consumes a lot of storage. For example, for around 8 hours of footage, 155 GB of storage were needed [6]. As a result, another approach would be to take Red Green Blue (RGB) images around the insulators and then these images would be passed into a deep learning classification algorithm. One might argue that images still need to be taken and storage will remain an issue. But this is not true since the amount of pictures taken will be reduced depending on the accuracy of the classification algorithm, for example the

ResNet101 convolutional neural network achieves a 96% accuracy. This implies that with only a few images, we could portray an accurate picture of the insulator condition [5].

7.1.3 Wire Inspection

7.1.3.1 Inspection Using Camera Event Tracking Algorithms

The use of event cameras has become more popular over the years due to their inherent robustness against low latency, motion blur, and high dynamic range [7]. As a result, given the properties that event cameras provide, they have become extremely attractive for robotic applications and specifically powerline inspections. However, while these event cameras are beneficial in performing agile maneuvers when avoiding powerline masts, their unconventional output bring up new challenges. For example, when tracking objects, specifically lines, the frequent change in appearance and sparsity of lines make it extremely difficult to track the lines properly [7]. Therefore, existing studies aimed at exploring different techniques to tracking these lines and the evaluation of the proposed method was based on correctness (ability to track distinct lines), instance (have a unique identifier when tracking lines), and persistence (how long a line can be tracked).

One of the recent and popular methods use Hough transform, in which the algorithm looks for the lines, which contain the largest number of events in a specific parameter space by making use of the polar coordinates to prioritize the lines [7]. This method of line tracking was mainly used for high-bandwidth and low-latency in a previous study [8] to control a dimensional dual copter. To estimate the dual copter state, the proposed tracker worked on a sliding window of events along with a Kalman filter. As a result, by using a Haugh transform method, the correctness and persistence of the lines were prioritized but not the instance, meaning the same line can be detected more than once. In addition, it is very computationally demanding to perform this pixel to parameter space conversion since the parameter will have to be searched on an event-by-event basis.

While the Hough transform method tackles the correctness and persistence properties, latency is overlooked and as a result, the spatio-temporal method was then considered. In this method, lines are assumed to travel in small time intervals and approximately at a fixed speed. Therefore, using a local neighborhood for every incoming event, a normal vector in the spatio-temporal was computed. Then, events with the related normal vectors were clustered together to form a line [10].

While this method tackles the instance and correctness of the detected lines, the persistence aspect was overlooked. As a result, a new method was considered, which takes the spatio-temporal method as a base, but introduces a hibernation feature to improve the persistence ratio by a factor of 10 compared to the previously proposed spatio-temporal method [7]. This method consists of four main steps as illustrated in Figure 1. First, the events are filtered based on a refractory time and neighborhood filter, such that if the event passes the filter, then the algorithm tries to add it to an existing line, else the event is added to a cluster. In the cluster, the event is either placed in a cluster of similar events or if no existing cluster fits, it is then left unassigned. Moreover, in a separate thread, the existing lines and clusters are checked periodically to remove old lines, clusters, and unassigned events and to update the exiting lines and cluster estimates.

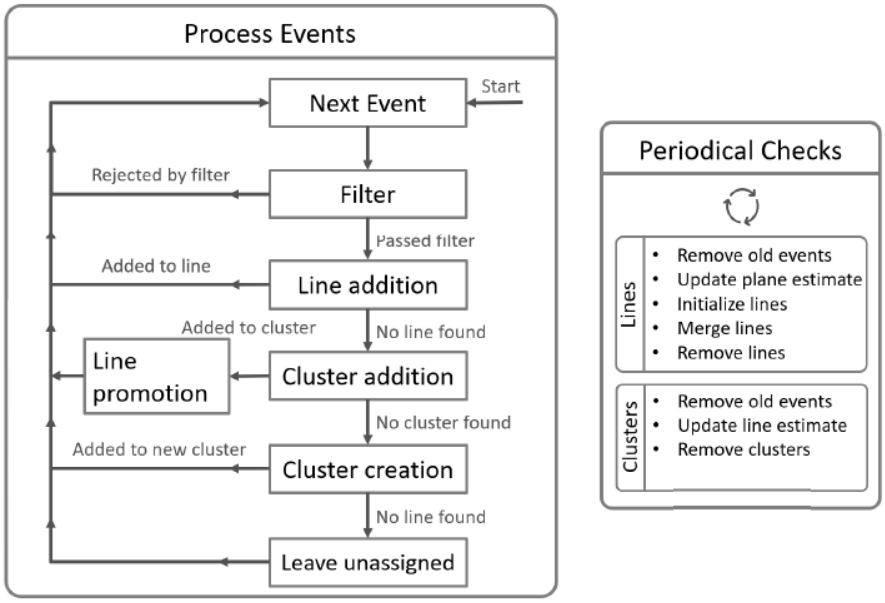


Figure 1: Overview of the new and customized spatio-temporal line detection method [7]

While this method looks promising, it does have its limitations. For instance, lines are assumed to have a constant velocity. This can pose a problem for rural areas where windy weather is the norm. In addition, for lines to be detected, several events must be fired. This makes it difficult to detect lines, should the drone fly parallel to the powerline. Moreover, in a texture rich environment, the tracker tends to detect "spurious lines", lowering thus the accuracy. However, one way of trying to solve these issues is introducing a hibernation feature to prevent the tracker from quickly detecting lines and allowing the drone to change location before starting the process again.

7.1.3.2 Inspection Using UV Defect Detection

In high voltage powerline (beyond 110kV), partial discharges, where strong electric field changes occur typically at the corners or sharp edges. Damaged or broken equipment along the powerline could be detected by partial discharges or by ionization of air resulting from discharges. This could be observed by the Ultra Violet (UV) sensitive camera as it captures images of the powerline and shows white blobs when detecting such discharge. In case multiple discharges are captured at a 3D location, this means that there is a problem in the infrastructure [9]. The existing inspection algorithm steps mentioned in [5], where it detects the bright or white blobs that represent the discharge, are shown in Figure 2. The first step of the algorithm is to detect the bright blobs that represent the discharge as projected into images. Then, some hypotheses regarding the 3D location of the discharges observed by the UV camera are created based on the geo-referencing approach. These hypotheses are tested using both more images from the camera and different viewpoints of the camera, which make the 2D observations more accurate. The aim is to reach a specific critical threshold of the 2D collected observations to detect the location of the partial discharge. Therefore, infrastructure elements such as the clamp, insulator suspension point, or conductor position are being searched for and the detected defect with the specified object are being associated. Finally, for an in-depth assessment to help identify the root cause of the problem, an operator can browse through the RGB images (with high resolution) of the specific location or component [5].

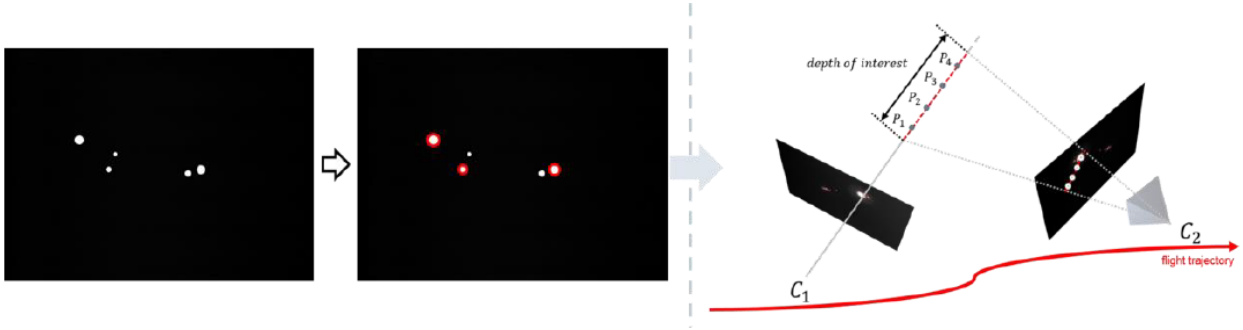


Figure 2: Outline of the UV inspection and partial discharge detection in 3D[5]

7.1.4 Clamp Inspection

Clamps are mechanical components used in powerlines for connecting high voltage powerline by providing all the needed mechanical support. One of the major reasons for clamp defection is the overload, which leads to wires and clamps heating up. In powerlines, some thermal effects of many components including the clamps are analyzed using thermal infrared sensors [5]. The clamp quality (i.e. connection quality of the wires) could be detected and evaluated by this thermal infrared spectrum using an appropriate camera to show this evaluation. Some clamps could heat up due to an internal corrosion or loosening of bolts. This leads to a poor connection quality by the clamp used. As mentioned earlier, this defect could be easily detected using a thermal sensor and a camera. The main challenge is though how to take thermal images of the clamps to correctly inspect them.

In [5], the authors used a high-resolution camera within the UV that can measure thermal infrared spectrum. The main approach used is taking RGB images of the powerlines clamps and then passing these images into a deep learning classification algorithm to decide on the quality of the clamp and thus the wire connection. The system at hand relies on the RGB images taken by the camera for clamp inspection. The process consists of projecting the 3D positions of the clamp into Infra-Red (IR) thermal images allowing for crop extraction and thus reducing the search space. This solves the issue of the storage since less images are needed. For the clamp inspection using thermal images, the measured thermal signatures could be affected by many factors such as the powerline load, surface condition, wind velocity, ambient temperature, solar radiation, sky, cloud coverage and other factors as highlighted in [5]. Moreover, the measurement may be affected by the distance to the powerline and the relative velocity. Doing experiments using an Infra-Red (IR) camera has proven that using the exact or absolute measure of the temperature could be somehow challenging and not very accurate [9]. Nonetheless, it would be more efficient to analyze the temperature differences instead of the absolute temperature to detect potential failures or hot spots. This makes the measurement feasible under dynamic conditions [9].

7.1.5 Drone Functionality Summary

In order to highlight the main strengths and weaknesses of the proposed functionalities methods, table 3 was created to provide a summary of sub-sections 7.1.1 through 7.1.4.

Table 3: Summary of drone functionality existing solutions

		Advantages	Disadvantages
Path Construction	**Cluster-First Route-Second [4]**	Is more efficient for small to medium scale powerline instances.	1. Time consuming process. 2. Highly dependent on the powerline system inspected.
	Route-First Split-Second [4]	Is more efficient for large-scale instances.	
	LiDAR sensor + point Cloud Classification System [1]	Is more dynamic in the sense that previous knowledge of the powerline blueprint is not needed.	1. More complex process. 2. More expensive solution due to the LiDAR.
Insulator Inspection	**Camera Captures and Video Stream [5]**	Provides a detailed inspection of the insulator and as thus is very accurate.	1. Extremely time consuming, such that the inspection of 11 towers can take up to 12 hours. 2. A big storage space is needed for video and images captured.
	RGB images + Deep Learning Classification Algorithm (ResNet101) [6]	This method is 96% accurate.	Already built classification algorithm that reaches an accuracy of 96%. Not as much input is needed.
Wire Inspection	**Hough Transform [8]**	Correctness and persistence of the lines were prioritized.	1. Same line could be detected more than once. 2. Requires a lot of time-consuming computation.
	Spatio-Temporal [10]	Latency and correctness were prioritized.	Persistence was overlooked
	Spatio-Temporal + Hibernation Feature [7]	Latency, correctness, and persistence are prioritized.	1. Lines are assumed to have constant velocity. 2. More difficult to detect lines when the drone is flying parallel to the powerline wire. 3. In texture rich environments, “spurious lines” can be detected.

	UV Defect Detection [5][9]	Relatively not expensive and simple detection.	1. Can be used for high voltage powerlines beyond 110kV. 2. Cannot identify the source of the discharge (ex. Problem with clamp or insulator…).
Clamp Inspection	**Thermal Sensor Detection[5][9]**	Good for close range detection and more efficient method to detect potential failures.	Several factors can affect the readings of the thermal sensor such as weather and powerline load.

7.2 SOLUTIONS RELATED TO BATTERY AND CHARGING ASPECT OF THE DRONE

One of the main issues to be tackled were the drone flying time and charging methods, thus, several solutions were proposed and studied. The first solution considered is the use of a ground vehicle. As the name suggests, the main purpose of this station or vehicle is to act as a base for the drone, such that when the battery is low, the drone can recharge. The ground vehicle is usually strategically placed such that it can act as a base for technicians to control the drone and process the data and images captured [4]. As a result, the technicians would ideally move the vehicle as the drone is inspecting. This solves the battery and endurance limitation problem of the drone, allowing thus the drone to perform its task over a larger area. Moreover, by incorporating this ground vehicle into our design, our drone's efficiency would be positively affected. This would also enable a more stable communication between inspectors and the drone. However, given that we aim to have minimal to no control of the drone during the inspection process, this feature is of no interest to us. As a result, we are only left with one feature to investigate, namely the charging aspect. Moreover, considering our minimal interference with the drone, we could set-up either several charging stations, which the drone can fall back to when the battery is low, or a single main station. Setting up several stations would be much more efficient compared to one station. However, setting up these charging stations is expensive and given our limited budget, it is simply not feasible. As a result, we need to consider other solutions.

A second solution, which can be considered, is the use of an additional drone, since one could be recharging, while the other could be flying. The latter approach would have the following advantages:

1. We will have a backup drone in case the other one malfunctions
2. One can be used while the other charges
3. The battery can be integrated, meaning there is no need to build a shell/accessible compartment, which can potentially save space and weight
4. The process can be semi-automated, meaning when the first drone's battery depletes, while it automatically returns and lands on the station, the second drone launches and resumes the inspection process.

To fully automate the process in point 4, we thought about using a drone charging pad, so that the drone starts charging as soon as it lands. The main disadvantage was the budget being doubled since double the components are needed, not to mention that the starting price of a done pad is $650, which is why we decided to adopt another solution, namely having multiple batteries at our disposal. Although this method is less "automated" than the methods discussed previously, it is a more practical and budget-friendly solution; however, the budget will be dependent on the number of batteries needed.

Moreover, one factor that needs to be taken into consideration is the battery capacity. For instance, a battery ≥ 4000mAh LI-Po (Lithium Polymer) can have a flight time exceeding 30 minutes (For reference, the DJI Mavic 2 Pro, one of the drones with the best flight times of around 30 mins, has a battery capacity of 3850mAh). It is important to note that the battery weight and price highly depend on the voltage (11.1V, 14.8V, and 22.2V) and the C rating (the capacity of energy the battery can safely discharge without damaging the battery). This is why batteries of high capacity, C rating and voltage can reach prices of $70-$150 and can weigh between 400g and 1500g. For this reason, we had to find a balance between battery quantity, capacity, and budget in a bid to achieve the best desired results.

8 PROJECT ARCHITECTURE

8.1 ARCHITECTURE 1

Our initial idea was to build a drone from scratch since it can be designed for a specific purpose in terms of customizability while offering us full flexibility with the budget. However, the limitations were overwhelming, the majority being related to the design of the controller. The controller is the bridge and center of the drone that connects all other components together. So, building one from scratch requires extensive knowledge, a dedicated amount of time and testing, and can risk destroying the prototype in the event of a malfunction that is likely to occur in mid-flight. An important matter as well is the weight distribution of the drone, which plays a crucial part in terms of stability. The latter leads to the conclusion that building a drone from scratch will not guarantee a symmetric design, which may result in an unstable and hard to control drone. The following table lists all of the components that are required to build such a design:

Table 4: List of components to build a drone from scratch

Components of a Drone		
Raspberry Pi	Capacitors	Drone Body
Micro SD Card	Diodes	Resistors
Infrared Camera	Transistors	Battery
High Torque Motors	LED's	Supporting Frame
Propellers	Wires and Connectors	LIDAR Sensor
Buttons and Switches	Drone Controller	

8.2 ARCHITECTURE 2

Building a drone from scratch was not feasible due to its limitations, so we were prompted into using a programmable drone. The drone is prebuilt and gives users the ability to program it to complete specific tasks by injecting code through USB connection or its dedicated wireless receiver. The drone supports languages such as python, and even has its own simpler language, where the documentation is provided by the developer's website. Some drones that were considered were the CoDrone and the DJI Tello, which are depicted in Figures 3 and 4,

respectively. After further research, we realized that we would face some other limitations that would prevent us from going with a programmable drone. Some of the main limitations are:

- Programmable drones cannot carry more than their own weight, meaning we cannot add other components such as the sensors and Raspberry Pi, among others.
- The battery is rather small and has a flight time of 8-12 mins, not to mention that the flight time will drastically decrease if we were to add more components.
- If we were to modify the drone by installing more powerful motors and batteries, we would be discarding components that we have already paid for. This beats the whole purpose of "premade programmable drone".
- These drones are not fully designed to be heavily customizable, especially when it comes to adding weight. Hence, any slight modification of said drones may cause it to destabilize or may not be able to launch if the added weight is significant.

Figure 3: CoDrone Programmable Drone

Figure 4: DJI Tello Edu

8.3 FINAL ARCHITECTURE

In our final design, we opted for the "Hybrid" drone design. It combines the process of building a drone from scratch with that of using an easy to setup premade flight controller to help connect all of the components together. This will give us the best of both worlds. The flight controller that we went with was the Pixhawk 4 (given in Figure 5), which also comes with all sorts of integrated sensors such as an accelerometer, a gyroscope, a magnetometer, a barometer, and a GPS [11]. The controller also offers all sorts of ports available, including ones to connect to motors, radio telemetry, GPS module, and onboard computers. The controller is also designed to

be paired with and controlled by a Raspberry Pi using specific libraries. This gives the user a variety of choices when it comes to flying a drone. Using this controller also provides us with the ability to choose the components of our choice in terms of battery, motors, body and communication between the drone and ground station. Figure 6 illustrates the different modes of operation of the drone. It has an autonomous flight mode where the drone is flown autonomously and controlled by multiple sensors, specifically a LiDAR sensor, GPS module, altitude sensor, accelerometer, etc… Furthermore, a manual flight mode is used for emergencies in case of failures in the autonomous mode. Note in this regard that the use of a PlayStation controller would make it possible to control the drone. The image processing or software part of the drone are built on top of a camera and thermal sensor. The drone is also augmented with a web application that offers several crucial features such as a live camera feed. The sections below will dive in depth into the hardware components used, the software part of the project such as web application and database, and finally the use cases of the project.

