B. TECH. PROJECT REPORT

On Ground Object Identification using Drones

BY

Rahul Kumar Gupta (190002050)

DISCIPLINE OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

On Ground Object Identification Using Drones

A PROJECT REPORT

BACHELOR OF TECHNOLOGY
in

ELECTRICAL ENGINEERING

Submitted by:
Rahul Kumar Gupta
Guided by:
Dr. ABHINOY KUMAR SINGH
INSPIRE FACULTY

INDIAN INSTITUTE OF TECHNOLOGY INDORE
NOVEMBER 2022

CANDIDATE’S DECLARATION

We hereby declare that the project entitled **“On Ground Object Identification using Drones”** Bachelor of Technology in ‘Electrical Engineering’ completed under the supervision of **Dr Abhinoy Kumar Singh, Inspire Faculty, Electrical Engineering, IIT Indore** is an authentic work.

Further, we declare that we have not submitted this work for the award of any other degree elsewhere.

Rahul Kumar Gupta

(190002050)

CERTIFICATE by BTP GUIDE

It is certified that the above statement made by the students is correct to the best of my knowledge.

Nov. 30, 2022

Dr. Abhinoy Kumar Singh

Inspire Faculty

Electrical Engineering

Indian Institute of Technology, Indore

Preface

This report on “On Ground Object Identification using Drones” is prepared under the guidance of Dr. Abhinoy Kumar Singh.

Through this report We tried to give the detailed Information about our B.tech Project On ground Object Identification using drones. We tried to cover each and every aspect of our project. We have tried to best of our knowledge to explain the content in the best manner. We have also added pictures for better understanding.

Rahul Kumar Gupta (190002050)

B.Tech. IVth Year

Discipline of Electrical Engineering

Indian Institute of Technology, Indore

Acknowledgements

We would like to express our sincere gratitude to our supervisor Dr. Abhinoy Kumar Singh and Stochastic Control Lab members for their invaluable guidance, comments and suggestions throughout the course of project.

Rahul Kumar Gupta (190002050)

B.Tech. IVth Year

Discipline of Electrical Engineering

Indian Institute of Technology, Indore

ABSTRACT

The field of drones and object tracking and detection is vast. It is a computer technology related to computer vision and image processing which detects certain semantics objects from a certain set of classes in digital images and videos. It has a wide area of applications including video surveillance, mapping , image retrieval etc. Many active research is still going on this field and it is continuously evolving .The technology of drones is in high demand today as its usage and applications are increasing day by day. Many countries are investing heavily in drone technology for its military and civil purposes .This report will give a brief idea of algorithms used for object tracking and detection, drone hardware and its related setup and software.

This project led to the making of a drone which contains a Pixhawk flight controller which contains various sensors . It receives signal from a RC transmitter. It ensures smooth movement and allows various flight modes. It contains a SJ4000 digital camera through which live feed of data can broadcasted on our screen. Then object detection algorithm will be used on these images and videos to detect targets set by user. It requires a high computing power with a GPU card. All data and results discussed at the end of every chapter

CONTENTS

Candidate's Declaration **3**

Certificate by BTP Guide **3**

Preface **3**

Acknowledgements **5**

Abstract **5**

Chapter 1:

Introduction……………………………………………………….1

1.1 Literature Review…………………………………………………………………..1

1.1.1 Oehmichen (1920)………………………………………………………………....2

1.1.2 De Bothezat Helicopter (1922)……………………………………………………..2

1.1.3 Queen Bee…………………………………………………………………………..3

1.2 Recent Advancements…………………………………………………………………3

1.3 TLD………………………………………………………..……………....4

1.4 Mean Shift……………………………………………………………………………..4

1.5 Compressive Tracking…………………………………………………………………5

Chapter 2: Drone Design… **5**

2.1 Components Required… 5

2.1.1 Frame… 5

2.1.2 Flight Controller… 6

2.1.3 Motors… 7

2.1.4 Electronic Speed Controller… 8

2.1.5 Battery… 8

2.1.6 GPS 9

2.1.7 Remote Controller… 10

2.1.8 Gimbal… 11

2.1.9 Camera 12

2.1.10 Video Transmission… 12

2.2 Components Setup…………………………………………………………………13

2.3 Methodology…………………………………………………………...…16

2.4 Software Setup…………………………………………………………………21
2.5 Results……………………………………………………………………………25

Chapter 3: Object Detection… 31

3.1 Introduction… 31
3.2 What is YOLO? 32
3.3 YOLO functioning 32
3.4 How to encode Bounding Boxes? 36
3.5 Intersection over Union & Non-Max Suppression 39
3.6 Implementation of YOLO… 42
3.7 Results… 42

Chapter 4: Object Tracking 45

4.1 Introduction… 45
4.2 Approach… 45
4.3 Experiments… 45
4.3.1 SiamFC 45
4.3.2 SiamRPN… 44
4.3.3 SiamMask… 47
4.4 Implementation of SiamMask… 47
4.5 Results ……………………………………………………………………… 48

Conclusions……………………………………………………………49

References And Sources ……………………………………………50

LIST OF FIGURES

1.1 Oehmichen Helicopter 2
1.2 George de Bothezat helicopter 3
1.3 De Havilland DH82B “Queen Bee” 3
1.4 Military using Drones to find out terrorist… 4
2.1 Frames for Quadcopter & Hexa-copter 6
2.2 Pixhawk 2.4.8 FC 7
2.3 Racerstar BR2212 Brushless Motors 8
2.4 Electronic Speed Controller 9
2.5 Orange Li-Po battery… 9
2.6 GPS Module 10
2.7(a) Transmitter Display… 11
2.7(b) Taranis X9D ACCESS Transmitter 12
2.7(c) Frsky RX8R Receiver 12
2.8 Gimbal 12
2.9 sj4000Camera 12
2.10 RC832 Receiver and TS832 Transmitter 13
2.11 Pixhawk 2.1.8 FC 14
2.12 8 main-out Pin slot &6 aux-out Pin slot on Pixhawk… 15
2.13(a) integrated Power distribution board for Quadcopter 15

2.13(b) integrated power distribution board for Hexa-copter 16
2.14(a) Power module 16
2.14(b) connection between power and Pixhawk 16
2.15(a) Quadcopter motor rotation 17
2.15(b) Hexa-copter motor rotation 17
2.16(a) Pitch, Yaw and Roll axes 18
2.16(b) movement against Pitch, Yaw & Roll axes 18
2.16(c) Pitch, Yaw and Roll axes of a plane 19
2.17(a) Left and Right side movement from rear-view 19
2.17(b) Leftward Motion… 20
2.17(c) Rightward Motion… 20
2.18 Forward and Backward motion of a drone from Rear-view 20
2.19(a) Forward Motion 20
2.19(b) Backward Motion... 20
2.20(a) Clockwise Motion 21
2.20(b) Anti-Clockwise Motion 21
2.20(c) Clockwise Motion 21
2.20(d) Anti-Clockwise Motion.. 21
2.21 choosing frame type in Mission Planner 22
2.22 Accel Calibration in Mission Planner 22
2.23 Compass Calibration in Mission Planner 23
2.24 Radio Calibration 23
2.25 Selecting Flight Modes in Mission Planner 24
2.26 Choosing Failsafe values in Mission 24
2.27 ESC calibration in Mission Planner 25
2.28 Message section 25
2.29(a) Quadcopter from Top-View... 26
2.29(b) Hexa-copter from Side-View 26
2.30 Hexa-copter 26
2.31 Testing of Hexa-copter in IIT Indore campus 27
2.32 Hexa-copter during flight 28
3.1 Object Detection… 29
3.2 Input image taken from our drone at IIT Indore campus... 30
3.3 Input image divided into 3x3 grids 31
3.4 4th grid image 32
3.5 Chair.. 32
3.6 Training Layers for YOLO model 33
3.7 Bounding box on the chair in a grid 34
3.8 Coordinates assigned for a particular bounding 35
3.9 B_x and B_y values assigned for a particular bounding box… 35
3.10 Intersection of actual bounding box and predicted box 36
3.11 Blue region is area of intersection.. 36
3.12 Predicted bounding boxes 37
3.13 Final bounding boxes 38
3.14 (a) and (b) are the inputs for object detection… 40
3.15 Object Detection output for (a) 41
3.16 Object Detection output for (b) 41
3.17 Image taken from our drone… 41
3.18 Image taken from our drone.. 42
4.1 Architecture of SiamFC 44

4.2 Architecture of SiamRPN 45
4.3 Architecture of SiamMask… 46

LIST OF TABLES

2.1 Datasheet table for BR2212 motor… 8
2.2 Frequency for FPV transmitter and receiver 13
3.1 Eight-dimensional coordinate values for each grid… 31
3.2 Eight-dimensional coordinate values for each grid… 32
3.3 Eight-dimensional coordinate values for center right grid… 33
3.4 Eight-dimensional coordinate values for center grid… 34
3.5 Eight-dimensional coordinate values for center grid… 35

Chapter 1: Introduction

One might have heard about drones, the most used product in the field of UAVs. They have become a common sight over the past few years and people are using them for all sorts of purposes, aerial photography, inspection high altitude structures search-and-rescue missions, etc. and all other sorts of purposes.

The use of reconnaissance drones in the Vietnamese War highlighted the main purpose of drones, then and now: to gather information. As the modern technology make advancements and becomes available, various groups such as military, engineers, researchers and hobbyists have been developing new designs and their implementations keep coming out. These days, the drones are commonly used for aerial photography, drone racing, and many other purposes, and even multinational companies are investing in drone equipment and software development.

Some work has been done in the past over object detection and tracking but with the use of a stationary camera. And that makes it quite difficult to locate the target if the target gets out of the frame of the camera mounted on the drone, so a system was needed that can follow the target if it tries to get out of the camera frame.

So, we decided to do a project on drone designing that can detect and track a target given by the user. In order to build the drone, FC, transmitter, receiver, camera, gimbal and a data logging SD card was used. Each component was interfaced, tested and verified to be working properly and were compatible with each other. After that, we have gone through the comparative analysis of detection and tracking algorithms, and finalised the algorithm that we used.

1.1 Literature Review

The earliest unmanned aerial vehicle [1] in the history of drones was seen in 1849 ITALY, when both Venice and Austrian soldiers were fighting in war. Venice was fighting against Austria for its independence. Austrian soldiers attacked the city of Venice, with hot-air, hydrogen, helium filled balloons, with explosives.

The first auto-pilot or pilot-less radio-controlled aircraft was used in World War I. In 1918, the U.S. Military developed the Kettering Bug [2], an unmanned torpedo, a forerunner aircraft of presentday cruise missiles, which was never used in war.

1.1.1 Oehmichen (1920)

Etienne Oehmichen [3] had done several experiments with rotorcraft design in 1920s. He tried six designs among which Helicopter Number 2 had total of 4 rotors and 8 propellers and all driven by a single engine. Oehmichen used a steel-tube frame in his Helicopter Number 2, with two bladed rotors at the end of each arm. To stabilise the helicopter laterally, five propellers spinning in horizontal plane, one propeller was mounted at the nose for steering and remaining pair of propellers were for forward motion as shown in Fig1.1.

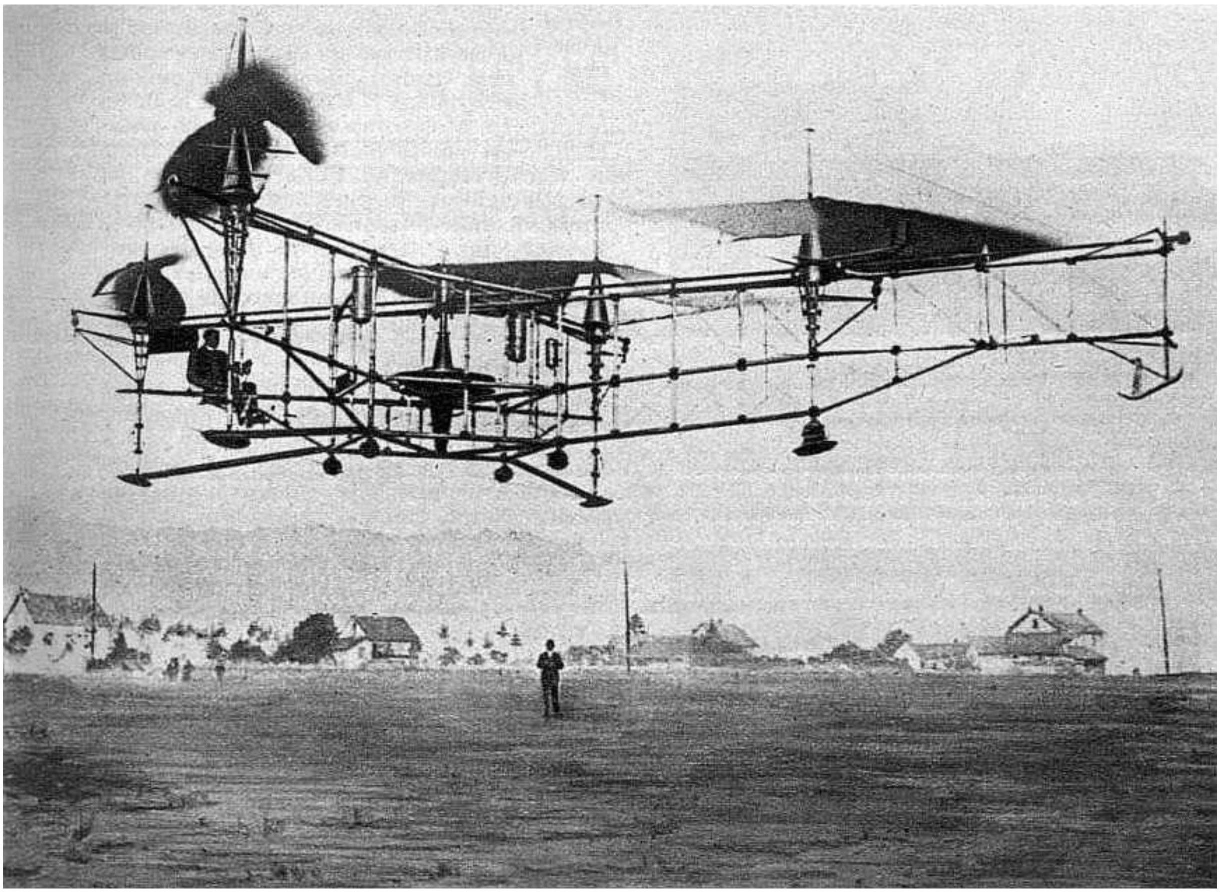


Fig 1.1: Oehmichen Helicopter

1.1.2 De Bothezat helicopter (1922)

Dr George De Bothezat and Ivan Jerome developed the aircraft vehicle, this vehicle has six bladded rotors at the end position of aircraft in X shaped structure. Two propellers are used here with different pitch are used for thrust and Yak control.

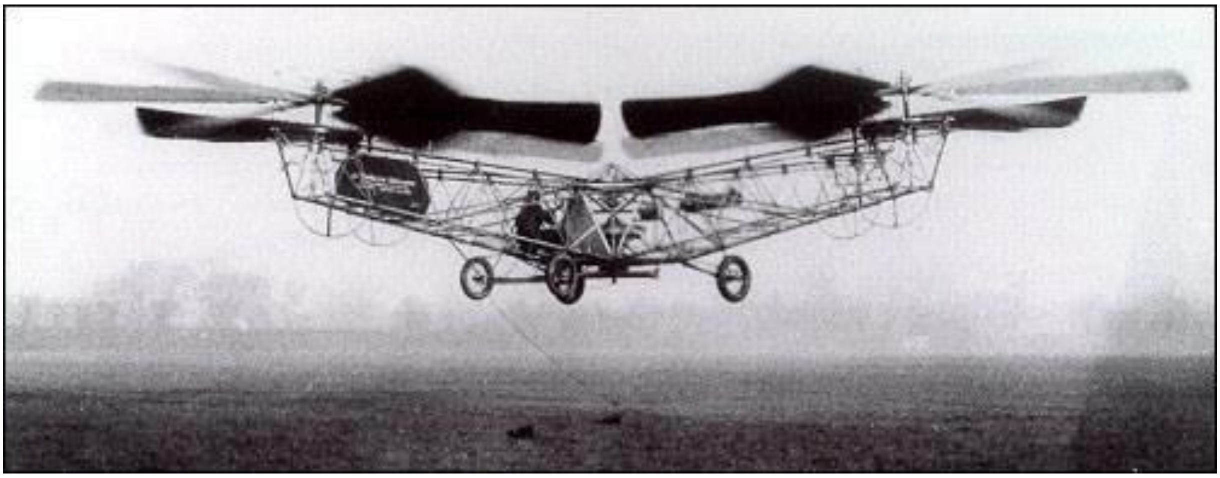


Fig1.2: George de Bothezat helicopter

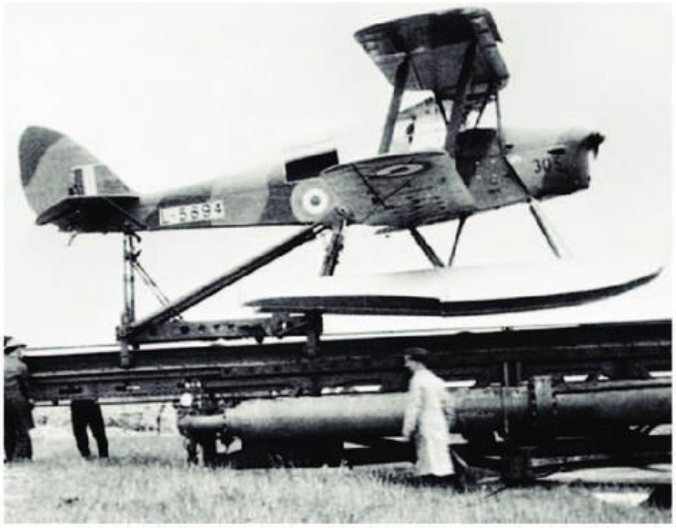


Fig1.3: De Havilland DH82B "Queen Bee"

1.1.3 Queen Bee

The First drone was made in 1935, as a fullsize retooling of De Havilland DH82B "Queen Bee" [5] biplane (Fig1.3). This first plane has a radio and servo operated control. This plane generally it flew pilotless. The term drone is taken from a play on the "Queen Bee" nomenclature.

1.2 Recent Advancements

In recent time many armies started using drones to find out Terrorist. As we all know it is not possible for human to go everywhere so our army used these drones to carry forward the search operation for terrorists. In recent Russia-Ukraine war drones are also used there. So from there we can think of that the importance of drones in future . we used drones because of the following reasons-

High Accuracy
Low cost
High availability

Fig1.4: Military using Drones to find out Terrorist

1.3 TLD (Track Learn Detect)

It is one of the best video object-tracking algorithm. It performs its function in mainly three steps, i.e. tracking, learning and detection [8]. The bounding box will follow the trajectory of the target given by user. The detector monitors the all the displayed tasks that have been running on the backend and modify the bounding box if needed. The learning updates the errors of the detection to avoid tracking failure in future. The learning methods used for estimating the errors is P-N learning by a pair of "experts": (i) P-expert calculates all the mixed detections, and (ii) N- expert calculates all the false alarms.