Figure 5: Pixhawk 4 Flight Controller [11]

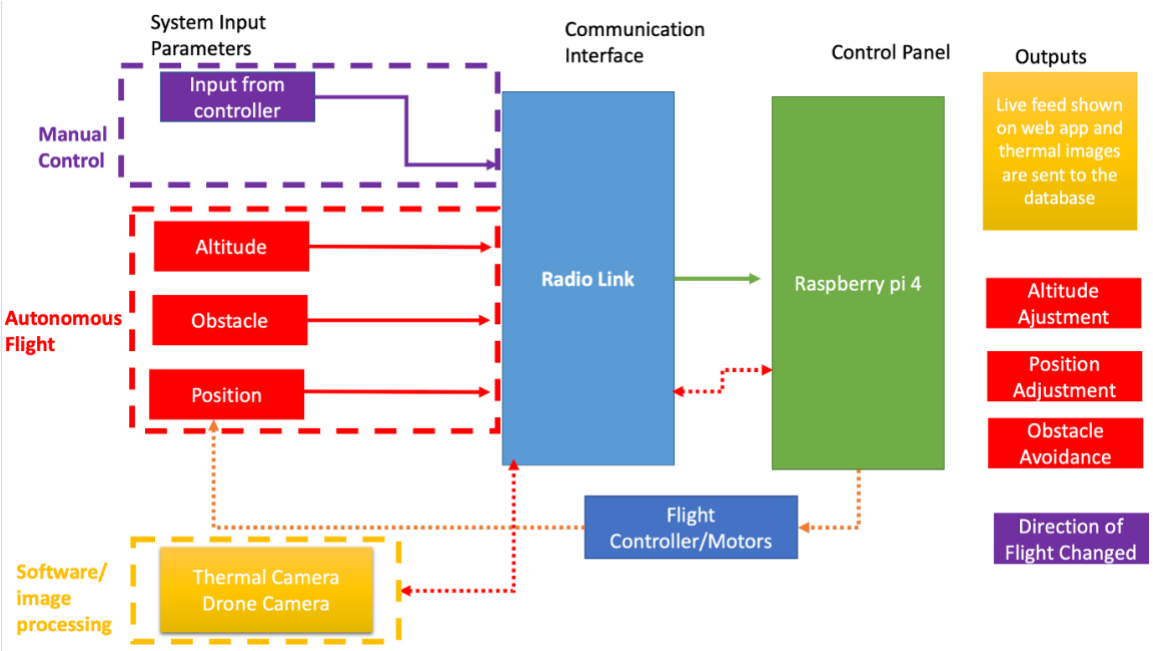


Figure 6: Drone mechanism diagram

9 HARDWARE COMPONENTS

We have looked up several types of sensors and actuators that can be used to accomplish our goal. In the section below, we will explain the sensors that we have chosen for our drone.

9.1 POSITIONING AND MOTION RELATED SENSORS

One of the most important building blocks of unmanned air vehicles or UAVs are motion and positioning sensors. These sensors allow the controller to receive information regarding the relative position of the drone as well as describe the environment around it. In this way, the controller is able to take appropriate decisions and move the drone to the desired position or height without the need for human intervention.

9.1.1 LiDAR Sensor

For the past 50 years, research has been directed towards LiDAR or light detection and ranging, which is a sensor composed of a laser transmitter and receiver. However, unlike regular proximity sensors, LiDAR sensors use light waves instead of sound waves or radio waves, making the response time much faster. The transmitter emits pulse light waves, which reflect off an object or obstacle and then return to the receiver as shown in figure 7. The time taken to leave, and return is then calculated by the sensor and thus the distance between the sensor and object can be found. This process is done millions of times per second, which allows for creation of a 3D map of the environment [12].

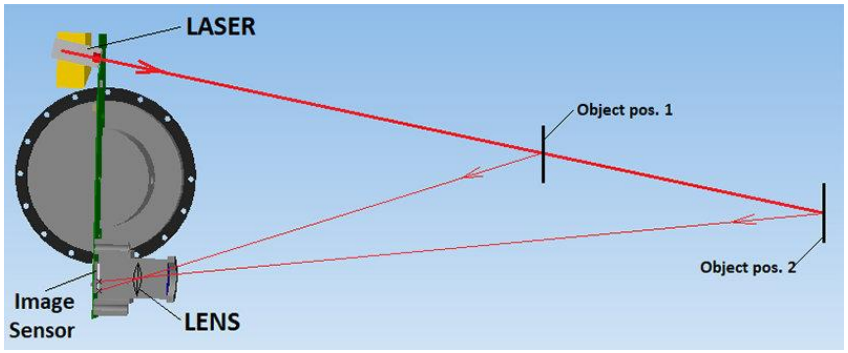


Figure 7: How LiDAR Sensors work [12]

Since we need precise obstacle detection and a very high response rate in order to adjust very quickly, then the best option is to use a LiDAR sensor. Moreover, LiDAR does exceptionally well

at long ranges compared to radar and ultrasonic sensors. Furthermore, LiDAR measurements are not affected by outside factors such as bad weather and lightning conditions. However, this depends on the type of LiDAR as well as the magnitude of the disruptive outside factors.

We have decided to use the Slamtec RoboPeak (RP) LIDAR A1 shown in figure 8 since it has an affordable price of $100 and delivers a scanning radius of up to 12m. Since transmission towers are generally placed in secluded areas, 12m of scanning is enough as there are not too many obstacles. Most of the obstacles to avoid would be trees and the metal bars of the transmission tower [13].

Figure 8 Slamtec RoboPeak (RP) LIDAR A1

9.1.2 Accelerometer

Accelerometers, like the ones depicted by figure 9, are one of the basic sensors used in moving vehicles such as cars, drones, aircrafts, etc... Accelerometers measure linear forces acting on the object and thus calculate the rate of change of velocity. However, accelerometers are affected by gravity and consider the acceleration of gravity in the object's acceleration. For example, if an object is in freefall, then the acceleration shown by the accelerometer would be 0, and when the object is resting on a flat surface, the acceleration is 9.81 m/s^2.

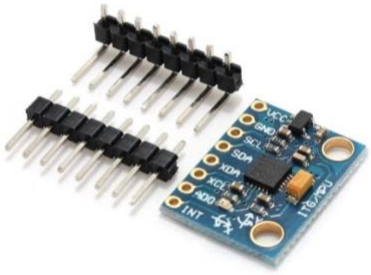


Figure 9: Accelerometer [14]

For drones, acceleration plays a crucial role. When a drone takes off, its velocity changes from 0 to the velocity provided by the motors. This sudden change in velocity leads to acceleration, and it is important to have a threshold to this acceleration since if the drone accelerates too much then it could reach instability or marginal stability and thus would require time to adjust. This event might happen very fast, and the controller might not be able to adjust at the same rate, which could lead into the drone hitting an obstacle, for instance. Therefore, the accelerometer is one of the most important sensors to have in a drone [14].

9.1.3 Gyroscope

A gyroscope is a sensor that measures the tilt, rotation, and angular velocity of an object. Nowadays, gyroscopes come in the form of MEMS (microelectromechanical system), which are microelectromechanical systems integrated on a chip or printed circuit board. Gyroscopes have a sensing mechanism that is shown in Figure 10. The sensing arm represents the reference axis. Hence any tilt or change in the gyroscope will result in the movement of the drive arms with reference to the stator. The gyroscope similarly can detect changes in angular velocity following the same principle. It is important to note that the gyroscope measures changes in tilt with respect to three reference axes in 3D space: yaw, roll and pitch also called Euler angles as shown in Figure 11.

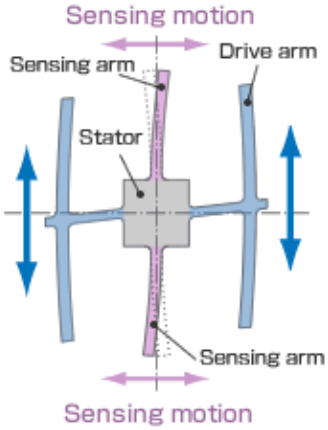


Figure 10: Gyroscope Structure[15]

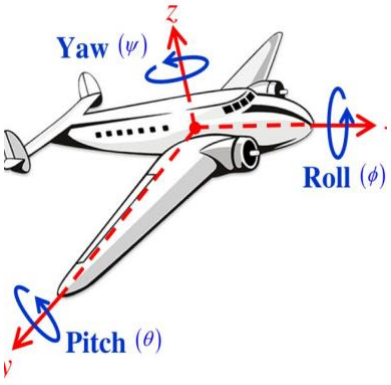


Figure 11: Euler angles [15]

For drones, gyroscopes are the most crucial components since maintaining a good tilt and balance is one of the essential aspects of drone control. In particular, the gyroscope ensures that the drone is always in a balanced position when moving or hovering [15].

9.1.4 Barometric/Altitude Sensor

Altitude sensors have a piezoelectric component based on the atmospheric pressure. When compressed, this component leads to a change in resistivity. As we go higher, the atmospheric pressure increases thus altitude can be determined by calculating the atmospheric pressure [16], this is illustrated in figure 12.

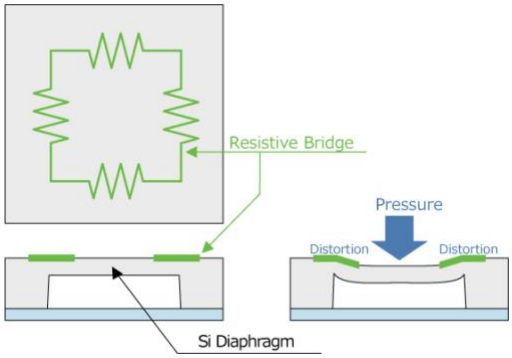


Figure 12: Altitude Sensor [16]

The drone's altitude is a very important piece of information. Since our drone is an autonomous vehicle, it needs to know how high it is flying.

We have found that the best option for accelerometer, altitude sensor, and gyroscope is the BerryGPS, depicted in figure 13. The BerryGPS includes on a single PCB a GPS, accelerometer, gyroscope, magnetometer (Compass), barometer/altitude sensor and a temperature sensor. It is also made for the Raspberry Pi. Since we will be using a Raspberry Pi, it would be a better choice to use an optimized sensor board. Moreover, using one board is much more weight efficient than using the several sensors especially that one of the biggest factors in drone performance is weight, which affects the motion of the drone as well as its flight time. The BerryGPS comes at a price of $70 and is portrayed in Figure 12 [17].

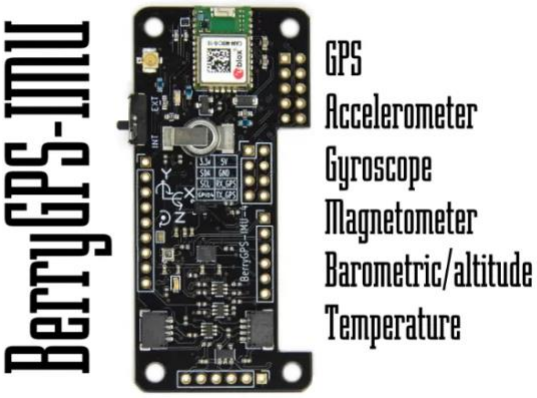


Figure 13 BerryGPS Module[17]

9.2 THERMAL SENSOR

Given that the main goal of the drone is to inspect powerlines using thermal imaging, a thermal sensor, such as the one given in Figure 14, is required to generate a thermal image, which is analyzed by a neural network to determine if the powerline is damaged or not.

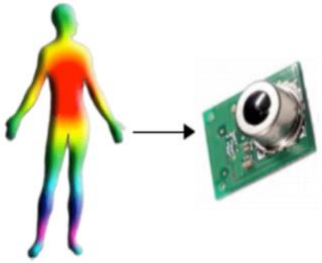


Figure 14: Thermal Sensor [18]

For our project, we will use the D6T-32L-01 thermal sensor. It can detect and generate a thermal image of 1024 or 32x32 pixels. Moreover, the area of detection is a square area compared to other sensors, which detect circular areas. It has several detection distances ranging from 1m where the detection area is 200cm x 200cm to 3m where the area is 600cm x 600cm. Furthermore, this sensor has a built-in microprocessor with an analog to digital converter and supports I2C (Inter-Integrated Circuit) communication, which is the connection that we will use with the Raspberry Pi as it is much faster than SPI (Serial Peripheral Interface) communication. This sensor has a resolution of 0.02ºC. This is a very accurate sensor, which is what is needed since we need precise measurements in order to have an accurate detection. This sensor has a cost of about $120 and is given in Figure 15 [18]. Its associated specifications are illustrated in Figure 16.

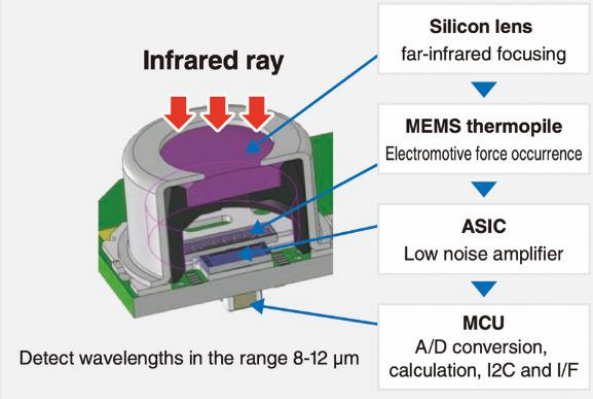


Figure 15: D6T-32L-01 Diagram [19]

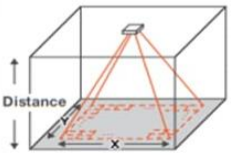
Sensor type	D6T-32L-01A
Appearance	
Number of elements	1024(32x32)
Number of elements X-direction Y-direction	X=90.0° Y=90.0°
Size of measurement area	
Distance 1m	X = 200cm Y = 200cm
Distance 2m	X = 400cm Y = 400cm
Distance 3m	X = 600cm Y = 600cm

Figure 16: D6T-32L-01 Specifications [19]

9.3 ELECTRONIC SENSORS

9.3.1 LTE and Radio Telemetry

Being able to control and visualize the drone's camera feed is crucial. This is why the fastest and most convenient way to be able to communicate with the drone with a high bandwidth and reliable range is the use of 4G LTE (Long Term Evolution). The following components are required towards this end [20]:

- A voltage regulator, such as the UBEC with 5V output and at least 5A output capacity: it converts the high voltage input from the battery to a lower voltage such as 5V, which is required for the 4G kit to run properly

- LTE module, such as the Sixfab 4G shield kit shown in Figure 17: The kit comes with the antennas, a bridge (connects the sim card, the module, and the raspberry pi together), and a miniPCIe module (Telit LE910C1-EU),
- Sim card for LTE connection.

The tools required to integrate the module in the drone include:

- 4x standoffs with M2.5 threads and 6mm length,
- Heat shrinks,
- Soldering iron and solder,
- Solder, and
- Scotch mounting tape

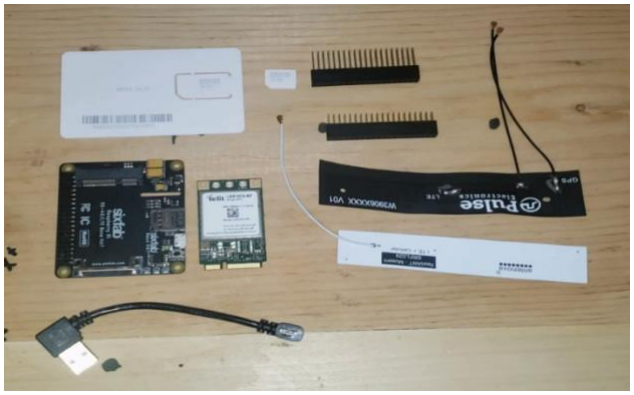


Figure 17: Sixfab 4G Shield

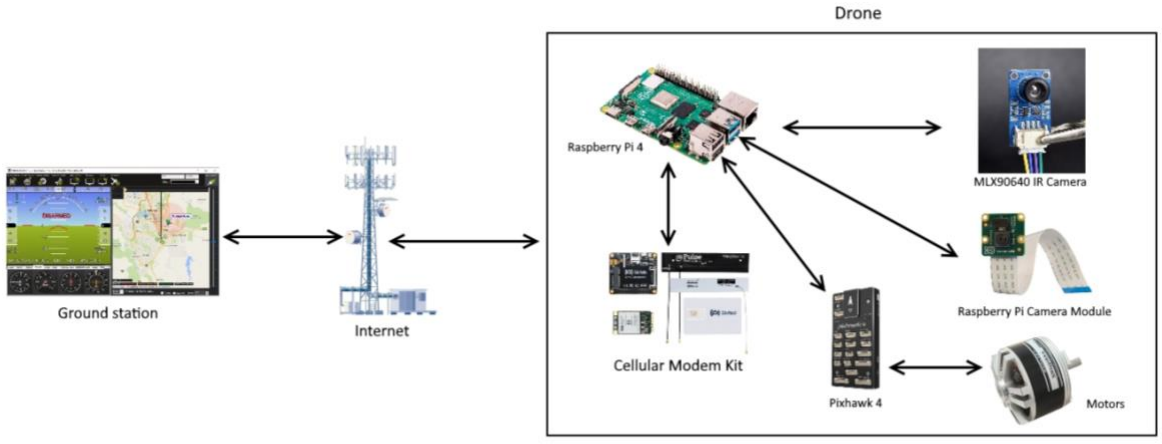


Diagram 1: LTE diagram

One of the benefits for this drone is that it can fly as far as we want it to as long as it has a cellular connection. An emergency protocol will be developed as well to enable the drone to return to the base station if it were to lose connection with the host. A downside of the use of LTE is that it is a subscription-based service, and a certain fee is required to be able to use it. The prices will depend on the subscribed bundle, and the cellular provider used. To account for that, a regular radio telemetry module with limited range (300m-500m) can be added to fly the drone, if the user does not wish to use LTE, or even if the drone is used in a place where cellular connection is unavailable. A good telemetry module is the 3DR (3 Dimensional Recording) Radio Telemetry V5 shown in Figure 18, with an advertised range of up to 500m.