1.4 Mean Shift

The bounding box for the target is defined in the first frame, as a region of interest. Mean shift [9] algorithm is responsible for separating the target from the background of frame. It moves tracked object data to the local maxima of probability density function. 'Aryabhatta' Coefficient [10] have been used to estimate the difference between two distribution.

1.5 Compressive Tracking

There are a number of tracking algorithms which have the disadvantage of lack of information of the target due to learning of false data. Compressive Tracking [11] algorithm works in a different way, it is a display model, based on the characteristics extracted from the multi scale information. An unique matrix is adopted to efficiently extract the characters. Some of the background and front targets are compressed using the above matrix mentioned. The video object tracking calculations are based on naive Bayes classifier and binary classification.

Chapter 2: Drone Design

In this chapter, we are going to discuss about the components that we require to build a fully function al drone (Quadcopter [12] and Hexa-copter) like flight controller, electronic speed controller, motors, etc. and their working. We are also going to discuss their working principle, about how they communicate with each other to work as a drone system and finally, about the software(s) we may need and use to configure the different configurable components.

2.1 Components Required

In this topic, we provide the necessary information about various parts required for better understanding and their working principle and their setup.

2.1.1 Frame

In Quad-Copter, there are two types Modes first one is X-Mode and the second one is + mode In X mode Frame is in the form of x while in the + form frame will be in + form, these structures holds our frame and provide stability to our drone these frame will be precise and also lightweight. All parts are going to be attached to the frame, so one has to choose the frame size depending upon the purpose it is used for, for e.g. fpv, sport racing, or photography, etc. After deciding the size of the frame, one is recommended to use the recommended setup with his frame for better performance instead of choosing any random setup to avoid improper functioning of the drone. Nowadays, the frame is readily available in market with different materials like carbon fibre, aluminium, wood, fibreglass, etc. we are going to use 250mm size for quad-copter and 550mm size for hexa-copter.

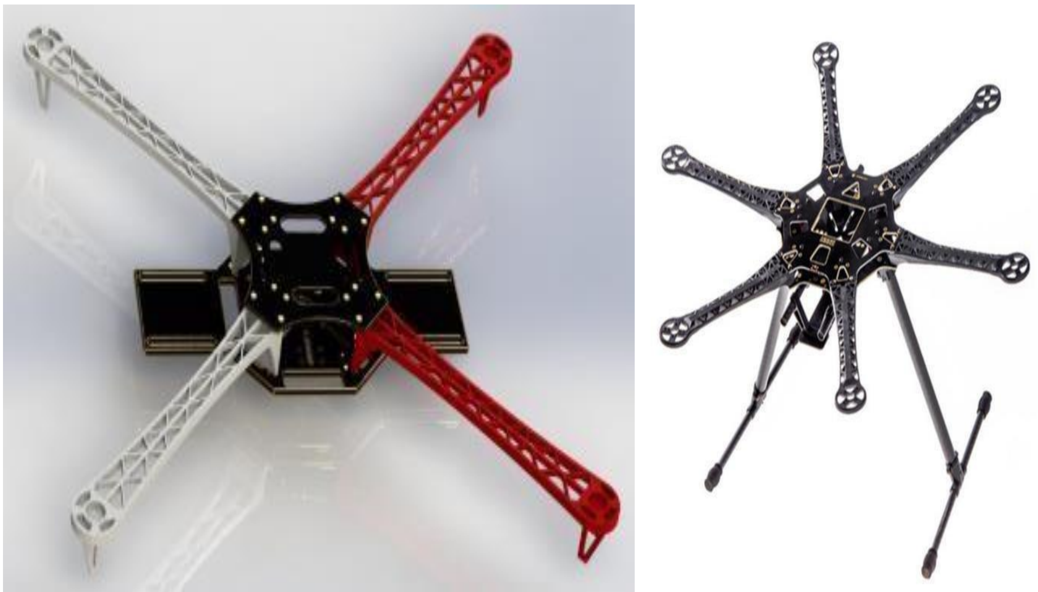


Fig2.1: Frames for Quadcopter and Hexa-copter

2.1.2 Flight Controller

A drone flight controller, or FC, is an important part of a drone and controls most of the onboard electrical components with the assistance of Arduino-like processor and an array of sensors. A flight controller can be used with either a quadcopter, or a helicopter, or a hexa-copter, etc. with the help of associated firmware, which the FC is flashed with. The firmware helps it define the information which the FC provides to different motors for flying. The FC takes the input from RC receiver and feeds it to different sensors like IMU, gyroscope, compass. It then takes the output from these sensors and send it to each motor connected through ESCs. An FC can be flashed using various configurators available like Mission Planner, Beta-flight , Clean-flight , Race-flight or KISS etc. but one has to check if his/her FC is compatible with that configurator.

In this project, we have used a FC called Pixhawk 2.4.8 designed by 3D ROBOTICS in an opensource project. It has an autopilot system capable of autonomous stabilisation [13, 14], way-point based navigation. It also has support for two-way telemetry with radio telemetry module. It can support 8 RC channels with 4 ports. It is provided with a micro-SD card slot for logging purposes which is also called as black-box of drone. There's an external LED system provided with 5 different colours which are red, blue, green, yellow and purple which comes with a flash sequence. The light patterns [15] are associated with sound [16] / tone patterns, which give us the status of the flight controller.

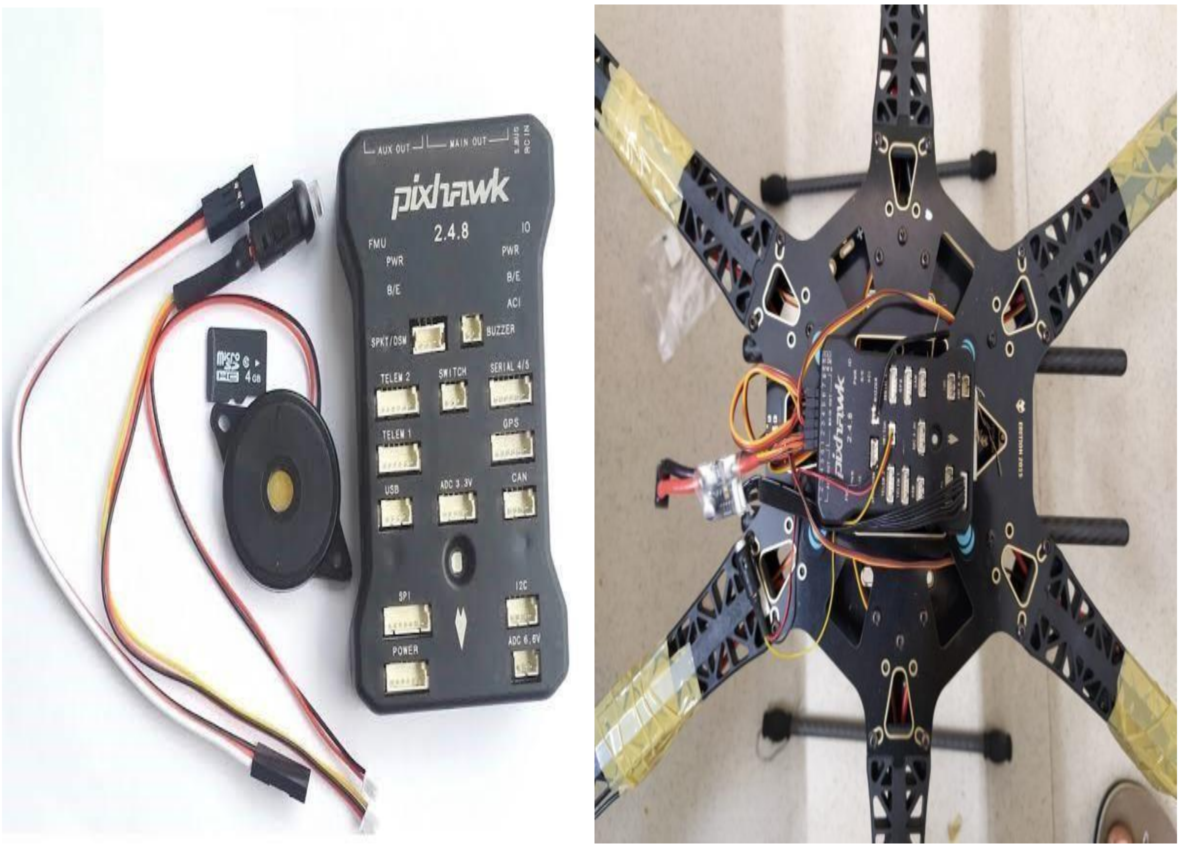


Fig2.2: Pixhawk 2.4.8 Flight controller

2.1.3 Motors

The drone motors are categorised in two parts, i.e., brushed and brushless motors [17]. We need reliable, high-quality brushless motors with rapid response for the drone system to fly smoothly. At some point of time while flying, if one or several of the motors faces any problem, as propellers [18] are attached to the motors, Motors are strong enough to lift our drone and also perform the movement which give through RC Controller stable. We require clockwise and counter-clockwise motor of the same quantity for stable flight.

On the basis of our requirements we have used Race-star BR2212 brushless motor. This Brushless Motor are designed in such a way that can control our drones and these motors are highly reliable. Specifications of each will be like they can give thrust &10 grams at 118 Watt with an efficiency of 6.0g/W.