Figure 18: 3DR Radio Telemetry V5

The downside of using it in the drone is the low bandwidth of 250kbps it provides [21], which is not nearly enough to transmit large files such as live video feeds. But, it is a good alternative in case the LTE module fails. In case the LTE option is not feasible, and to be able to get a live feed of the camera in action, a 5.8GHz wireless analog video transmitter, such as the one depicted in Figure 19, can be used.

Figure 19: FPV 5.8GHz AV Receivers

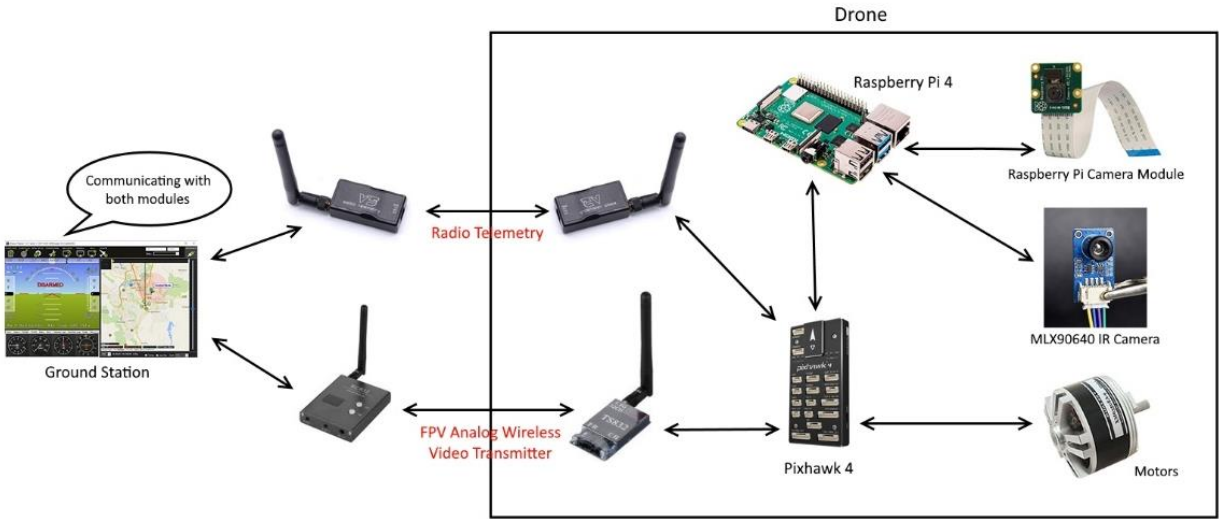


Diagram 2: Radio telemetry and FPV wireless video transmission diagram

Figure 20: Mission Planner

The software used to communicate with the drone is called the “Mission Planner”. A screenshot of its user interface is shown in Figure 20. The way it works is that a firmware is uploaded and installed to the Pixhawk 4, and using the telemetry device of choice, the drone will be able to communicate with the ground station [22], as illustrated in Figure 21.

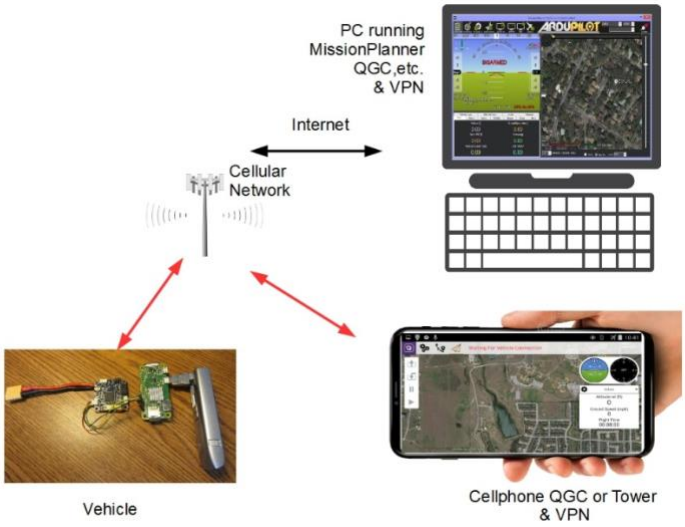


Figure 21: Mission Planner using 3G/4G Cellular Telemetry

9.3.2 GPS

The GPS is an extremely helpful and crucial part of the drone that lets the user detect its current location on a world map. The PixHawk 4 supports global navigation satellite systems (GNSS) using specialized receivers that communicate via the u-blox, MTK Ashtech or Emlid protocols, or via UAVCAN (UAV Controller Area Network). It also supports Real Time Kinematic (RTK) GPS Receivers, which extend GPS systems to centimeter-level precision [23]. The module illustrated in Figure 22 is connected to the Pixhawk using the dedicated GPS module port.

Figure 22: PixHawk 4 GPS Module [11]

9.4 CAMERA

The camera will be our eyes as the drone is flying; hence having a high-resolution camera is key to having good visuals. We will be using the Sony IMX477R (refer to Figure 23), as it is a very high-resolution camera having a 12 Mega Pixel resolution as well as a video quality that can go up to 4056 x 3040: 4K resolution. In addition, the Sony IMX477R is made specifically for the Raspberry Pi, which will be the single board computer that will be adopted for the project [24].

Figure 23 IMX477R camera

9.5 MICROCONTROLLERS

9.5.1 Raspberry Pi 4

To control, process information and make decisions, we will use the Raspberry Pi 4 as our main single board computer, which will play a major role in drone control and data processing. The Raspberry Pi has 8GB of RAM, which makes processing of data and later on implementing machine-learning algorithms much faster than other microprocessors or microcontrollers. Moreover, we will be using I2C communication to interface with our sensors and actuators. I2C allows us to have multiple devices interface with the Raspberry Pi without having a device conflict with other devices since it is an address-based communication, and each device has its unique address. Moreover, the Raspberry Pi has SPI communication as well which is faster than I2C and since there is no analog to digital converter built-in then, SPI would be a good choice to be used with an analog to digital converter. Most sensors and actuators used have Python libraries tuned and adapted to the Raspberry Pi. This makes the implementation phase faster and easier to modify, should we need to modify the behavior of a sensor to fit our needs. However, the primary benefit of using the Raspberry Pi is that it is fully compatible with the PixHawk flight controller and thus

machine learning algorithms for automatic flight can be directly deployed onto the drone, without the need to program from scratch an unknown flight controller. This feature has a huge benefit when it comes to increasing the flexibility of our drone as well as adding autonomous features to it.

9.5.2 PixHawk Flight Controller

The core of the project will be the PixHawk 4 flight controller (given in Figure 24). It is an advanced flight controller designed for autonomous flight and fully compatible with the Raspberry Pi. It has Python libraries designed to be used with the Raspberry Pi. Moreover, the PixHawk 4 has built in sensors, in particular:

- Accelerometer/Gyroscope: ICM-20689,
- Accelerometer/Gyroscope: BMI055 or ICM20602,
- Magnetometer: IST8310, and
- Barometer: MS5611

In addition, it has an integrated GPS module, which means that we do not need a separate GPS module. This helps us save on both cost and weight. Moreover, the PixHawk offers multiple communication options and interfaces, namely:

- 8-16 PWM (Pulse Width Modulation) outputs (8 from IO, 8 from FMU)
- 3 dedicated PWM/Capture inputs on Flight Management Unit (FMU)
- Dedicated R/C input for Combined Pulse Position Modulation (CPPM)
- Dedicated R/C input for Spektrum / DSM and S.Bus with analog / PWM RSSI input
- Dedicated S.Bus servo output
- 5 general purpose serial ports
- 3 I2C ports
- 4 SPI buses
- Up to 2 CANBuses for dual CAN with serial ESC
- Analog inputs for voltage / current of 2 batteries [25]

Figure 24 PixHawk 4 Flight Controller [11]

9.6 DRONE BATTERY AND FLIGHT TIME OPTIMIZATION

9.6.1 Drone Flight Time formula:

We can use the following formula to estimate the flight time depending on the battery level [26]:

$$Fly\ Time\ (Hours) = \frac{Capcity\ (Ah) \times Discharge(80\%)}{\frac{AUW(Kg) \times P\left(\frac{W}{Kg}\right)}{V(Volts)}} \qquad [1]$$

where:

1. Time is the flight time of the drone, expressed in hours.
2. Capacity is the capacity of the battery in amp hours (Ah).
3. Discharge is the battery discharge that you allow during the flight. As Lithium-ion Polymer (LiPo) batteries can be damaged if fully discharged, it is common practice to never discharge them by more than 80%.
4. AUW is the all-up weight of the drone - the total weight of the equipment that goes up in the air, including the battery, in (kg)
5. P is the power required to lift one kilogram of equipment, expressed in watts per kilogram. A conservative estimate of 170 W/kg will be used. Note that some more efficient systems can take less, for example, 120 W/kg.
6. V is the battery voltage, expressed in volts.

Note that the time calculated with our tool might differ from the real flight time. We assume that the drone mostly hovers in the air, but every time it performs some elaborate maneuvers, one of

the motors will have to work faster, meaning it will consume more power than usual. Hence, the following number can be used as a rule of thumb:

1. If the drone only moves around slightly (when the drone is inspecting the powerline), the flight time will be about 75% of the calculated time.
2. If the drone flies in the presence of strong wind or moves around a lot (moving towards the power grid), the flight time will decrease to around 50%.
3. If the drone is flying at high throttle levels (breaking, maneuvering), the flight time will decrease to 25-30% of the calculated value.

Statistically, while being conservative, the drone will be inspecting powerline for 65% of the time, moving from one point to another 30% of the time, and will be breaking and accelerating 5% of the time, meaning a certain constant must be applied to the final time to deduce the drone battery capacity. Hence the actual flight time is:

$$Loss\ Factor = [(0.65 \times 0.75) + (0.30 \times 0.50) + (0.05 \times 0.25)] = 0.65$$

$$Fly\ Time\ (Actual) = Fly\ time \times Loss\ Factor = Fly\ Time\ \times 0.65$$

9.6.2 Battery Charging time

The battery recharge time depends on the C rating of the battery. The equation is as follows:

$$charge\ per\ discharge = \frac{60\ minutes}{C\ Rating} \qquad [2]$$

Where results are in charge/discharge time [27].

9.6.3 Calculating final drone flight time and recharge time

Using an example Battery of Lumenier 4000mAh 6S 120c CineLifter LiPo Battery XT90 [28] with the assumption of a drone weight of 1.6Kg, we get:

$$Fly\ Time\ (Hours) = \frac{4 \times 0.8}{\frac{1.6 \times 170}{22.2}} = 0.261\ hours = 15.67\ minutes\ on\ idle\ speed$$

$$Fly\ Time\ (Actual) = 15.67 \times 0.65 = 10.18\ minutes$$

9.6.4 Additional Information

Table 5: LiPO Battery Size Chart [29]

Number of LiPO Cells	Nominal Battery Voltage	Common Quadcopter Applications
1S	3.7V	Indoor micro brushed (e.g. Tiny Whoops)
2S	7.4V	30-70mm micro brushless
3S	11.1V	100-220mm brushless
4S	14.8V	220mm brushless race/freestyle
5S	18.5V	220mm + brushless race/freestyle/quadcopters
6S	22.2V	220mm + brushless

The above table presents the numbers of cells required for different values of the nominal battery voltage. The following conclusions can be drawn when it comes to our project based on the table:

- The most common capacity for drones with 5-inch propellers is 1300mAh for 10 mins of flight time.
- Since the battery capacity and C rating are high, a 6S (22.2V) battery is needed.
- 75C pack is the recommended C rating, but to get the best out of the quad, C ratings of 80-100C and higher are recommended [30].

9.6.5 Raspberry Pi Battery

There are two ways for powering up the drone. The first way is to use the main battery along with a voltage regulator, since the battery delivers a voltage of 22.2V, while the Raspberry Pi has an input voltage of 5V (3.7V works as well). The second and safer way is to use a power bank to power up the delicate components. Any battery pack with a capacity of 5000mAh and has a 2.5A output current is enough to run electrical components such as the raspberry pi, cameras and sensors, and can last up to 3 hours.

10 SOFTWARE

10.1 RASPBERRY PI OPERATING SYSTEM

The Raspbian OS will be the main operating system used to code and test different parts of the project, such as testing the sensors, cameras, drone flight, etc… Furthermore, Raspbian offers a graphical user interface that allows remote access and control of the Raspberry Pi. Hence we will be able to test different areas such as camera, LiDAR, autonomous flight while having full control over the Raspberry Pi. However, the project will be coded in a single main script, which will be ran on startup once the project is complete, thus the Raspbian OS is an intermediary tool used for the project development.

10.2 DATA MANAGEMENT AND INTEGRATED DATABASE

The purpose of this section is to explore the choices available for our database. For the proposed project, we will need a database capable of storing a large number of images. The database will have two main functionalities, namely, to store the training images for the machine learning algorithm and to store the captured images from the inspection and a generated report.

10.2.1 Relational Vs. Non-Relational Databases

Applications commonly use relational databases and as such, generally have a good performance when handling a limited amount of data. However, with a larger volume of data such as social media, internet, and multimedia, the traditional relational databases become inefficient. As a result, to be able to handle this huge amount of data, NoSQL (Structured Query Language), non-relational, databases were introduced. NoSQL is a methodology composed of several competing and complementary tools and unlike relational databases, NoSQL can handle unstructured data like documents and multimedia efficiently. Moreover, non-relational databases do not use a relational database management system and thus do not store the data in schemas or tables, but instead have a simple data model, where identification keys are used to find data associated with that specific key. As such, there are four ways to store data in a non-relational database which include Key-Value (ex. Riak), Document (ex. MongoDB and Couchbase), Column (ex. HBase and Hypertable), and Graph-Oriented. Moreover, it is important to note that while NoSQL is much

more efficient than relational databases when a lot of querying is involved, joint operations are not allowed. As a result, to achieve a similar behavior, references are used [31].

10.2.2 Choice of Database

As discussed earlier, the main purpose of our proposed design is to be able to identify and detect defects within the powerline hardware, which includes insulation, clamps, etc…. As a result, choosing the appropriate database is crucial for storing both training images and captured images. Moreover, to be able to choose our database type, relational or non-relational, a deep understanding of our data is needed. For instance, given the nature of the proposed project, efficient image and video processing will be of top priority, therefore, the chosen database needs to support this. In addition, the database needs to support a huge image archive given that for a typical powerline inspection, at least 35 images must be considered for processing per tower. As a result, scalability is another factor affecting the database choice [32].

On a different note, it is also important to consider the type of images needed for processing. In other words, both training images and captured images may vary in type depending on the camera used and depending on the pictures used for training. While, we can easily control the training images, it would be difficult to do it for our captured images, since it would vary with the camera used, Digital Imaging or Raw format. Furthermore, to have a more dynamic detection software, it must be assumed that the media used is characterized as semi-structured [32].

As a result, if a relational database is to be considered, the following challenges need to be addressed which include:

1. Relational databases do not scale out.
2. Breaking apart images and videos and sharing them over several servers will be inefficient.
3. Search functions that are integrated are not available.
4. Storing the semi-structured data in tables is extremely difficult.

However, if a non-relational database is to be applied, then the above mentioned challenges that come with the relational database will be addressed as non-relational databases:

1. Provide automatic horizontal scaling.
2. Support the breaking apart of huge images/videos for a smother auto-sharing.
3. Support integrated search functions.
4. Are better suited for storing the semi-structured data compared to a relational database.

Therefore, given the nature of our data and the need for low-latency processing time, a non-relational database is the better option for the proposed project [33].

Once the database type has been decided, choosing the database platform is the next step. Given the many options to choose from, for this project, MongoDB was selected for several reasons. First, MongoDB can handle different image file formats. This tackles one of the previously discussed concerns. Second, the platform can also handle larger images/videos, which was also another concern. Moreover, MongoDB provides Binary-JSON (BSON) to exchange and store data. Additionally, some of the important features is that the images can be updated or changed individually, and all related images can be stored in one place, allowing thus for a faster retrieving capability as opposed to other platforms. Moreover, for larger binary files, MongoDB uses GridFS to handle them by splitting them into smaller "chunks" [32].