Fig2.3: Racer-star BR2212 brushless motors

MODEL	KV (rpm/V)	Voltage (A)	Prop	Load Current	Pull (g)	Power (W)	Efficiency (g/W)	LiPo Cell	Weight (g) Approx.
BR2212	920	11.1	8045	7.3	465	81	5.7	2-4S	50
			1045	9.5	642	105	6.1		
	980	11.1	8045	8.1	535	90	5.9		
			1045	10.6	710	118	6.0		

Table 2.1: Datasheet table for BR2212 motor

2.1.4 Electronic Speed Controller

The speed of Brushless Motor is controlled by Electronic speed controller and also to apply dynamic brake electronic speed controller is used. An ESC has three types of wire. One is the set of 2wires that connects the ESC with power supply, another set of 3-wires connects to the Throttle pins of the flight controller and another set of 3-wires connects to the motor. An ESC draws two kinds of current from the battery named as Burst current and constant current. Burst current is the current required to initiate the rotating of the motors. Constant current is the current needed after the motor rotation speed saturates to continue rotating. The current required depend on the task the system is performing. We are using Emax BL-Heli 30A electronic speed controller whose pic is given below.

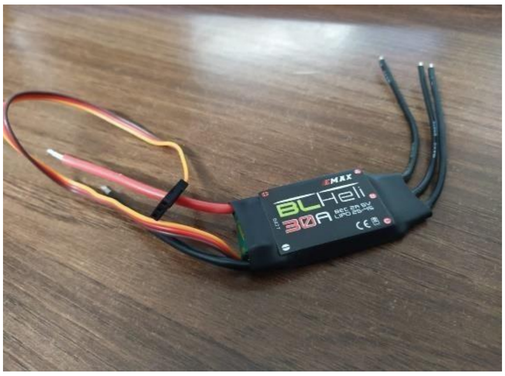


Fig2.4: Electronic Speed Controller (ESC)

Fig2.5: Orange Li-po battery

2.1.5 Battery

A battery (Fig.2.4) is a source of constant power supply which supplies the required input current by the circuit. Batteries are labelled in terms of discharge, i.e., max continuous discharge rate, burst discharge rate also the number of cells available in the battery. For example, some batteries come with 4-cell, others come with 3-cell, the difference between them is the number of cells associated with them, i.e., a 3-cell battery has three cells of battery plates connected in series and a 4-cell battery has four cells of battery plates in series connection. Discharge rate describes the current or power supplied, which are of 2 types, max burst and max continuous discharge. Max continuous discharge rate are described as the maximum current or power that can be provided by the battery for some time continuously. Max Burst discharge are described as the maximum current or power that can be supplied to switch on the system. The battery we used is 5200 mAh 3S 40/80C LiPo battery which has three cells of batteries connected in series 40C as max continuous discharge rate and 80C as max burst discharge rate which can store 5200 mAh of power.

2.1.6 GPS Module

Global positioning system, or in other word GPS, is a satellite-based navigation system which was initially maintained by the US and designed to assist soldiers and military vehicles, but now it is readily available for common masses in the form of a GPS receiver. It is generally used for live positioning or location of a device wherever it may be present on earth. It has different uses based on the purpose required by the user, for example, monitoring, security, mapping and surveying, and other different purposes. In this project, we have used the GPS module [19] (Fig.2.5) mainly designed for the flight controller for various purposes like live positioning of our drone, waypoints mapping. It's also used as a failsafe method for our drone system, i.e., if the output voltage of the battery drops to a pre-fixed value, it will perform a set of function pre-fixed by the user like returnto-land, or it may even try to return to the position where the remote controller of the drone is positioned. It also has an attached compass with GPS module which helps us with better navigation of the drone system.

Fig2.6: GPS module

2.1.7 Remote Controller

A drone controller is a device used by the pilot to give a set of commands to the drone system. It works by sending a signal from the transmitter, which tells the drone what to do because of which it is also called remote controller. A remote controller has two elements, the transmitter

[20] which the pilot holds in his hand to control the drone's movement [21] and the receiver is installed on the drone. The transmitter and the receiver communicate using a band of radiofrequency.

A drone transmitter comes with 6-8 frequency channels, and sometimes it also comes with 12 frequency channels. Each channel gives different feed based on the sticks' movement on the transmitter. The different frequency channels can be customized to perform a different function by the pilot as per his requirement. But it has four channels which give fixed input that is throttle, yaw, pitch, roll. Every transmitter has different compatibility and supports different communicating protocols like PPM, PWM, SBUS, IBUS, DSM2/X etc. We have used Radiolink AT10II remote controller (Fig2.7), which comes with 12 channels, has an inbuilt display (Fig.2.6), in which from channel 5 to 12 can be customized. It supports SBUS, PWM AND PPM protocol. It also has a function in which throttle to thrust ratio can be customized. It comes with R12DS as the compatible receiver. It also comes with OSD flight control telemetry module either PRM-01 or PRM-02 or PRM-03 depending on the model and requirement which gives us the information about throttle, speed, rise, voltage, longitude, latitude, altitude, GPS, etc. on display provided on the transmitter [22].

Fig 2.7(a): Transmitter display screen

Fig2.7(b): transmitter

Fig2.7(c): FR SKY RX 8R receiver

2.1.8 Gimbal

A drone gimbal comes with either single axis, 2-axis or 3-axis freedom of movements or in other words, it has degree of freedom (dof) ranging from 1 to 3 different axes. A drone gimbal is mainly used for stabilising the camera installed on it, enabling it to avoid facing any vibration occurred because of sudden movement during flight and hence allows the camera to capture videos or photos smoothly. We have used a 2-axis gimbal (Fig2.8), which enables the camera to have two dof enabling it to move freely in two different axes of rotation.

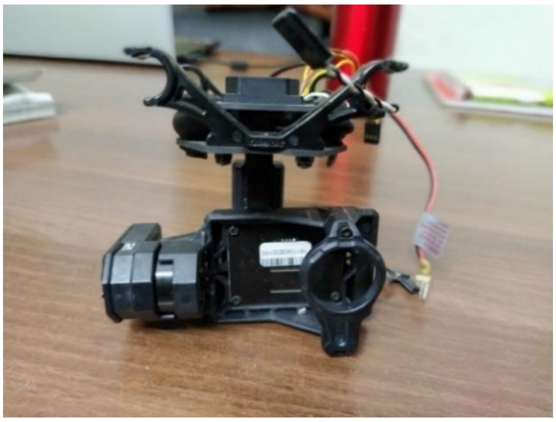


Fig2.8: Gimbal

Fig2.9: Sj4000Camera

2.1.9 Camera

The camera required by the drones is named as FPV camera [23] or first-person video camera, which means that it can send the video to the FPV display screen using a FPV radio. In this project, we have used sj4000 WIFI camera as it is FPV supported. It has a 14MP sensor and 170° wide-angle and 1280*720 HD resolution, which helps us to cover a relatively wide area with much clear resolution. It also comes with 90 min battery life which enables it to record longer without any headache that the camera's battery may die during our flight.

2.1.10 Video Transmission

Video transmission is achieved by using the compatible FPV transmitter and receiver. The camera sends the video in electrical signal to the FPV transmitter which then gets converted into radio signal by the FPV transmitter and transmitted. The FPV receiver then receives the transmitted radio signal and converts it to an electrical signal and sends it to the FPV display screen which the user holds. It has a different operating range (Table 2.2) from the remote controller to avoid any disturbance or mix-up between different radio signals and hence preventing any loss of video signal. In this project, we have used TS832 and RC832 (Fig2.10), which is commonly used for video transmission purposes.

CH FR		**CH**							
		CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8
FR	FR1 or (A)	5895M	5845M	5825M	5805M	5785M	5765M	5745M	5725M
	FR2 or (B)	5733M	5752M	5771M	5790M	5809M	5828M	5847M	5866M
	FR3 or (C)	5705M	5685M	5665M	5645M	5885M	5905M	5925M	5940M
	FR4 or (D)	5740M	5760M	5780M	5800M	5820M	5840M	5860M	5880M

Table 2.2: Frequency for FPV transmitter and receiver

Fig2.10: RC 832 transmitter and TS 832 receiver

2.2 Components Setup

In this topic, we will be discussing how different components communicate with each other so that the final assembly becomes a fully functional quadcopter.

Fig2.11: Connections of Pixhawk

Fig2.11 gives us a rough idea about the connections of between Pixhawk board & GPS, Telemetry, buzzer and safety switch through the dedicated slots in Pixhawk board.