10.3 WEB APPLICATION

Our design will feature in addition to the drone, a web application which will expand and broaden the user's interactions with the system. It will combine most of the software into one graphical user interface. First and foremost, the web application will give the user the possibility to see a live feed of the drone's camera in order to have a direct visual from the drone's perspective. Moreover, the web application will allow the user to have a live feed of the thermal sensor and thermal imaging of the powerline as it is being scanned. Hence, the user is able to detect any defects directly in case it is apparent; before enabling the machine learning algorithms which are thoroughly discussed in a later section. In addition, our project features a database that can be accessed through the web application at any time. This database houses, among others, all previous recorded data in a way supporting easier access. Furthermore, the web application provides a login page so that only authorized person has access to the recordings and data.

11 MACHINE LEARNING AND AUTONOMOUS ALGORITHMS

The drone will use machine learning for autonomous flight as well as automatic powerline inspection. In order to perform these tasks, we will need to use both deep learning and reinforcement learning. Deep learning is autonomous, it is a self-teaching algorithm and uses existing data to train algorithms to find patterns. Then, it uses these patterns to make predictions about new data. On the other hand, reinforcement learning is an autonomous system that essentially uses trial and error. It is based on a reward system and performs actions with the aim of maximizing rewards. Hence, based on the reward granted per action, the system can make a decision [34].

11.1 MACHINE LEARNING FOR AUTONOMOUS DRONE FLIGHT

In order to have autonomous drone flight, we will need to use deep learning, specifically we will need two neural networks: a value neural network and a policy neural network, which are depicted in Figure 25. The value network is essentially a function that represents how good a certain state is and it is used to guide the policy training. The value network evaluates the drone at random states then checks the value at this current state and thus determines the appropriate action. On the other hand, the policy network is the "control" network for the drone. It is responsible for controlling the hardware of the drone. It takes the full state of the drone and based on the value network, it takes the appropriate action such as increasing the thrust of each motor. Moreover, the value network is updated through a series of iterations called exploration. Throughout each iteration, the state is read, modified, then another iteration takes place with the modification updated. Through this method the model is able to learn and have a more accurate value, which is then fed back into the policy network. The policy network (as illustrated in Figure 26) evaluates the state of the drone and takes the appropriate action. An essential part of autonomous flight is obstacle avoidance. Obstacle avoidance is achieved using reinforcement learning. When the drone gets close to an obstacle, the reward immediately starts decreasing and as the drone avoids the obstacle the reward starts increasing. Since the model is always looking to maximize the reward, the moment that the reward decreases, the drone will change its direction to where the reward will increase again [35].

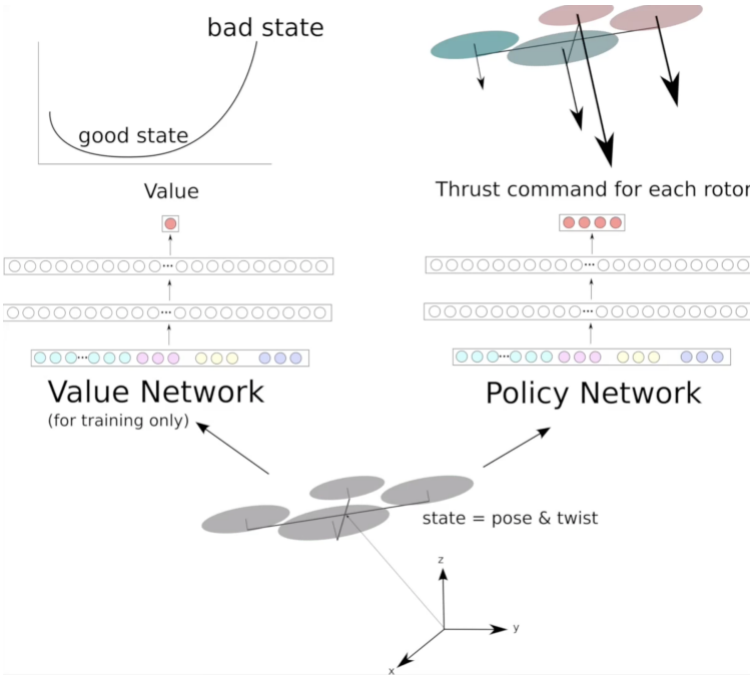


Figure 25: Value network and policy network

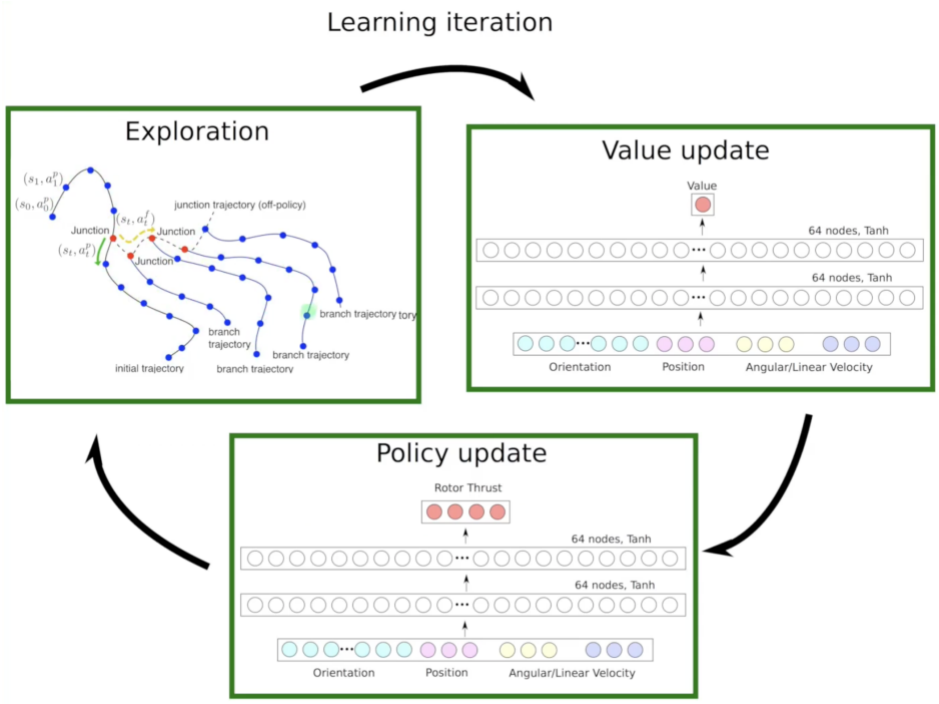


Figure 26: Machine learning model training

11.2 MACHINE LEARNING FOR POWERLINE INSPECTION

When it comes to inspecting powerlines with machine learning, we will heavily rely on deep learning. Powerline fault detection will be done using a convolution neural network. The Neural Network (NN) will be fed a large selection of powerline thermal images and damaged powerline thermal images. The model will be trained based on the images fed and thus it will be able to detect faulty powerline and healthy powerlines. Moreover, since the majority of powerlines are located in secluded areas and far from buildings, we will use color analysis as a primary factor for training. We will read from the thermal sensor the thermal image of the powerline and then run it through the neural network, which will output whether the powerline is damaged or not.

12 DRONE MODES OF OPERATION

Figure 27: Dualshock 5 controller

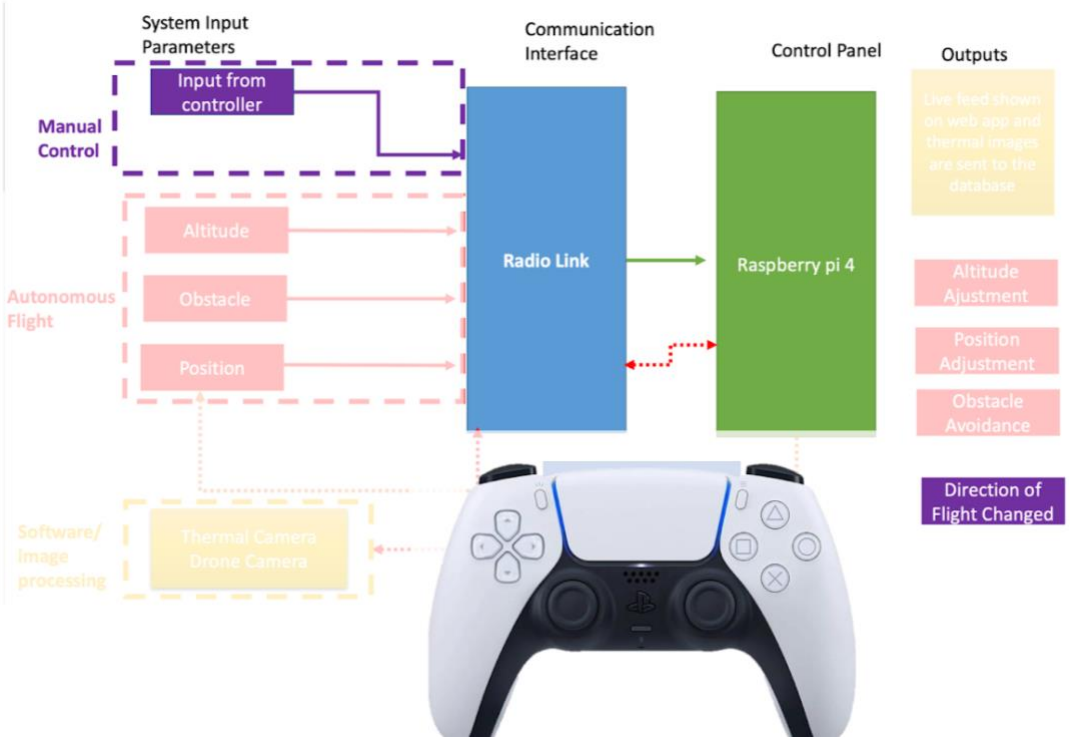


Figure 28: Done manual control

Figure 28 illustrates the mechanism of the drone and the ground station, and can be divided into 3 main parts. First the manual control, where the user input is sent from the ground station to the drone's onboard computer through the radio link. The commands are then sent from the onboard computer to the Raspberry Pi, to the Pixhawk, and to the motors (if LTE was used). In other words, a Dualshock 5 (refer to Figure 27) controller will be paired and used to manually hover the drone, while the ground station will have a screen for a FPV (first person view) if needed. Basically, the controller sends an input to the ground station, which is then sent to the Raspberry Pi, processed, sent to the Pixhawk, and then sent to the motors. The Raspberry Pi will also send information such

as altitude, GPS location, media such as thermal imaging and live video feed (if needed) to the ground station. The basic controls of hovering the drones will be as follows:

- When the user moves the left analog, the drone will move on the x and y axes.
- When the user moves the right analog, the drone will rotate around the z axis
- When the X button is pressed, the drone hovers upwards
- When the O button is pressed, the drone hovers downwards

Other buttons will be assigned to other functions such as toggling manual mode on/off, changing view mode (camera or thermal), taking pictures, starting video recording, etc… Moreover, the side web application will be delivering a live camera feed so that we are able to see the flight process from the drone perspective. Since it is the manual flight mode, having a live feed with as short of a delay as possible is crucial to avoiding obstacles and reaching the desired destination.

When the automatic control, depicted in Figure 29, is enabled, the machine-learning algorithm will take control of the drone. Using the onboard sensors, it automatically reads values such as the altitude and position and adjusts its position in order to meet the requirement. For obstacle avoidance, a LIDAR sensor is used to detect objects at a certain proximity. Moreover, the LiDAR sensor is also used for path construction, which with the aid of machine learning, is able to construct and follow the most optimal path to the designated powerline. Furthermore, the Raspberry Pi is able to detect an oncoming object by reading the data from the LIDAR sensor. It is also able to reconstruct the 3D map of the surrounds and in the event of a detection, the pi will automatically send a command to the motors to stop the drone, change directions, and hover away from said object until the sensor ceases to detect it. The entire flight process can be seen live through the live feed of the drone camera provided by the web application. Through machine learning, the drone will automatically detect the power grid using the onboard camera, and travel towards the designated destination. On arrival, the thermal sensor will scan the grid using a machine learning algorithm, and images are sent to the Raspberry Pi, which are then uploaded to a database for further processing (using LTE telemetry) to determine whether the power grid is faulty or not.

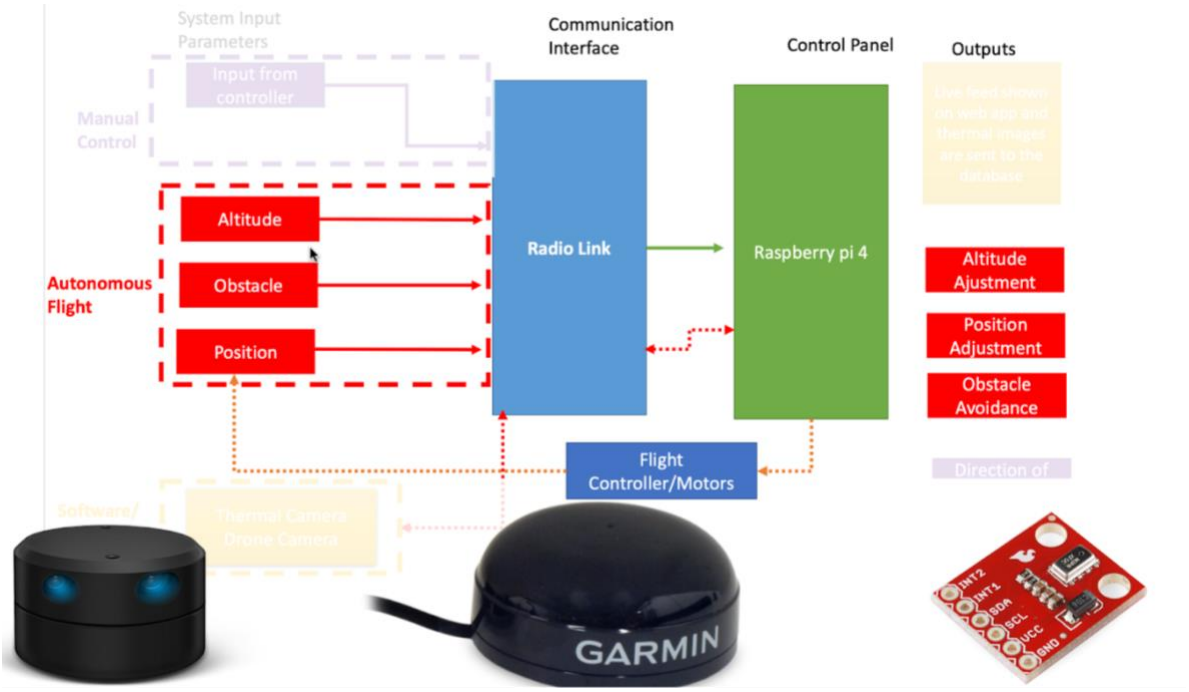


Figure 29: Drone automatic control

As discussed in the hardware components section of the present report, we will use a thermal sensor in order to inspect powerline. Once the drone has reached the powerline; either autonomously or manually, the inspection phase begins. The thermal sensor will start scanning and sending the data to the Raspberry Pi, which in its turn, will construct a thermal image of the powerline. Once the image is complete, it will be automatically compared with thousands of images found in the database. Our trained model will be able to compare the current scan with the existing images of both damaged and healthy powerlines and thus will be able to determine if the powerline is damaged and where the point of damage is located. Once the inspection phase is done, we are able to access the scanned thermal images using the web application or even older thermal images stored on the database using the web application as well. These different steps are summarized in Figure 30.

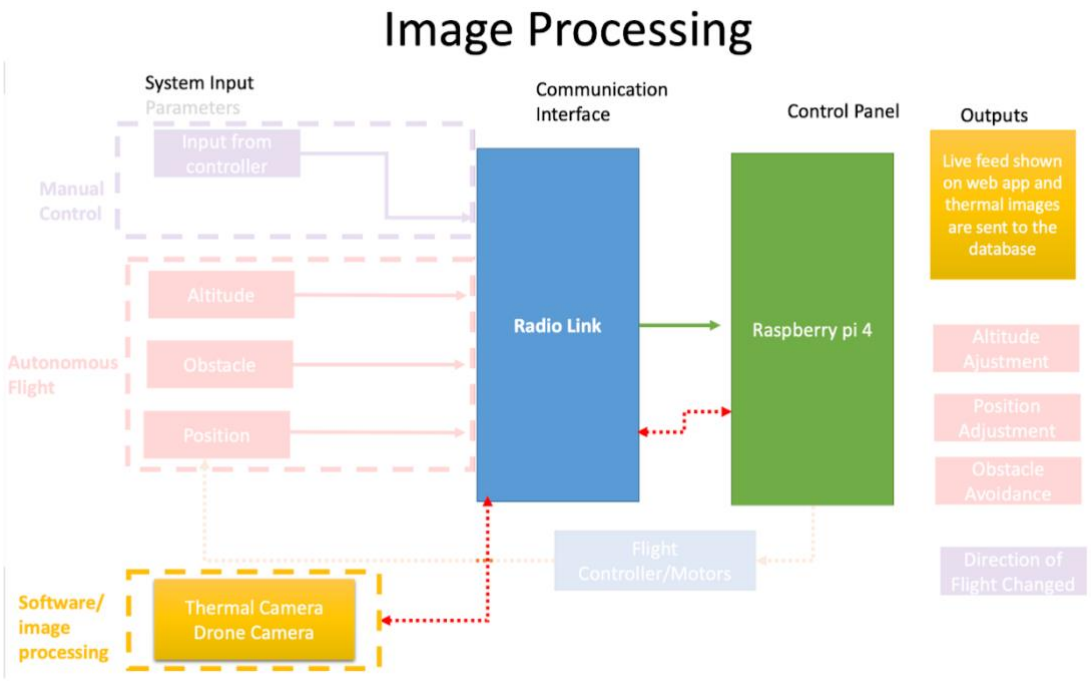


Figure 30: Drone image processing

12.1 DRONE FUNCTIONALITY

The drone functionalities, as discussed in the existing solution section include path construction and wire, insulation, and clamp inspection. Moreover, the LiDAR sensor will be the main component associated with the path construction, given that a ground vehicle is too expensive to setup and maintain. So having a highly accurate sensor that enables us to scan our surroundings is crucial for a successful autonomous flight mode. Moreover, for the insulation and wire inspections, the camera will be the main component used for inspection. The camera chosen can detect UV, which will allow us to detect any wire discharges through the "white blobs". Moreover, the images captured will allow us to inspect the insulators using our machine-learning algorithm. Finally, thermal sensors will be used for the clamp inspection given that inappropriate temperature readings are mainly caused by faulty/old clamp usage.

13 SOFTWARE IMPLEMENTATION

13.1 INSULATOR TRAINING MODEL FOR INSPECTION

Insulators are generally used in electrical equipment to help separate and support conductors, without allowing any current to pass through. In other words, powerline insulators should endure the horizontal and vertical loads and tension from the conductors and withstand the high voltages of these powerlines. Moreover, given that these insulators should operate in diverse and complex environments, it is not uncommon to experience shattering, string breakage, cracks, erosions, etc… However, not attending to these damaged insulators would drastically affect both safety and performance of powerlines. Hence, rapid and accurate detection of these defects is crucial to help reduce power grid losses and large-scale power outages [39].

13.1.1 Data Set

13.1.1.1 Image Collection

The first step in training the model was to gather and create a custom data set. Ideally, the dataset envisioned would have been composed of an already existing insulator data set merged with our own collected images. One of the main reasons this was important is accountability, since already existing data sets can include errors and biases. Therefore it was important to make sure that the data is well balanced and large enough to lower any additional inaccuracies. Another reason for wanting to collect our own images is because there are no existing datasets for Lebanon. So we wanted to contribute to this topic and maybe help start an initiative towards this kind of data collection. Moreover, though there were good intentions, powerlines in Lebanon were not easy to access since many of them are either located in residential areas or extremely far away from our current location. In addition, to be able to capture these images, DJI drones are typically used as they provide good images and quality. However, with the limited resources we currently have in Lebanon along with the economic crisis we are facing, buying, or renting a DJI drone was out of budget. As a result, we had to resort to already premade readily available datasets.

The dataset used for the model is the China powerline insulator dataset (CPLID) [https://github.com/InsulatorData/InsulatorDataSet], which is made publicly available for everyone to use. The dataset consists of 600 normal and 248 defective insulator images.

13.1.1.2 Data Labeling

The dataset should then be thoroughly looked at and annotated before the training of the detection model can take place. For this project, the labelImg [https://github.com/tzutalin/labelImg] labeling tool was used to annotate the category, defective or non-defective, by a bounding box and location of the insulator. LabelImg is a free open-source tool for labeling images graphically. The tool is written in Python and uses QT for its graphical interface. Moreover, the interface is easy to use, intuitive, and supports both VOC XML and YOLO text formats, making it a suitable option to use.

13.1.2 Training Data and Detection Model

13.1.2.1 YOLOv3

Object detection has become one of the most daunting challenges in computer vision. One of the most popular object detection algorithms is You Only Look Once Version 3 (YOLO V3) which is a convolution neural network, CNN, for performing object detection in real-time. Moreover, CNN algorithms are classifier-based systems, which help process input images or videos as structured arrays of data to help identify patterns. In other words, YOLOv3 shows the idea of residual network to improve object detection accuracy level such that it performs perfectly on detection speed since it uses a one-stage strategy. Figure 31 highlights a detailed architecture of this network where the feature extraction network, which is also known as Darknet-53, outputs a small-scale feature map smaller than the original images. YOLO V3 takes as an input images with 416 ×416 dimensions so that the size of the feature map extracted from the Darknet-53 becomes 13 × 13. This mechanism of decreasing the size of the images aims to detect objects of large size. After that, YOLO V3 network outputs a large-scale feature map by enlarging the small-scale feature map obtained from the previous step. To detect small objects the large-scale feature map contains information of previous layers in the network and other complex features from deeper layers. There are three scales of feature maps which are 8, 12 and 32 time smaller than the original image [39].

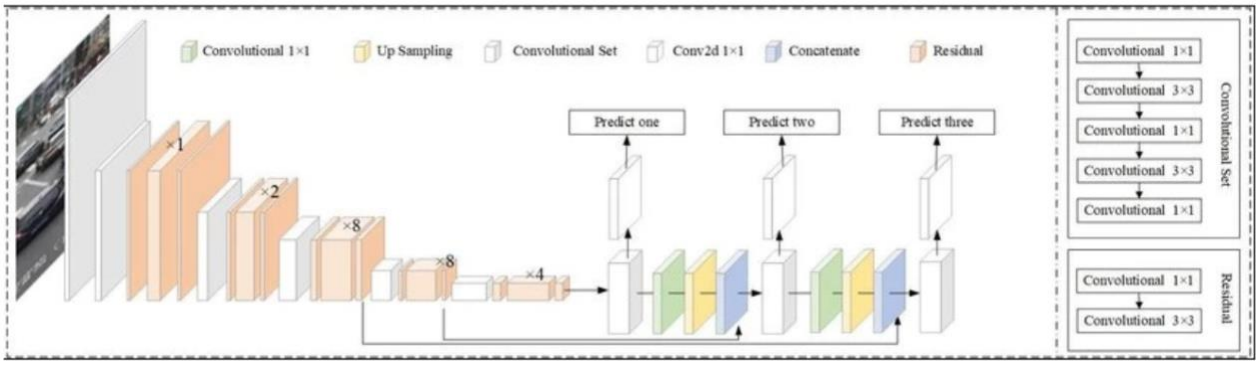


Figure 31: YOLOv3 Architecture [39]

13.1.2.2 YOLOv3 Compared to Newer and Older Versions

It is important to note that when it comes to older versions of YOLO, there are major differences in speed, accuracy, and specificity of classes. For instance, YOLOv2 uses the Darknet-19 which contains only 19 convolution layers while YOLOv3 uses Darknet-53 which has 53. This makes YOLOv3 much more powerful than Darknet-19 and more efficient compared to ResNET-101 and ResNet-152. Figure 32 shows a graphical representation of YOLOv3 compared to other algorithms. (https://arxiv.org/pdf/1804.02767.pdf)

However, while YOLOv3 has been proved to be overall better than other popular algorithms such as the ResNet-101 and ResNet-152 and older versions such as YOLOv1 and v2, there are newer and improved versions such as the YOLOv4 and v5 which must be considered. For instance, YOLOv4 uses the Darknet Cross Stage Partial Network, CSPDarknet53, and YOLOv5 unlike other releases uses PyTorch instead of Darknet. As a result, YOLOv5 ranks the highest in terms accuracy followed by YOLOv4 then YOLOv3, while YOLOv3 ranks the highest in terms of detection speed followed by YOLOv4 and v5, respectively. The higher accuracy and lower speed of YOLOv5 compared to YOLOv4 is due to the fact that YOLOv5 uses auto learning bounding boxes which improves the overall accuracy but increases complexity. The better accuracy of both YOLOv4 and v5 compared to v3 is due to Darnet53's struggle in detecting small objects (https://doi.org/10.3390/s22020464). As a result, given that in our model we will not be detecting small objects and we will be performing a real-time detection, the faster YOLOv3 is a better option compared to the slower YOLOv4 and v5 [39][40].

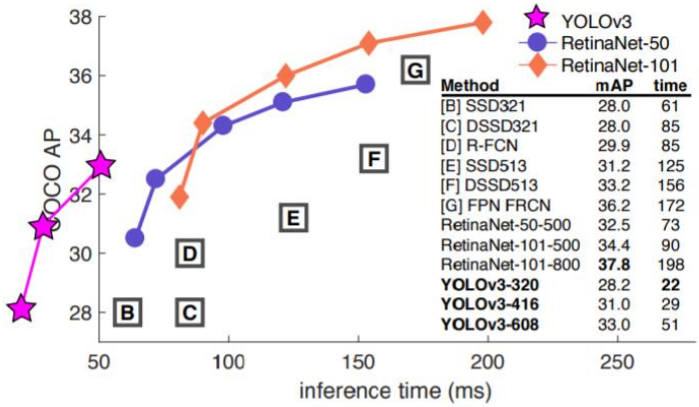


Figure 32: YOLOv3 vs. ResNet-101 vs. ResNet-152 [40]

13.1.2.3 Training Model

13.1.2.3.1 Environment Setup

To be able to start our training, we first created a Gmail account dedicated for our project, by which the training would take place through Google Collab. Once the notebook has been created, we were able to start coding our training model. First, we had to mount our google drive files to Collab, since our labeled training images had been previously uploaded in a zip folder to google drive. Second, we had to clone Darknet into Collab, and then we had to configure our make file such that our GPU and OpenCV are enabled.

13.1.2.3.2 Configuration of yolov3.cfg File

In this step, we had to modify the yolov3.cfg file depending on the number of classes we have in our model. First, we set our batch number to 64 to use 64 images for every single training step. Second, we changed our subdivisions to 16, meaning that our batch will be divided by 16 to help us lower RAM and GPU requirements. Third, we changed the maximum number of batches to at least 2000 times our number of classes, which in our case is 2 classes, thus making us set the max-batched to 4000. Fourth, we had to change the number of filters based on the number of classes we have. To calculate the filter, there is an equation that we must follow which is

$$filters = (classes + 5) * 3$$

Therefore, given that we need to train 2 classes, we had to set our filters to 21. Fifth, we will need to change the number of objects we want to detect, which is the number of classes we will have. As a result, Figure 33 shows a screenshot of the corresponding code from google Collab.

```
# Change lines in yolov3.cfg file
!sed -i 's/batch=1/batch=64/' cfg/yolov3_training.cfg
!sed -i 's/subdivisions=1/subdivisions=16/' cfg/yolov3_training.cfg
!sed -i 's/max_batches = 500200/max_batches = 4000/' cfg/yolov3_training.cfg
!sed -i '610 s@classes=80@classes=2@' cfg/yolov3_training.cfg
!sed -i '696 s@classes=80@classes=2@' cfg/yolov3_training.cfg
!sed -i '783 s@classes=80@classes=2@' cfg/yolov3_training.cfg
!sed -i '603 s@filters=255@filters=21@' cfg/yolov3_training.cfg
!sed -i '689 s@filters=255@filters=21@' cfg/yolov3_training.cfg
!sed -i '776 s@filters=255@filters=21@' cfg/yolov3_training.cfg
```

Figure 33: Configuration of yolov3.cfg File

13.1.2.3.3 Creating the obj.class and obj.data Files

When using pre-trained models, the obj.class file (refer to Figure 34) is commonly known as the coco.names located in the data folder, and this file is used when displaying the name of our detected object in the bounding box. Afterwards, we will need to create our obj.data file which contains the number of classes we wish to train, the directory path (location) of our train.txt, test.txt files, and obj.names files, and the path to save the trained yolo weights.

```
[ ] !echo -e 'defective_insulator\nnon-defective_insulator' > data/obj.names

[ ] !echo -e 'classes= 2\ntrain  = data/train.txt\nvalid  = data/test.txt\nnames = data/obj.names\nbackup = /mydrive/yolov3' > data/obj.data

[ ] !cp cfg/yolov3_training.cfg /mydrive/yolov3/yolov3_testing.cfg
    !cp data/obj.names /mydrive/yolov3/classes.txt
```

Figure 34: obj.class and obj.data Files Code

13.1.2.3.4 Place and Unzip Images

Afterwards, we will need to create a folder named “obj”, which will be storing our images after we unzip them. Afterwards, we use the python glob module to dynamically get the names of the images which must be stored in the train.txt file instead of manually writing them (refer to Figure 35).

```
[ ] !mkdir data/obj
    !unzip /mydrive/yolov3/images.zip -d data/obj

    import glob
    images_list = glob.glob("data/obj/*.jpg")
    with open("data/train.txt", "w") as f:
        f.write("\n".join(images_list))
```

Figure 35: Image unzip and train.txt File Code

13.1.2.3.5 Training the Insulator Detection Model

Finally, we will need to download the per-trained weights for the convolution layers and place them in the darknet directory, and then we can run our training command. The training command consists of the obj.data file which is used to identify our training information, yolov3_training.cfg,

which is used to identify our training network configurations, and the last parameter is the darknet53.conv.74 for identifying the per-trained network. Finally, once we get our first "weights" file, we re-trained our model using that file in order to increase our accuracy (refer to Figure 36).

```
[ ]  !./darknet detector train data/obj.data cfg/yolov3_training.cfg darknet53.conv.74 -dont_show
     # Uncomment below and comment above to re-start your training from last saved weights
     !./darknet detector train data/obj.data cfg/yolov3_training.cfg /mydrive/yolov3/yolov3_training_last.weights -dont_show
```

Figure 36: Insulator Training Model

13.1.3 Sample Testing

Figures 37 and 38 show the results obtained for both a defective and non-defective insulators.

Figure 37: Defective Insulator Test

Figure 38: Non-Defective Insulator Test

13.1.4 Powerline Testing

Sample results for powerline inspection are provided for completeness in Figures 39 and 40.

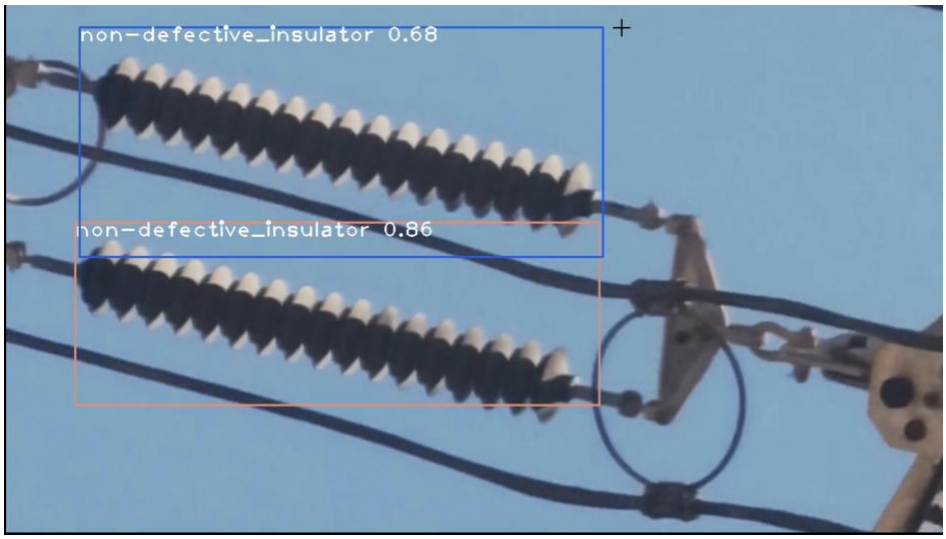


Figure 39: Powerline Inspection Capture 1

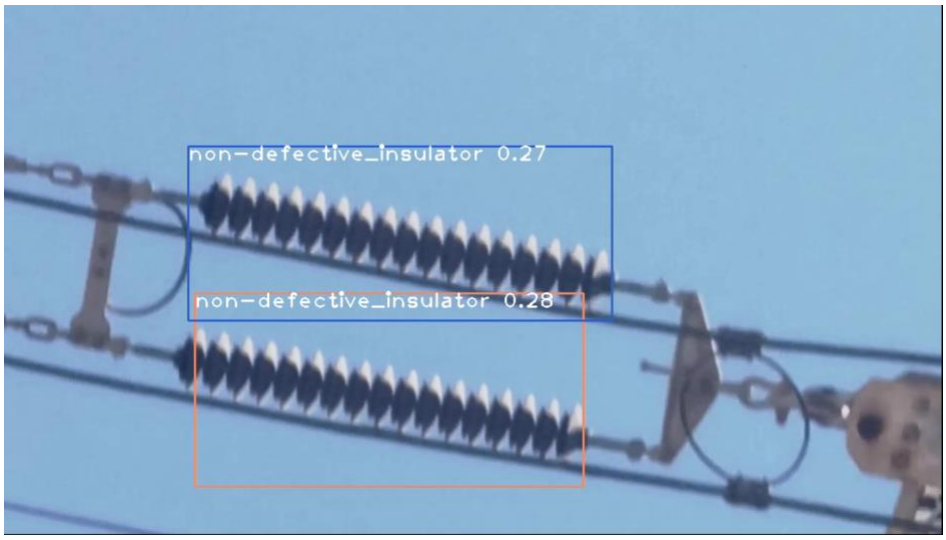


Figure 40: Powerline Inspection Capture 2

13.2 THERMAL INSPECTION

Thermal inspection of the powerline, while is not technically considered part of the inspection process, but it is an important feature to have. Sometimes powerlines may be heating some components and physical inspections would not detect the issue. As a result, integrating the

thermal feature would provide us an insight on whether there are underlying problems that inspectors need to address.

13.3 CLAMP INSPECTION

13.3.1 Data Set

The first step to training the model was to gather and create a custom data set. Similarly, to insulators, the dataset envisioned would have been composed of an already existing insulator data set merged with our own collected images. Due to cost-related impediment, it was clear that gathering our own dataset was not feasible. As such, resorting to pre-existing datasets was our solution. However, unlike insulators, there are no public datasets available, but instead all datasets related to clamps are restricted by the government and access to them would require us to contact the government of that specific set. Although we tried reaching to them, we received no reply, and with no data set at hand, we had no option but to disregard the clamps during the inspection process.