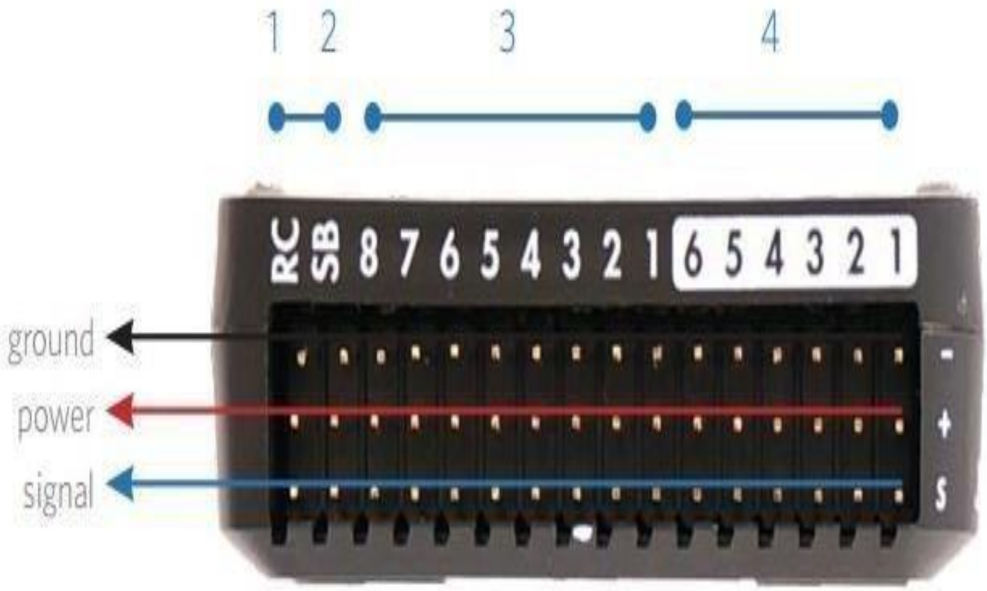


Fig2.12: 8 main-out pin slot and 6 aux-out pin slots on Pixhawk

Pixhawk board has been divided into 4 parts as labelled. No.1 represents the RC-in (RC) slot, No.2 represents S-bus (SB) slot, NO.3 has a total of 8 main-out pin slots, and No.4 has a total of 6 auxout pins in which each slot has three pins. Each pin represents either power or ground or signal as mentioned. The remote-control receiver connects to either RC or SB, but it is advised to connect the receiver to RC pins. The eight main-out pin slots are output pins slot of pixhawk which are used to communicate with the ESCs.

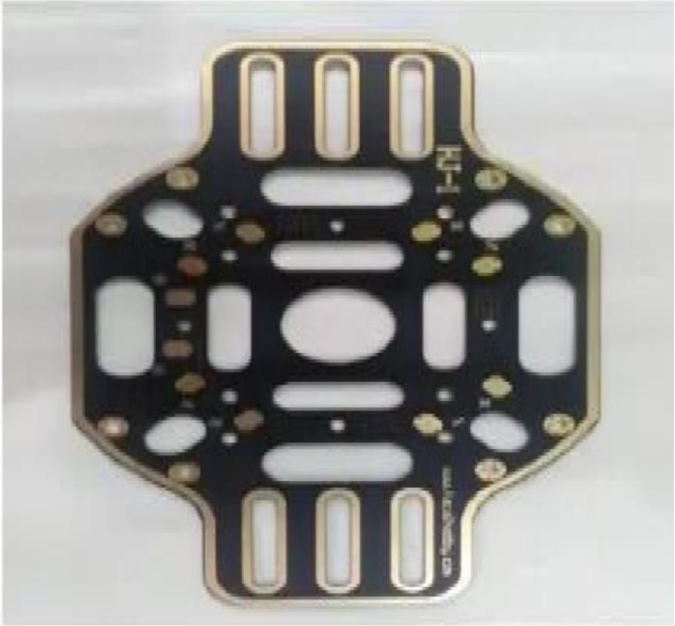
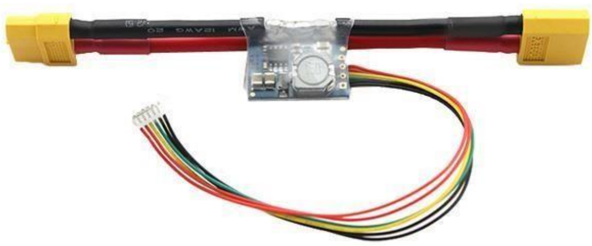


Fig2.13(a): Integrated power distribution board for quadcopter

Fig2.13(b): Integrated power distribution board for hexa-copter

Integrated power distribution board helps to reduce connections directly to the battery. This board provides a better solution by decreasing the number of wires used to supply power to ESCs, which are connected to the motors. Through the power distribution board, we can distribute the power to all different component of our quadcopter.



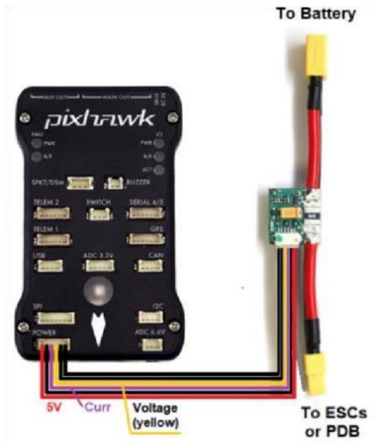


Fig2.14(a): power module Fig2.14(b): connection between power module and pixhawk board

The power module (Fig2.14) is an integral part of the quadcopter which regulates the power supply to the drone circuit. It has one end connected to the battery and another end to the power distribution board. But it also has a set of output wires which is connected directly to the Pixhawk board through the dedicated slot as shown in the figure below.

2.3 Methodology

Today we can fly Drones very easily in any direction, the engineering involved behind the drone is pretty much similar to aeroplane flying.

So, flying process of quadcopter can be explained with the working of the motors and the rotating direction of propeller [24] attached to them as shown in Fig2.16(a) and (b), i.e., propeller design and motor thrust and their setup and the working of the sensors installed in the FC. We send information to our RC controller which is then received by receiver through this receiver Flight Controller will receive information after that flight controller will send the required information to ESCs.This whole process can be depicted as:

Steps -

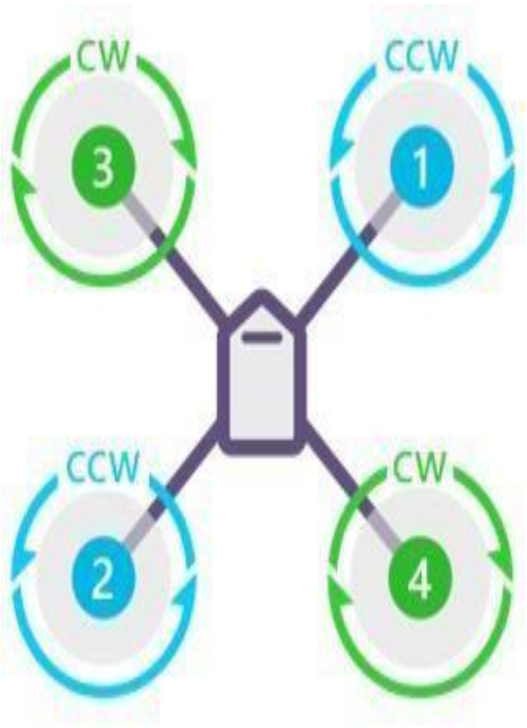
Remote Control Sticks Movement

Central Flight Controller

ESC

Motors and Propellers

Drone movement or Hover.



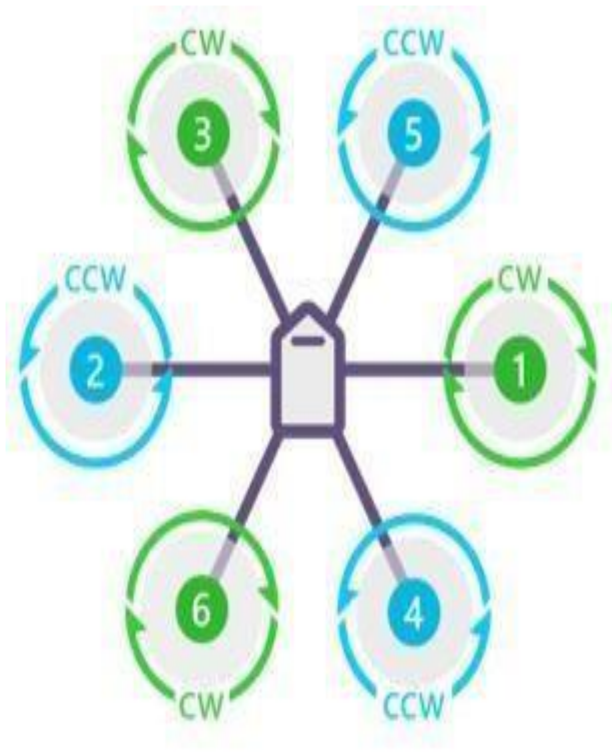


Fig2.15(a): Quadcopter motor rotation Fig2.15(b): Hexa-copter motor rotation

In the Fig2.16(a), one can see from the setup of motors in a quadcopter, that the alternate motors are rotating in one direction where the next to them motors are rotating in opposite direction to ensure the torques produced by each and every propellers cancels each other or in other words the net torque produced is zero and same can be said for the working of motors of hexa-copter. Drone will fly in upward direction when thrust force acting on drone will be greater than Gravitational force on drone if the gravity is greater than thrust force drone will not able to fly.

The Three Axes are Yak, Roll and Pitch. Yak allows the movement of quadcopter in clockwise or anticlockwise, Pitch allows the movement quadcopter in forward or backward direction and Roll allows the movement of quad-copter in left or right direction. People gets easily confused between Roll and Yaw movement so to get a clear idea of the all movements and the 3axes of movement can be known from the Fig.2.16(a), (b) and (c).