14 HARDWARE IMPLEMENTATION

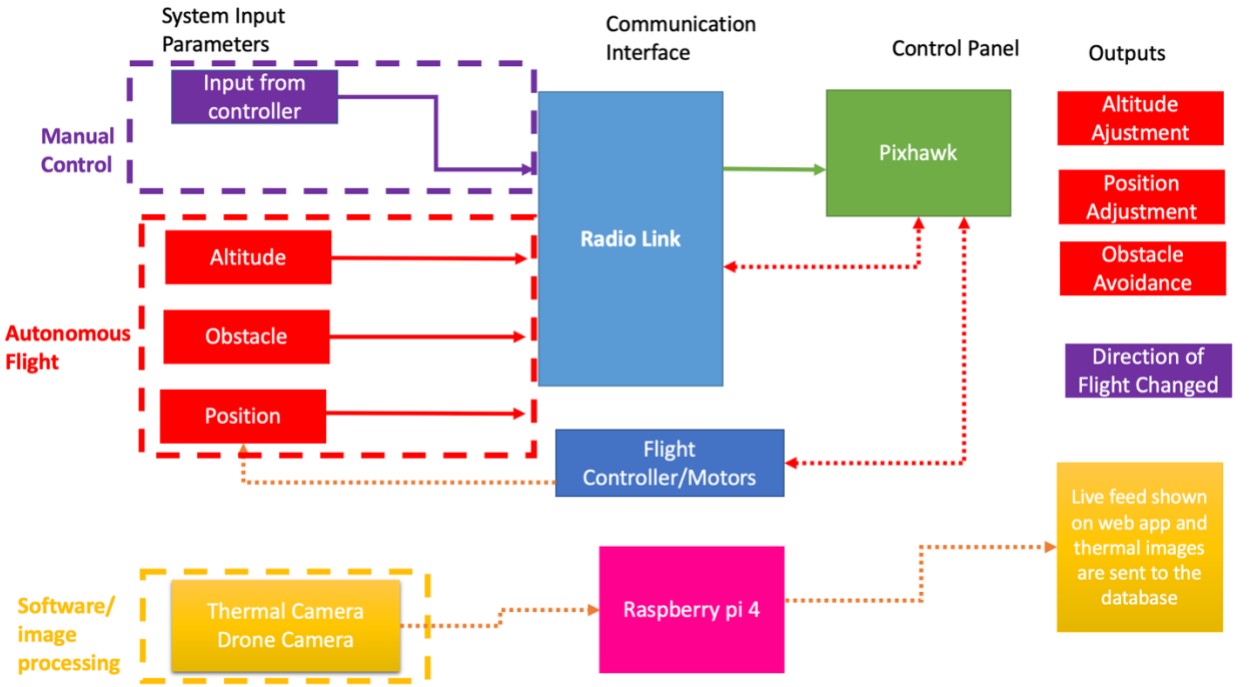


Figure 41: Overall System Architecture

The architecture of our proposed solution can be seen in Figure 41. Our design involves a combination of software and hardware interfaced together such that the desired functionalities are met. These components include motors, batteries, electronics speed controllers and an RC controller. This is coupled with sensors which, enable us to achieve the desired level of control that would allow the drone to fly correctly.

Moreover, we have used two controllers in our design: a Raspberry Pi 4 and the Pixhawk flight controller. The Pixhawk was responsible for the drone flight and flight modes. It interfaces with the motors and sensors, reads from an RC controller, and makes sure that the drone is balanced and not tilted. Using its built-in compass and sensors (gyroscope, accelerometer, etc..), the Pixhawk makes sure that the drone is flying in a stabilized form. Moreover, the Pixhawk has the capability to connect to a variety of obstacle detection sensors such our LiDAR sensor, which offers the capability of autonomous flight and obstacle avoidance. A GPS is also connected to the Pixhawk and offers multiple functionalities. Furthermore, the Pixhawk connects to the RC remote and enables drone control.

On the other hand, the Raspberry Pi is our single board computer used for the inspection part of the project. First, we are using the MakerFocus Raspberry Pi camera to live stream a live feed from the drone and take photos of the powerline insulators. These images will be later ran into an artificial neural network which will determine the status of these components. Furthermore, the MLX90640 is the thermal camera that we have adopted to scan components as well. The Raspberry Pi will generate a thermal image as well as record the temperature of the components, which are saved onto the database.

In the following section, we will thoroughly go into the individual hardware and software parts of the project highlighting the decisions made, problems faced, and solutions developed to fully build this project.

14.1 HARDWARE COMPONENTS

14.1.1 Brushless DC Motors

For our DC motors, we have opted for the A2212 brushless DC motors 2700KV as each motor has the capability of lifting about 1KG. We went for a quadcopter design; thus, we have used four brushless motors. 2700KV means that since our battery delivers 12V DC our motor can rotate at a max speed of 12x2700= 32400. We have taken account in our body design and motor number choices the weight of the frame as well as the weight of the components to be carried by the drone notably: the battery and LiDAR sensor.

14.1.2 Electronic Speed Controller

Electronic speed controllers (ESC) take an input signal from the Pixhawk and channels the appropriate current to the motor. The more current passes to the motor, the faster the rotation is. Since our motors are three-phased motors, each phase is soldered onto a slot on the ESC and since some motors need to spin in a clockwise direction and others in a counterclockwise direction; the motor rotation can be changed simply by switching the motor phases on the ESC.

14.1.3 Battery

The drone required a high-capacity battery to be able to accommodate all the weight that was being thrown into the drone. Additionally, the drone was required to fly for about 10 minutes to

be able to perform several inspections before a recharge is required. As a consequence, we used a 5400mAh 3s battery with a discharge rate of 35C, which is more than enough since the drone will be in loiter mode most of the time.

14.1.4 RC Controller

When it comes to controlling the drone, the FlySky FS-i6 device (refer to Figure 42) was the best option to use since it was relatively cheaper than other RCs. It is a great entry-level 6-channel telemetry 2.4ghz computer transmitter that uses the solid and reliable Automatic Frequency Hopping Digital System (AFHDS) spread spectrum technology.

Figure 42: Flysky FS-I6[34]

14.1.5 Pixhawk

The Pixhawk connects to the Mission Planner, which is the software used to program the drone, configure the RC, setup the different flight modes. The Pixhawk also comes with a radio receiver and transmitter, which represents the means used to communicate with the drone.

Radio telemetry is also used to control the drone using the RC remote controller. A GPS is connected to the Pixhawk and allows us to: 1) have a precise location of the drone at the ground station, and 2) send the drone to a specific location from the ground station.

14.1.6 Raspberry Pi

The Raspberry Pi was connected to the Makerhawk Battery Module shown in Figure 43 which powers up the Raspberry Pi while it is on the drone using 3.7V lithium-ion batteries.

Figure 43: Makerhawk Battery Module [38]

14.1.6.1 Live Feed

The first feature of the Raspberry Pi that we have implemented is the ability to live stream the drone live footage while it is in the air. This is achieved using the MakerFocus camera, which has a resolution up to 3280 x 2464 or 4K. Having a high a resolution camera is crucial since we are inspecting high voltage powerlines with high magnetic fields. So, it is essential to keep the drone as far away as possible from the powerline.

14.1.6.1.1 Software Implementation

To have a live feed from the drone (refer to Figure 44), we have implemented a script that opens the Raspberry Pi camera and live streams the camera onto a local host, which can then be accessed using the Raspberry Pi. This live feed is always accessible from the Web Application which will be discussed later on in the report.

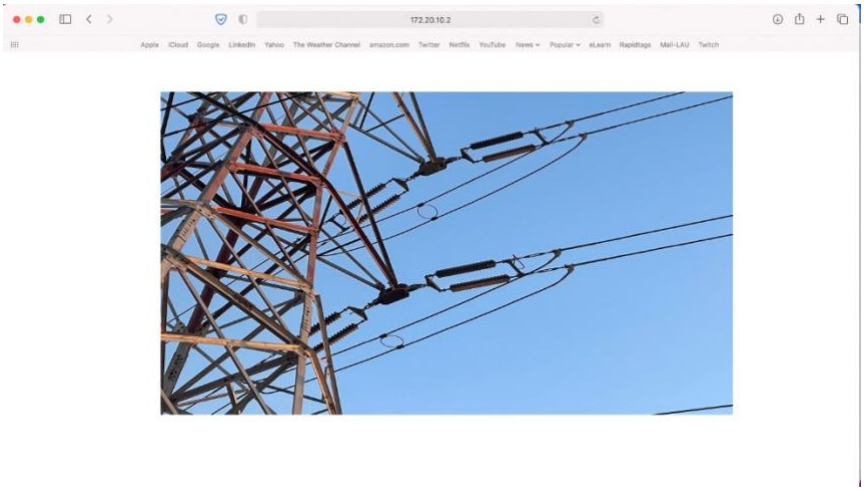


Figure 44: Raspberry Pi Live Feed Screenshot

14.1.7 Image Capture

Another use of the MakerFocus is capturing images and live feed for the insulator inspection. These images were stored on an AWS S3 bucket, which was then passed on into an artificial neural network (ANN) to determine whether the insulator is defective or non-defective.

14.1.8 Thermal Imaging

We used the MLX90640 thermal sensor, since it provides a 32x32 thermal image as well as the ability to read the temperature of the scanned component. The component's temperature is the value sought after as it gives an indication that the component is overheated and thus is most likely damaged.

14.1.8.1 Thermal Sensor

We are using the MLX90640 thermal sensor (refer to Figure 45). We are reading through I2C communication temperature. Then through Matplot Lib we are plotting on a 32x32 plot each pixel by taking the temperature at each pixel, comparing it with a scale in order to choose a color then move onto the next pixel. This loop continues until we have reached the end of the image. Then the image is returned and then saved on the S3 bucket.

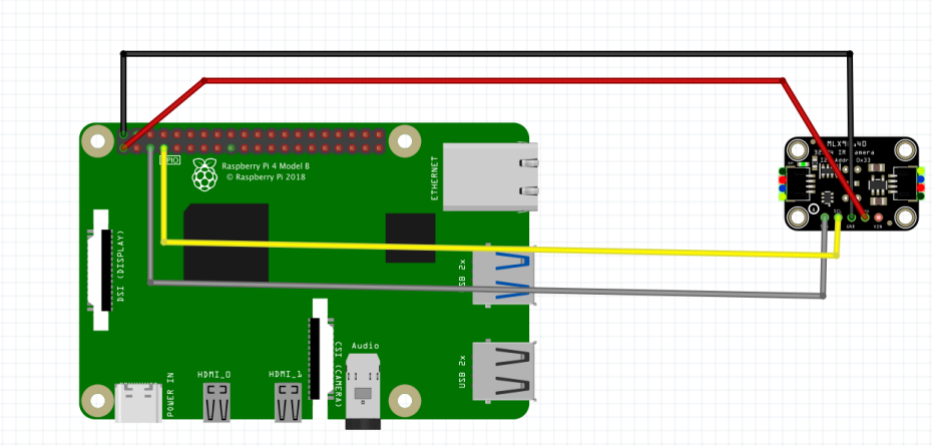


Figure 45: MLX90640 Connection

14.2 MISSIONS PLANNER

Mission planner was the software used to calibrate and setup the Pixhawk. There are several steps to take to properly setup a Pixhawk for its first test flight.

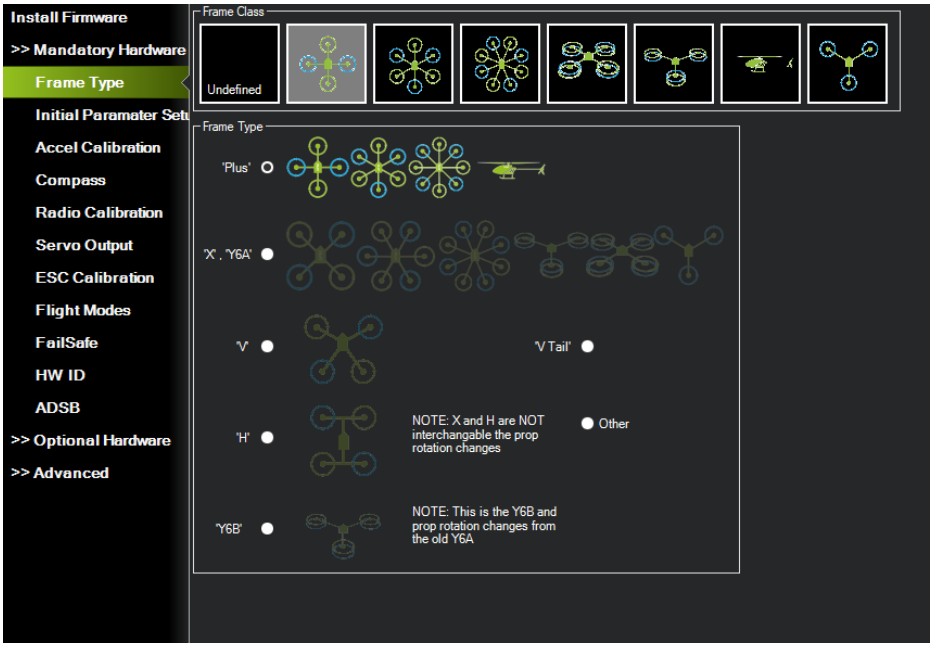


Figure 46: Mission Planner Frame Type

This first one if to set the frame type. Since the Pixhawk can be used to control all sorts of drone types, the user has to specify the frame to be employed. In the case of this project, the quad plus drone frame was selected (refer to Figure 46).

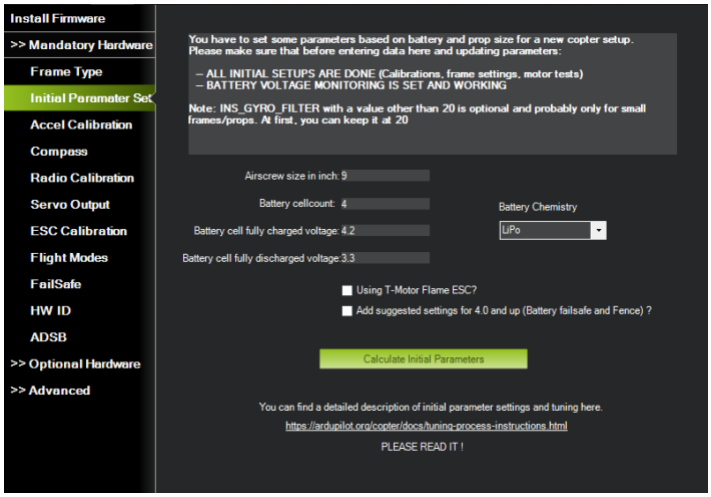


Figure 47: Mission Planner Initial Parameter Setup

Mission Planner simplifies the drone setup, in the sense that it takes the basic parameters that are entered by the users such as the propeller size, battery cell count, cell full and empty charge voltages (refer to Figure 47). These parameters are then used by the Pixhawk for self-calibration refer to Figure 48). Some parameters that are set by this option are essential for the sensors used by the controller such as the accelerometer, gyroscope, battery sensor, and other parameters that will result in a smoother flight experience.

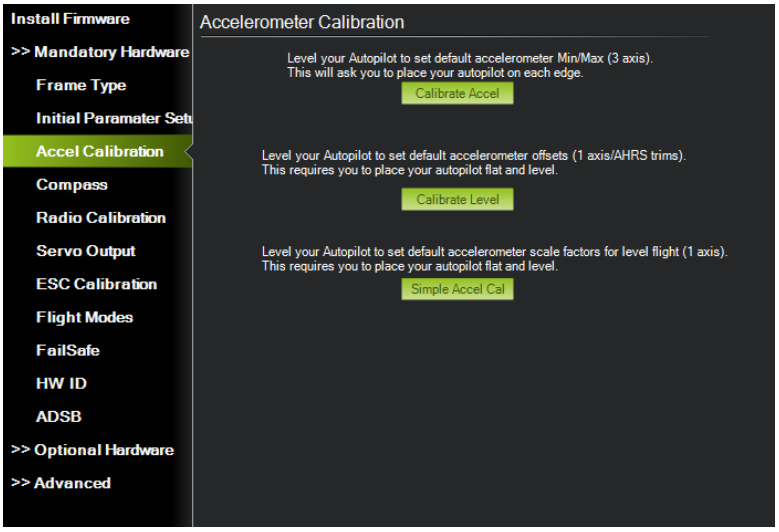


Figure 48: Mission Planner Accelerometer Calibration

At this point, the Pixhawk calibrates the accelerometer. The second and third calibrations will only require the drone to be placed on a leveled surface, but the first one will require the user to carry the drone (props/motors disconnected) and perform certain tasks, such as rotating the drone

to the left, the right, nose down/up, and its back. This will give the Pixhawk an accurate reading of its orientation.