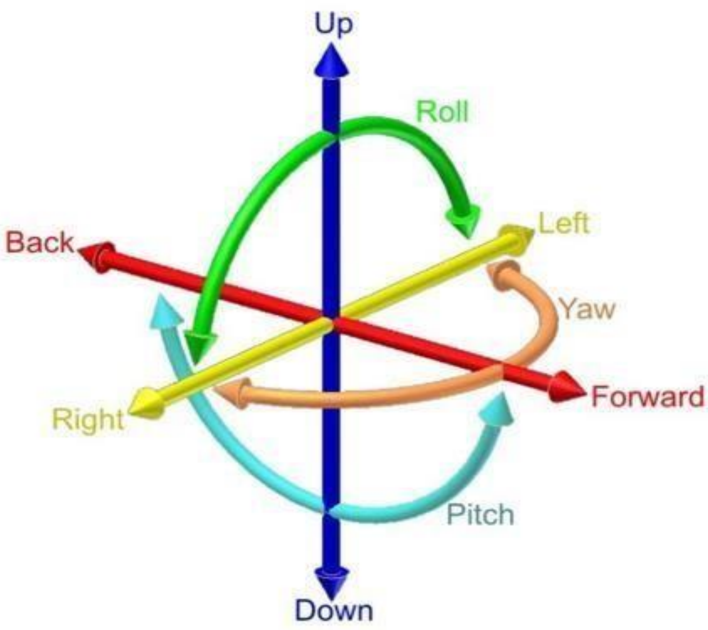


Fig2.16(a): Pitch, Yaw and Roll axes

Fig2.16(b): movement against Pitch, Yaw and Roll axes

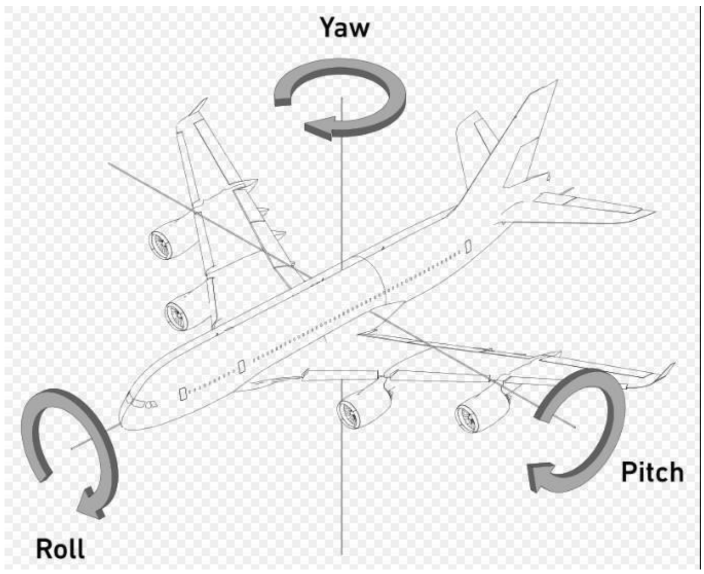


Fig2.16(c): Pitch, Yaw and Roll axis of a plane

Now, the question comes, how a quadcopter performs action according to Yaw, Pitch or Roll. the answer to this question comes from the different thrust produced by each and every propeller. The movement depending on the Roll-axis can be described as when the thrust produced by two rightmost propellers exceeds to that of the to left-most motors, a net torque is produced by the thrust produced by the propellers which tilts the quadcopter towards left and similarly can be explained for hovering towards right by the thrust produced by different motors as shown in Fig2.20, 2.219(a) and (b).

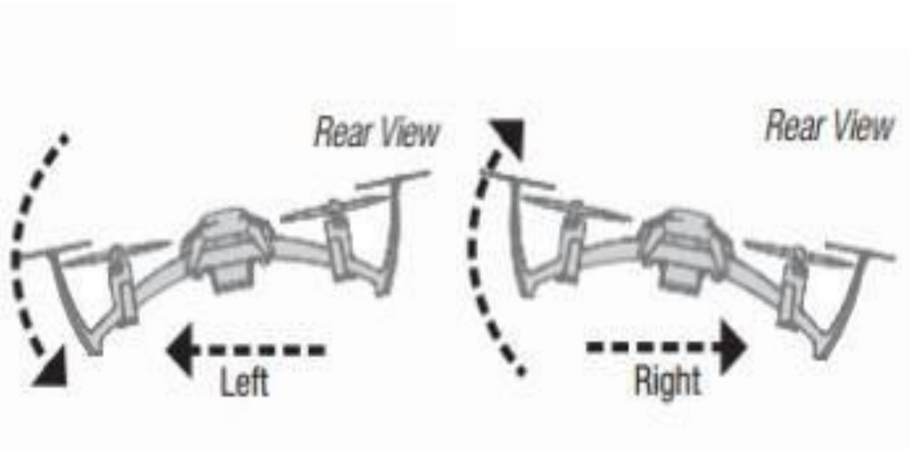


Fig2.17(a): left and right movement of a drone from reaer-view

Fig2.17(b): Leftward motion

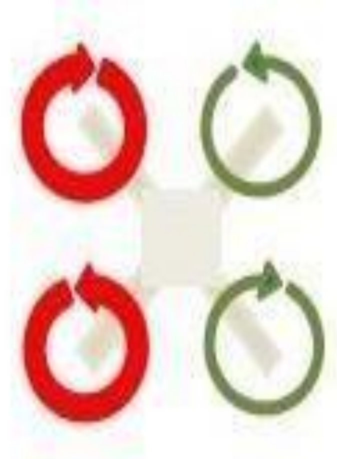


Fig2.17(c): Rightward motion

The movement depending on the Pitch-axis can be described as when the thrust produced by two front-most propellers exceeds the thrust by two rear-most propellers then the quadcopter tilts towards and moves backwards. And the forward movement motion can be explained similarly.

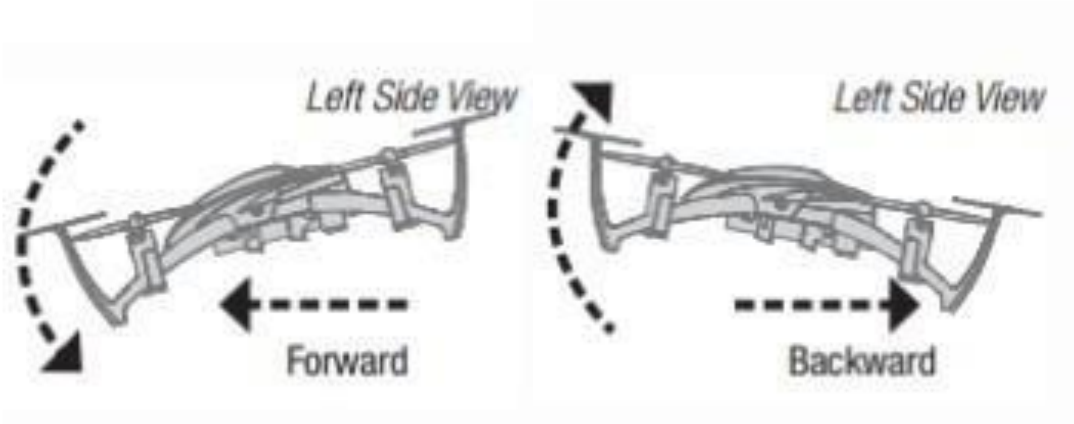


Fig2.18: Forward and Backward movement of a drone from Left-side view

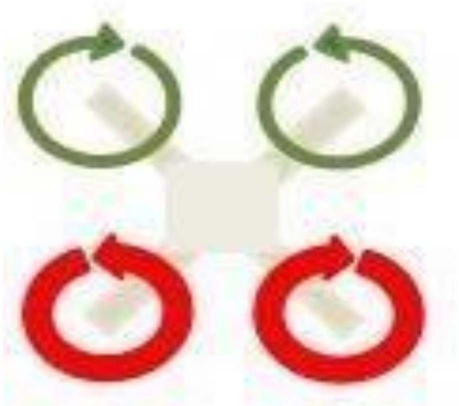


Fig2.19(a): Forward motion

Fig2.19(b): Backward motion

And the motion based on Yaw-axis can be as when all the clockwise rotating propellers produce thrust greater than the thrust produced by all the anti-clockwise rotating propellers, then an angular velocity is generated by the net torque produced in clockwise direction and hence, the quadcopter rotates in clockwise direction.

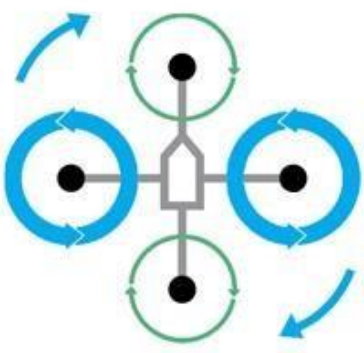


Fig2.20(a): Clockwise motion

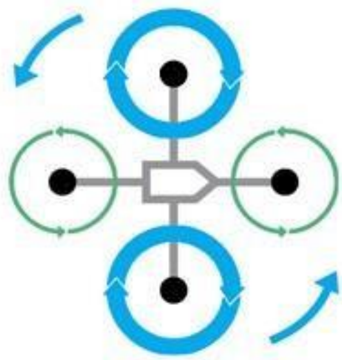


Fig2.20(b): Counter-clockwise motion

Fig2.20(c): Clockwise rotation

Fig2.20(d): Counter-Clockwise rotation

2.4 Software Setup

In this topic, we are going to see how a Pixhawk board is configured to fulfil our requirements. As Pixhawk board is an open-source project, we can use various configurators to configure our FC. Some of the configurators that can be used are QGroundControl, Beta-Flight, or Clean-Flight , etc. but we are going to be discussing about a software available for the same Mission Planner [26] (MP), which is designed by the same company which designed Pixhawk board is also designed keeping Pixhawk and APM board in the view. With this software, we can change the variables and even PID values [27, 28] pre-set by the company to make the drone more stable while flying.

To connect the drone to a ground control system, for e.g. laptop, a mode of communication is set between them using a radio telemetry [29] called SiK Telemetry Radio. This telemetry set uses MAVlink protocol [30] for its communication. It consists two elements, the receiver and the transmitter. It has a range of (out of the box) better than 300m but can be changed by changing some PID values using MP. It is used to retrieve flight information of the drone to our laptop in order to follow several parameters of the drone on the ground.

The first step is to select the frame type and upload the firmware for the same that we are going to use, i.e., X-type Quadcopter.