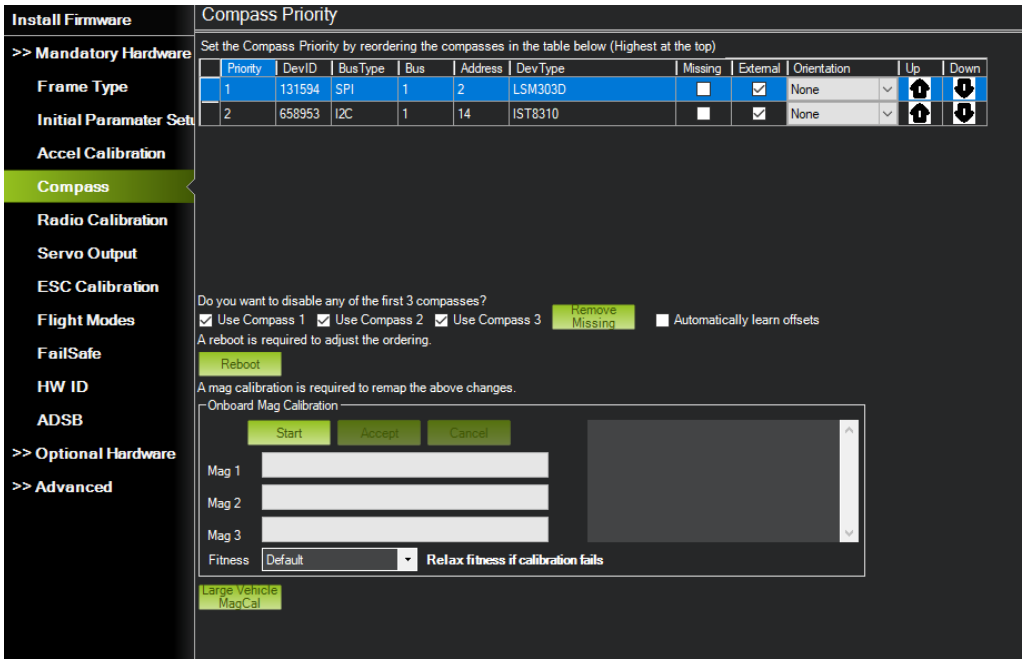


Figure 49: Mission Planner Compass Calibration

The table portrayed in Figure 49 represents the number of connected compasses. In most cases, there is an internal compass that is integrated inside the Pixhawk, and an external one that is located in the GPS module of the Pixhawk. It is crucial that both the GPS module and Pixhawk are aligned; otherwise, the controller will trigger a compass inconsistent failsafe and will prevent the drone form arming (unless failsafe is manually disabled). To calibrate the compass, the user must press the “start” button, and start rotating the drone in all possible directions. The latter will fill the ‘Mag 1” and “Mag 2” bars. Once the process is completed, the controller will be able to retrieve the offset values.

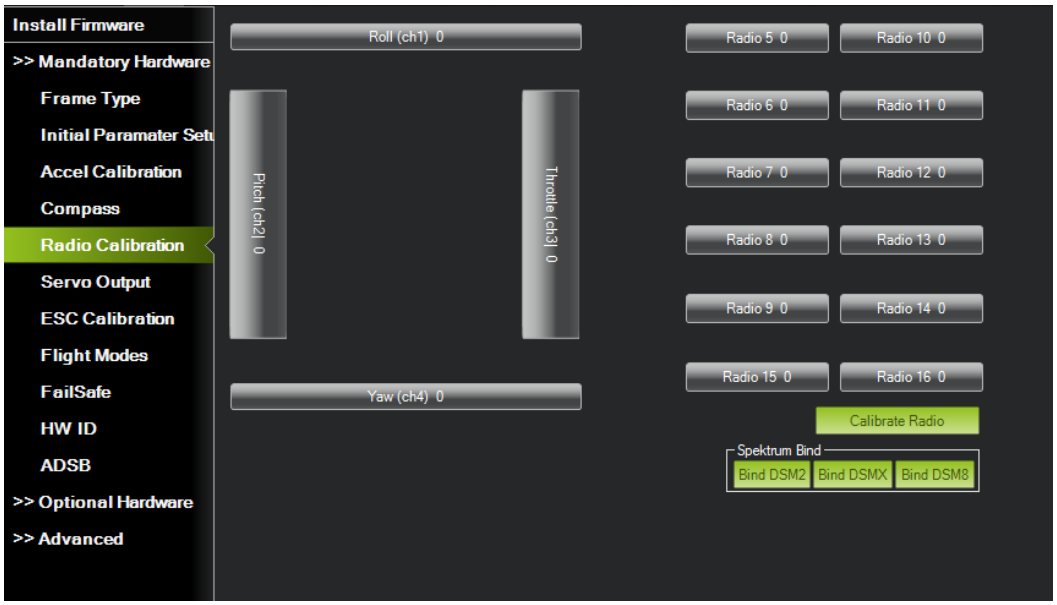


Figure 50: Mission Planner Radio Calibration

At this stage, the RC receiver is well calibrated, in which case each channel will control a certain drone parameter (refer to Figure 50).

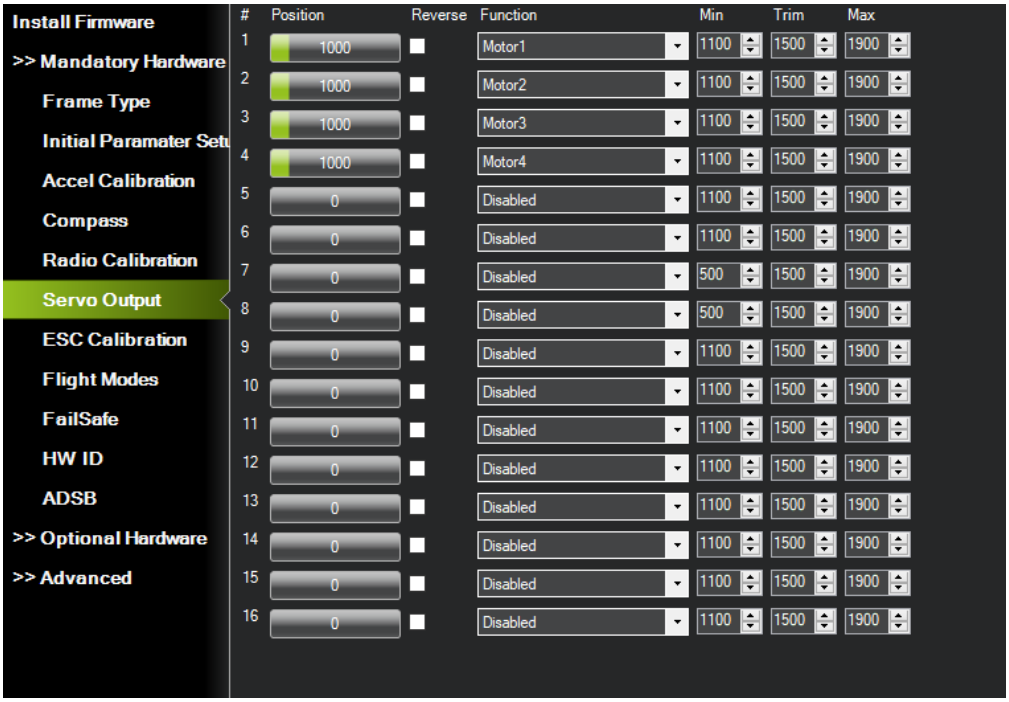


Figure 51: Mission Planner Servo Output

The following option will let the user set the function of each data pin that are used in the Pixhawk (Main Out). In this case, the motors were selected and assigned at their corresponding order (refer to Figure 51).

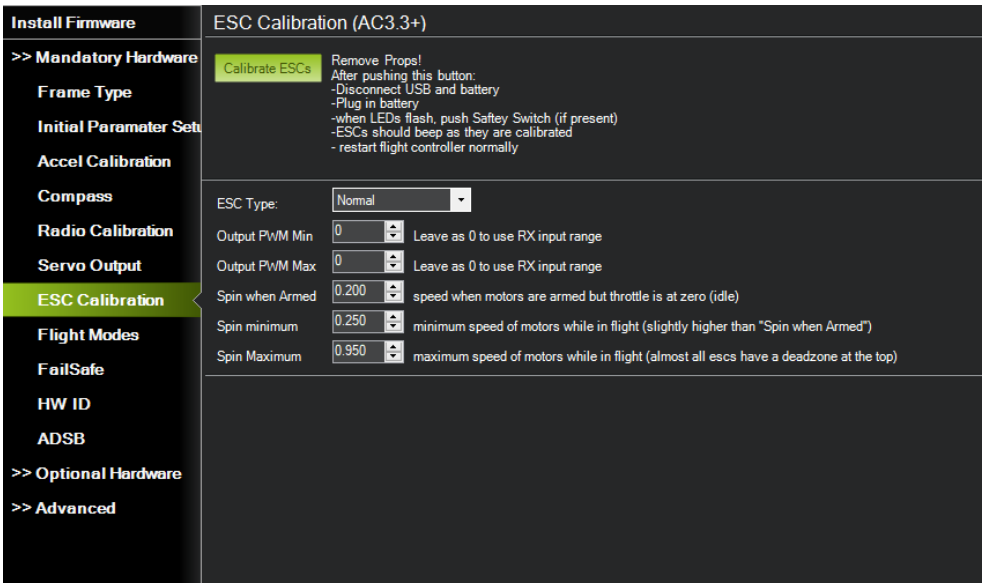


Figure 52: Mission Planner ESC Calibration

Calibrating the ESCs (refer to Figure 52) is the most important part of the process, since a miscalibration will result in major problems, such as the drone motors not spinning at all, some motors spinning faster than other motors, or motors spinning too fast. The ESC must also be specified, which can be found on the ESC's datasheet.

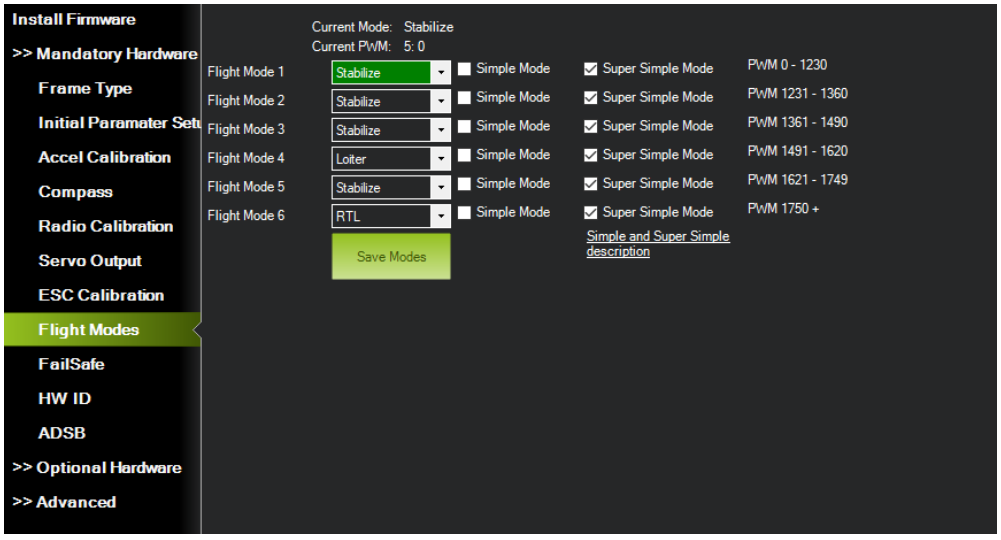


Figure 53: Mission Planner Fight Modes

Setting up flight modes entirely depends on the purpose of the drone (refer to Figure 53). The user has a selection of several dozens of flight modes. The most used modes are:

- Stabilize, which will keep the drone flying at a stable pace,
- Loiter mode, which will keep the drone hovering at the same place
- RTL (Return to Launch), which will land the drone in the same location it first took off (GPS required)

All the options can be toggled via the switches on the RC remote.

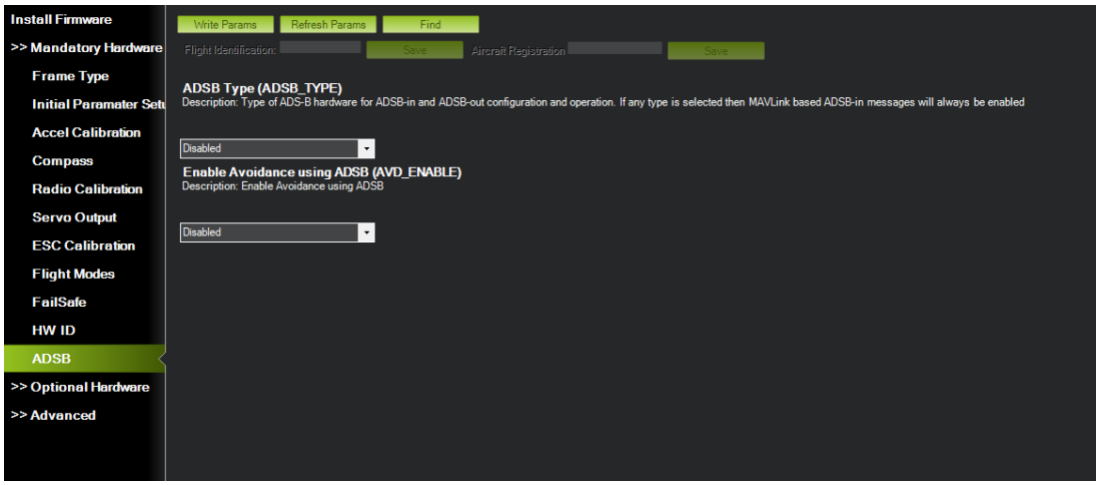


Figure 54: Mission Planner ADSB

The option given in Figure 54 enables object detection for the Pixhawk using the connected LiDAR sensor, and consequently enables object avoidance.

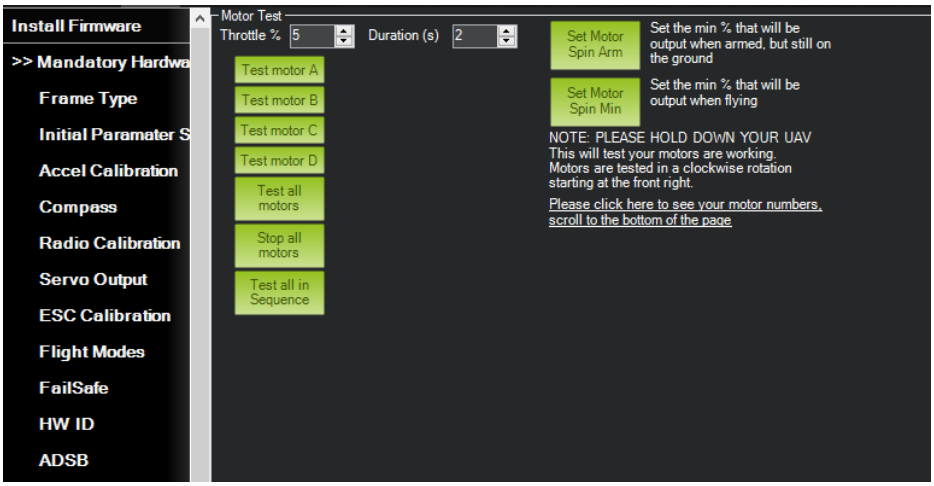


Figure 55: Mission Planner Motor Test

The optional setup depicted in Figure 55 is used to test whether the motors are working properly or not. Each motor can be tested individually while adjusting the throttle level, along with the duration of the test.

15 WEB APPLICATION

Our system has a web application which expands and broadens the user's interactions with the design. It combines most of the software into one graphical user interface. To be able to use the application, the users have to login first using their own given credentials so that only the authorized person has access to the recordings and data. An authorized user can then view a live feed of the drone's camera to have a direct visual from the drone's perspective. This enables the user to detect any defects directly in case it is apparent.

The whole website was created using HyperText Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript. For the authentication purposes, we used Firebase which provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to the application. To achieve this, it supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook, and so on. In our website, we went with the password and email type of authentication. In the below screenshot, the login page is shown for the user to log into the application.

Figure 56: Website Login Page

Also, since only the authenticated users can log into the account, we have created a contact form so that if anyone has any problem, he/she can navigate to the "Contact us" form shown below. He/she then can send an email enclosing their inquiry.

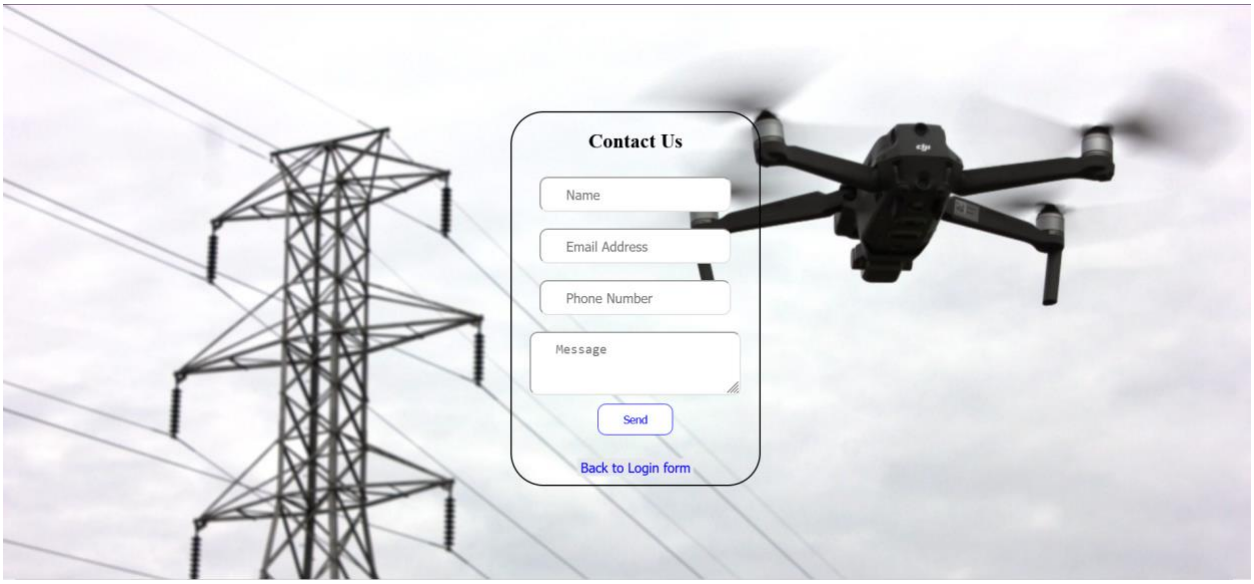


Figure 57: Website Contact us Page

Now, after logging into the account, the user has the choice to either view a live stream from the drone or view the previous images/videos saved in the online s3 bucket. The general view of the website with its navigation bar elements is shown in the sequence of figures below.