Fig2.21: Choosing frame type in Mission Planner

After choosing the type and uploading the firmware, we need to calibrate the accelerometer sensor present in the FC so that The FC sets its reference point.

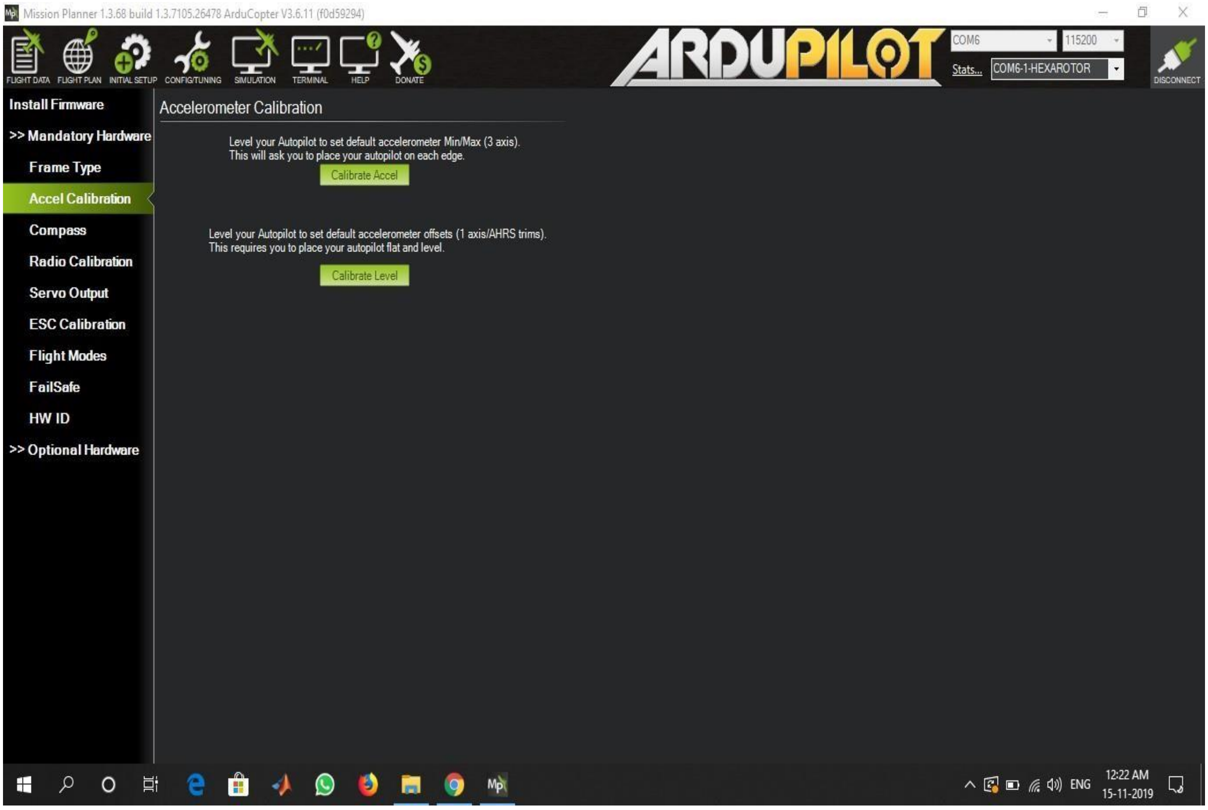


Fig2.22: Accel Calibration in Mission Planner

Then comes the part for compass calibration, in which the inbuilt compass aligns itself with the ever-present magnetic field of the earth for better navigation. One can use externally mounted GPS/compass to reduce the error and better compass alignment.

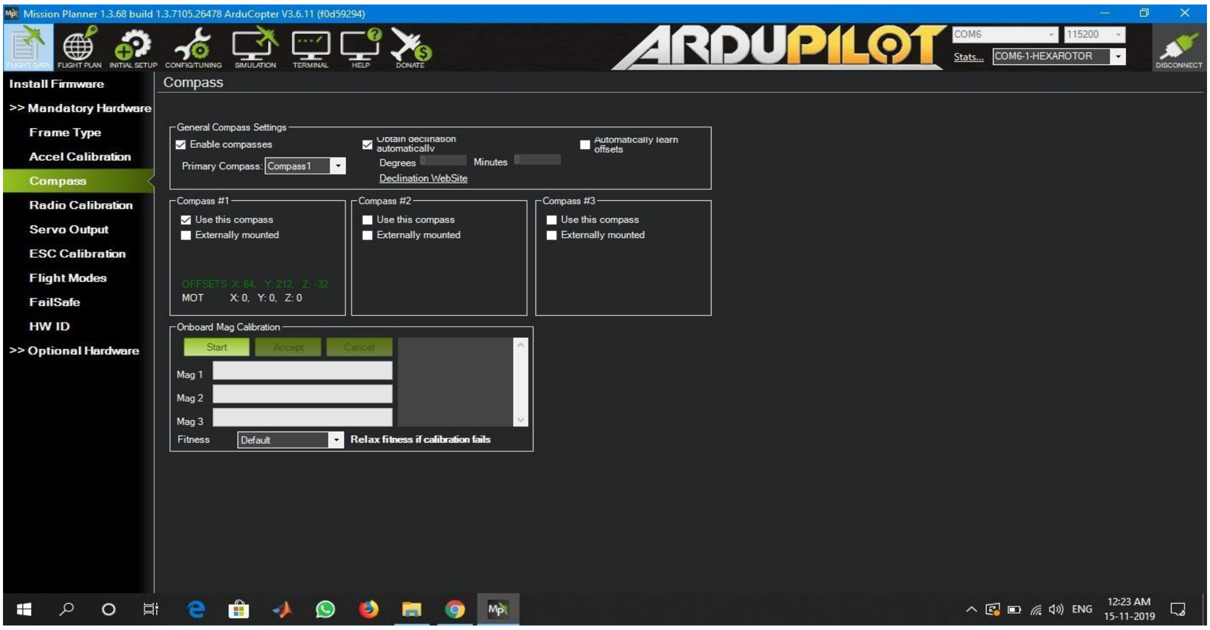


Fig2.23: Compass Calibration in Mission Planner

After compass calibration, comes radio calibration in which we need to check if all the channels present on the RC are functioning properly and each channel has different function. It is also used for the FC to get the max and min value of the sticks if they can be varied.

Fig2.24: Radio Calibration in Mission Planner

After Radio calibration, we need to set the Flight Modes [31]. Flight Modes is a feature available in which we can choose one of the transmitter sticks to perform each function based on its positions, like Stabilise, Alt. Hold, Autotune, Training, Land, etc. We can choose at-most three different flight modes available for our purpose.

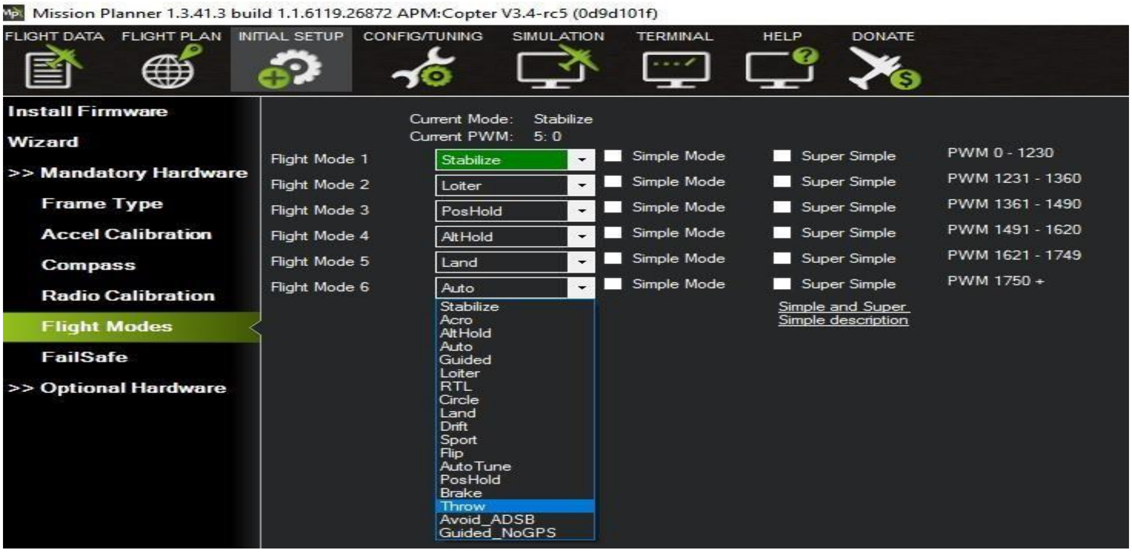


Fig2.25: Selecting Flight modes in Mission Planner

After setting different Flight Modes, we set the Fail Safe value of the battery, i.e., if the output voltage of the battery drops to a certain value, the drone performs a set of pre-loaded commands. It can be set even if you are not using the GPS, but the available set of commands that the drone can perform gets limited.

Fig2.26: Choosing Failsafe values in Mission Planner

After setting Fail Safe value, we need to Calibrate all the ESC [32] we are using. ESCs are designed with different firm-wares. The installed firmware decides the performance of the ESC. This gives information about which protocols it supports and what configuration interface can be used. There are different types of firmware defined for ESC, which are BL-Heli, BL-Heli_S, SimonK, KISS, BLHeli_32 and other manufacturer's own software.

The ESC protocols [33] are the language that the FC and ESC use to communicate, one of the basic tasks is to tell how fast the motor should be spinning. There are different protocols defined for ESCs like Analog PWM, Standard PWM, Oneshot125, etc.