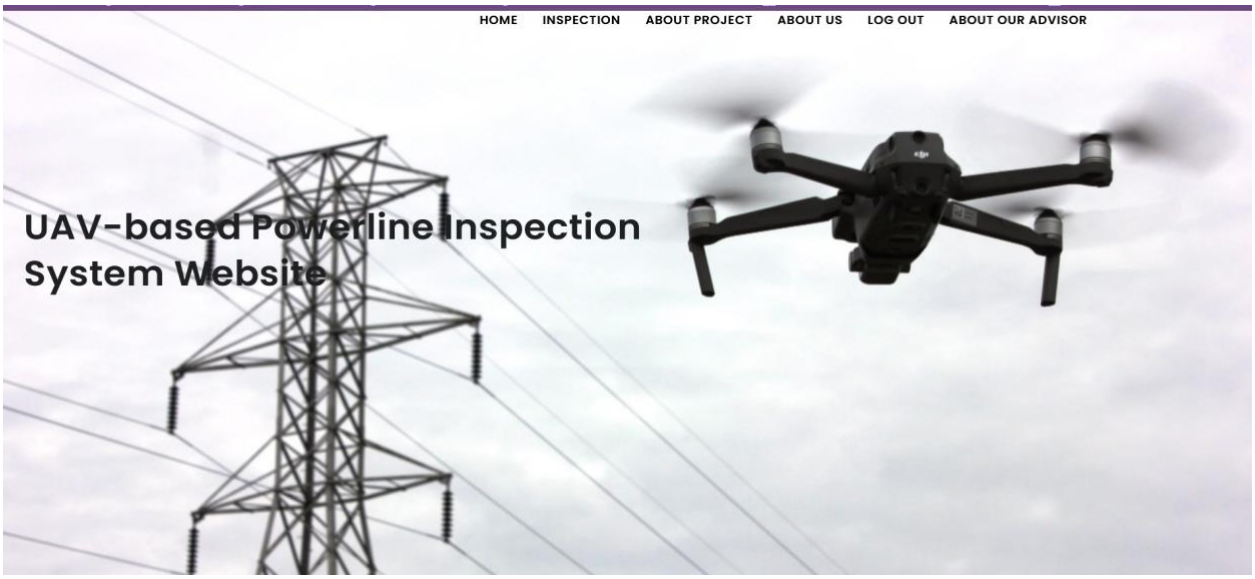


Figure 58: Website Homepage

Figure 59: Navigation Bar Inspection Elements

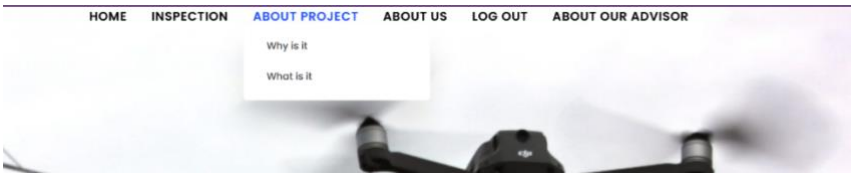


Figure 60: Navigation Bar About Project Elements

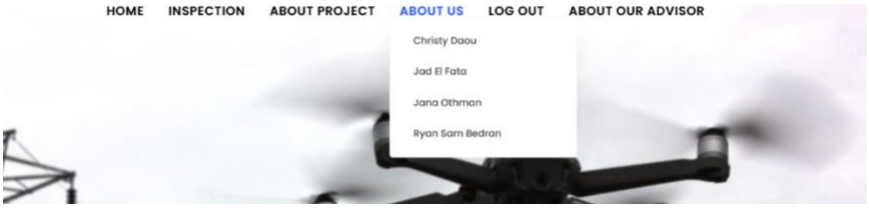


Figure 61: Navigation Bar About Us Elements

For checking the data previously saved in the s3 bucket, the user can use the interface depicted in the Figure below.

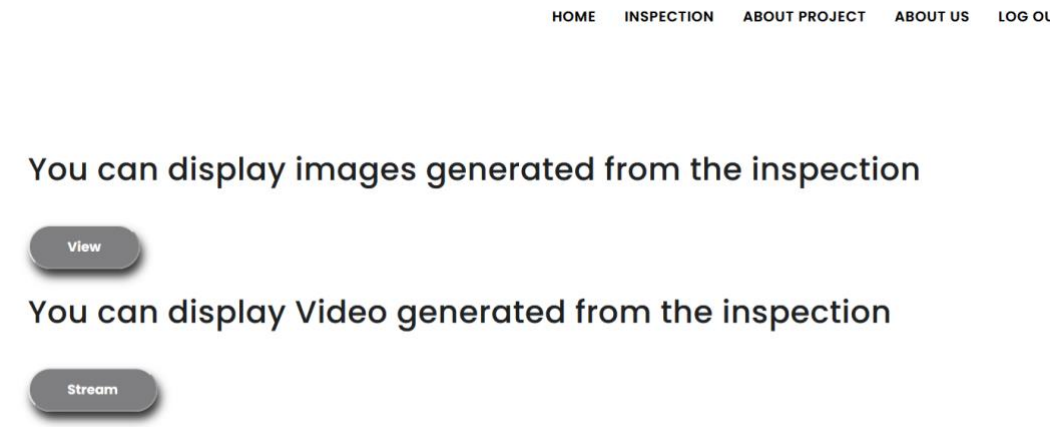


Figure 62: View Saved Data Page

15.1 CONNECTION

S3 bucket is a service provided by Amazon Web Services (AWS) for storing objects in a public cloud. The interaction between the customer and the Amazon s3 bucket happens through the AWS Management Console shown in Figure 63. The benefits of this storage mechanism are that there is no limit on the number of the objects that can be stored inside the s3 bucket. Moreover, the s3 bucket users can use the versioning feature of this service to track the different versions of the objects in the bucket. To upload objects to our bucket, we open an S3 socket from the inspection code, which automatically uploads the metadata to the corresponding S3 bucket. To access these objects, the access point was changed to be a public one so that each object can be accessible through a Uniform Resource Locator (URL).

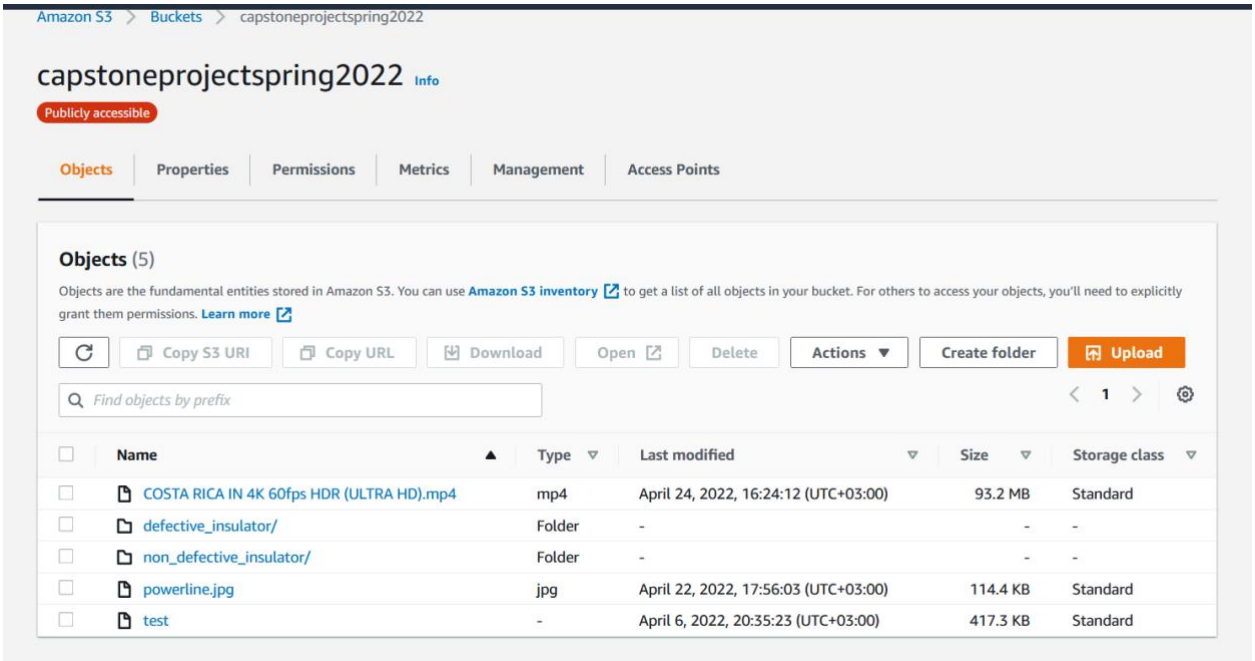


Figure 63: S3 Bucket View

16 CONCLUSION

This report introduced our solution for the powerline inspection problem. It provided some of the most recent solutions regarding the use of a drone for powerline inspection. In addition, we also covered some of the existing models along with their functions to get a better understanding of the required improvements. We also covered many sensors, controllers, and heuristics that are relevant to our project and decided on which one to use based on several factors, which include availability and cost. We concluded with our final architecture, which allows us to meet all the desired drone functionalities. Our powerline inspection is unique in providing almost double the features currently supported by existing drones while maintaining a relatively low construction cost.

17 REFERENCES

[1]H. Guan et al., "UAV-lidar aids automatic intelligent powerline inspection", International Journal of Electrical Power & Energy Systems, vol. 130, p. 106987, 2021. Available: 10.1016/j.ijepes.2021.106987.

[2]M. Ahmed, J. Mohanta and M. Zafar, "Development of smart quadcopter for autonomous overhead power transmission line inspections", Materials Today: Proceedings, 2021. Available: https://doi.org/10.1016/j.matpr.2021.05.271.

[3]J. Souter, "Why Drones are Essential for Power Line Inspection | Landpoint", Landpoint, 2022. [Online]. Available: https://www.landpoint.net/drone-power-line-inspection/.

[4]Y. Liu, J. Shi, Z. Liu, J. Huang and T. Zhou, "Two-Layer Routing for High-Voltage Powerline Inspection by Cooperated Ground Vehicle and Drone", Energies, vol. 12, no. 7, p. 1385, 2019. Available: 10.3390/en12071385.

[5]O. Kähler, S. Hochstöger, G. Kemper and J. Birchbauer, "AUTOMATING POWERLINE INSPECTION: A NOVEL MULTISENSOR SYSTEM FOR DATA ANALYSIS USING DEEP LEARNING", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. -4-2020, pp. 747-754, 2020. Available: https://doi.org/10.5194/isprs-archives-XLIII-B4-2020-747-2020.

[6]Z. Bayasgalan, A. Sukhbaatar, U. Tudevdagva and W. Hardt, "Top Inspection and Monitoring of HV Power Line Towers Damage by UAV", 2021 International Conference on Technology and Policy in Energy and Electric Power (ICT-PEP), 2021. Available: 10.1109/ict-pep53949.2021.9601091.

[7]A. Dietsche, G. Cioffi, J. Hidalgo-Carrio and D. Scaramuzza, "Powerline Tracking with Event Cameras", 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021. Available: 10.1109/iros51168.2021.9636824.

[8] R. S. Dimitrova, M. Gehrig, D. Brescianini, and D. Scaramuzza,“Towards low-latency high-bandwidth control of quadrotors using event cameras,” in IEEE Int. Conf. Robot. Autom. (ICRA). IEEE, 2020, pp. 4294–4300.

[9]]G. Komar, O. Pischler, U. Schichler, and R.L. Vieriu, 2019. Performance of UV and IR Sensors for Inspections of Power Equipment. In Proceedings of the Nordic Insulation Symposium, No. 26, 82-87.

[10] C. L. Gentil, F. Tschopp, I. Alzugaray, T. Vidal-Calleja, R. Siegwart, and J. Nieto, “Idol: A framework for imu-dvs odometry using lines,” in IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS). IEEE, 2020.

[11]“Pixhawk 4,” Pixhawk 4 | PX4 User Guide. [Online]. Available: https://docs.px4.io/master/en/flight_controller/pixhawk4.html.

[12] (GenerationRobots, n.d.)

[13] (Slamtec RPLIDAR A1, n.d.)

[14] (Amazon, n.d.)

[15] (Sturm, n.d.)

[16] (Barometric Pressure Sensor, n.d.)

[17] (BerryGPS, n.d.)

[18] (D6T MEMS Thermal Sensors, n.d.)

[19]D6T Thermal Sensors. OMRON, 2022.

[20] C. Bergquist, “How to make a DIY 4G drone for Bvlos Control,” Drone Dojo, 30-Aug-2021. [Online]. Available: https://dojofordrones.com/4g-drone/.

[21] “Sik Telemetry radio,” SiK Telemetry Radio - Copter documentation. [Online]. Available: https://ardupilot.org/copter/docs/common-sik-telemetry-radio.html.

[22] “UAVcast 3G/4G cellular telemetry¶,” UAVcast 3G/4G Cellular Telemetry - Copter documentation. [Online]. Available: https://ardupilot.org/copter/docs/common-uavcast-telemetry.html.

[23] “# GPS & Compass,” GPS & Compass | PX4 User Guide. [Online]. Available: https://docs.px4.io/master/en/gps_compass/.

[24] (IMX477-AACK, n.d.)

[25] (PixHawk 4, n.d.)

[26] B. Szyk, “Drone flight time calculator,” Omni Calculator, 07-Jun-2018. [Online]. Available: https://www.omnicalculator.com/other/drone-flight-time#drone-flight-time-formula.

[27] “Batteriesinaflash.com, inc..,” BatteriesInAFlash.com. [Online]. Available: https://www.batteriesinaflash.com/c-rating-calculator.

[28]"Lumenier 4000mAh 6S 120c CineLifter LiPo Battery XT90", English, 2022. [Online]. Available: https://www.getfpv.com/lumenier-4000mah-6s-120c-cinelifter-lipo-battery-xt90.html.

[29] M. Niggel, “All about a multirotor FPV drone battery,” GetFPV Learn, 19-Feb-2021. [Online]. Available: https://www.getfpv.com/learn/new-to-fpv/all-about-multirotor-fpv-drone-battery/.

[30] DNDrone Nodes is an online communication platform that brings together experts and enthusiasts in drone research, “How to choose best lipo battery for your drone: Quadcopter 4S: 3S: 2S: 1S,” Drone Nodes. [Online]. Available: https://dronenodes.com/best-lipo-drone-battery-explained/.

[31] C. Győrödi, R. Győrödi, G. Pecherle and A. Olah, "A comparative study: MongoDB vs. MySQL," 2015 13th International Conference on Engineering of Modern Electric Systems (EMES), 2015, pp. 1-6, doi: 10.1109/EMES.2015.7158433.

[32]D. Revina and I. Shanthi, "A NoSQL Solution to efficient storage and retrieval of Medical Images", International Journal of Scientific & Engineering Research, vol. 7, no. 2, -2016, pp. 545-549, 2021. Available: https://www.ijser.org/researchpaper/A-NoSQL-Solution-to-efficient-storage-and-retrieval-of-Medical-Images.pdf.

[33]B.Jose and S. Abraham, "Performance analysis of NoSQL and relational databases with MongoDB and MySQL", Materials Today: Proceedings, vol. 24, pp. 2036-2043, 2020. Available: 10.1016/j.matpr.2020.03.634.

[34] (Artificial Intelligence: What's The Difference Between Deep Learning And Reinforcement Learning?, n.d.)

[35] (Jemin Hwangbo, 2017)

[36]"What is Amazon S3 bucket? - Definition from WhatIs.com", SearchAWS, 2022. [Online]. Available: https://www.techtarget.com/searchaws/definition/AWS-bucket.

[37] Flysky FS-I6 AFHDS 6CH transmitter and 6CH FS-IA6B receiver. English. (n.d.). Available: https://www.getfpv.com/flysky-fs-i6-afhds-6ch-transmitter-and-6ch-fs-ia6b-receiver.html#:~:text=The%20FlySky%20FS%2Di6%20is%20a%20great%20entry%20level%206,programming%20is%20simple%20to%20use.

[38] 2022. [Online]. Available: https://www.ubuy.vn/en/product/4WREC8Y-makerhawk-raspberry-pi-ups-power-supply-uninterruptible-ups-hat-18-650-battery-charger-power-bank-po. [Accessed: 2022].

[39] Z. Qiu, X. Zhu, C. Liao, D. Shi and W. Qu, "Detection of Transmission Line Insulator Defects Based on an Improved Lightweight YOLOv4 Model", Applied Sciences, vol. 12, no. 3, p. 1207, 2022. Available: 10.3390/app12031207.

[40] U. Nepal and H. Eslamiat, "Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs", Sensors, vol. 22, no. 2, p. 464, 2022. Available: 10.3390/s22020464.