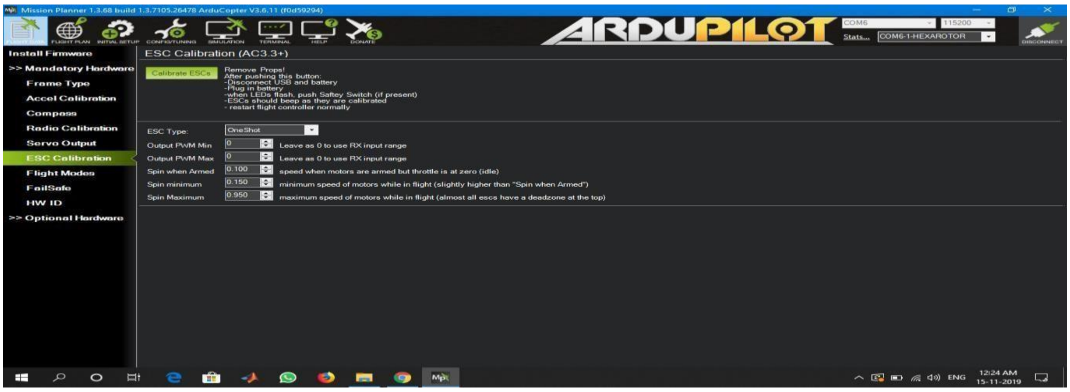


Fig2.27: ESC Calibration in Mission Planner

After all these processes, our quadcopter is ready to fly. But one can get the live status of the drone on the mission planner before or even while the drone is flying in MESSAGES part under the FLIGHT DATA section in MP.

Fig2.28: Message section

2.5 RESULTS

We have successfully achieved to fly our quadcopter using Radiolink AT10II transmitter with Pixhawk FC.

Fig2.29(a): Quadcopter from top-view

Fig2.29(b): Hexa-copter from side-view

Fig2.30: Testing of Hexa-copter

Fig2.31: Testing of hexa-copter in IIT Indore campus

Fig2.32: Quad-copter during flight

Chapter 3: Object Detection

3.1 Introduction

The field of object detection is vast and dynamic. A lot of research has already been done on this field. Due to this there are various algorithms which can be used for object detection .It will give us maximum result accuracy with minimum effort otherwise we have to write our own algorithms and it will a lot of time.

Machine learning researchers uses open-source technology which is very helpful as it reduces cost and time for various enthusiastic students and researchers who wants to use their contribution in their own projects. Most of the work and research has already been done by top researchers. We only have to understand the mechanism behind the algorithm and implement the algorithm on our system.

One such open-source technology related to object detection is YOLO algorithm (You Only Look Once) as shown in fig 3.1. It is supremely fast and gives accurate result . It is a pre-trained models which has its own defined classes and it can also be externally trained. Due to such advancements we do not have to build our algorithms from scratch. It saves us time and increases our knowledge. But we have to understand the mechanism behind the algorithm for better understanding.

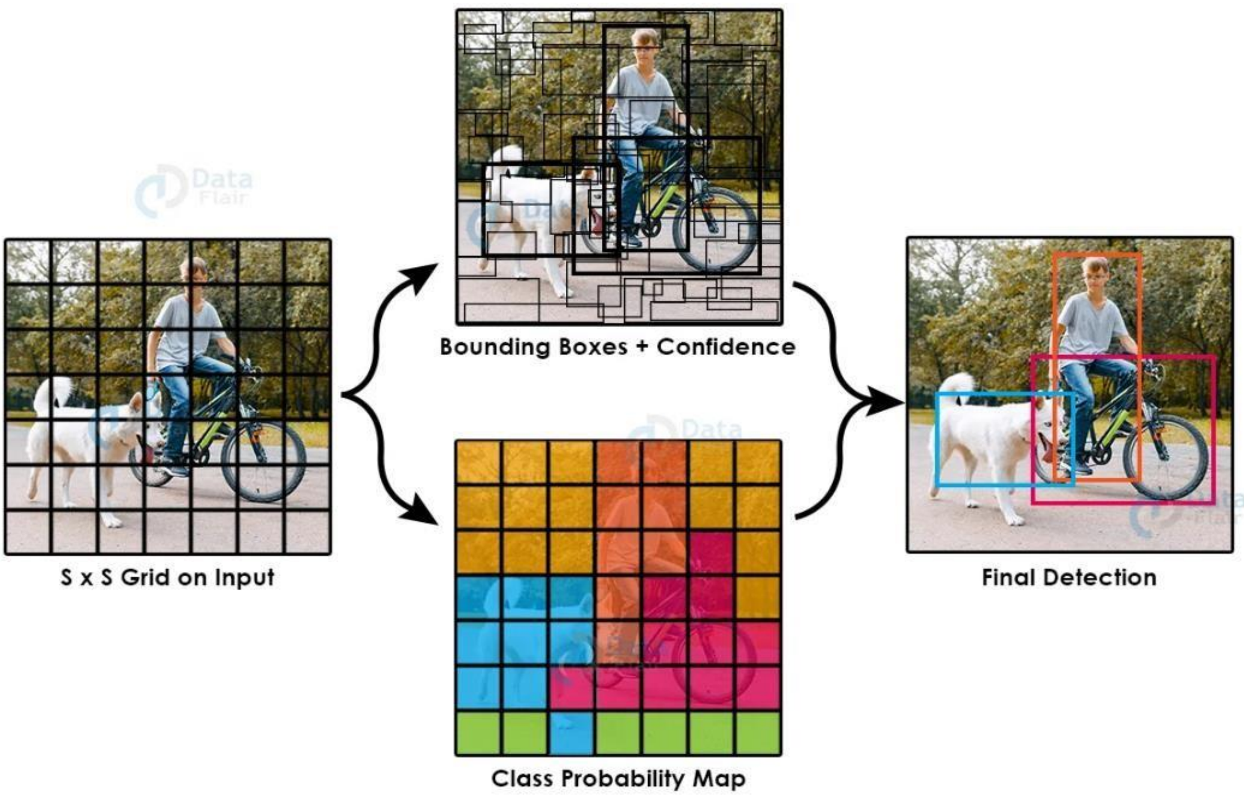


Fig 3.1: Object Detection

3.2 What is YOLO?

YOLO works differently from other object detection algorithm. It takes an image and breaks it into several grids. Each grid has will have its own bounding box co-ordinates and class probabilities. Then data is fed into training algorithm into the model. The main advantage of using YOLO algorithm is due to its high speed - it is extremely fast and can run over 45 frames per second. It will detect the object which is related to that pre- defined class.

It is better than R-CNN technique as it uses sliding window technique and then detect the best bounding box for object detection whereas YOLO is more regression problem which increases its speed and accuracy. Now, we will explain some of the methods used in YOLO.

.

3.3 YOLO functioning

Now we will see how Yolo algorithm actually works and its related steps. Here we will see how images are broken down into grids and its classes are predicted. We will also various parameters related to output variable.

- It will take an input image from user as shown in fig 3.2.

Fig 3.2: Input image taken from at IIT Indore campus room

- It will divide the input image into SxS grids (S = 3) as shown in fig 3.3.

Fig 3.3: Input image divided into 3x3 grids

- Image localization and classification will be applied on every single grid. It then predicts the corresponding class probabilities and bounding box co-ordinates for the objects, if found.

Now step by step explanation of above image grids and its parameters.

The labelled data will be passed into the model for training purposes. The image is divided into 3X3 matrix and let’s have three classes which we want our objects to be classified into.
Let's say we have classes door, chair and tube-light. So for each grid, the label y will be defined which will be a 8 dimensional vector as shown in table 3.1.

Y =	P_c
	B_x
	B_y
	B_h
	B_w
	C_1
	C_2
	C_3

Table 3.1: Eight-dimensional coordinate values for each grid

Here,

- P_c defines whether there is an object is present in the grid or not.
- B_x, B_y, B_h and B_w will specify the bounding box if there is an object present.
- C_1, C_2, C_3 will represent the detected classes. So, if the object found is a chair, C_2 will be 1 and C_1 & C_3 will be 0, and so on.

We have selected the 4th grid from the above example as shown in fig 3.4:

Fig 3.4: 4th -grid image

Since there is no object found in this grid, P_c will be zero and y label for this particular grid as shown in table 3.2.

Y =	0
	x
	x
	x
	x
	x
	x
	x

Table 3.2: Eight-dimensional coordinate values for each grid

Here, ‘x’ means don’t care. It does not matter what all the coordinate values are, as there is no object found in the grid. We have taken another grid in which we have found a chair ($C_2 = 1$) as shown in fig 3.5.

Fig 3.5: Fig 3.5: Bounding box on the chair

Before going further and calculate y label, it is important to first understand the concept of YOLO. How it decides if there is an object found in the grid. In the input image, there are two objects (two chairs), so YOLO specify the center point of these two objects and these objects will be marked to the grid which contains the mid-point of these objects. The y label for the center right grid with the car as shown in table 3.3.

Y =	1
	B_x
	B_y
	B_h
	B_w
	0
	1
	0

Table 3.3: Eight-dimensional coordinate values for center right grid

Since we have found an object in the grid. So, P_c will be equal to 1. Rest x, y, h, w coordinates for bounding box will be calculated relative to the particular grid cell we are considering. Since chair is the second class, $C_2 = 1$ and C_3, $C_1 = 0$. So, for each of the 9 grids, we will have an eight-dimensional output vector. The output will have a shape of 3 X 3 X 8.
So now we have an input image and its output vector. The input image and the output vector will be into the model for it training as shown in fig. 3.6.

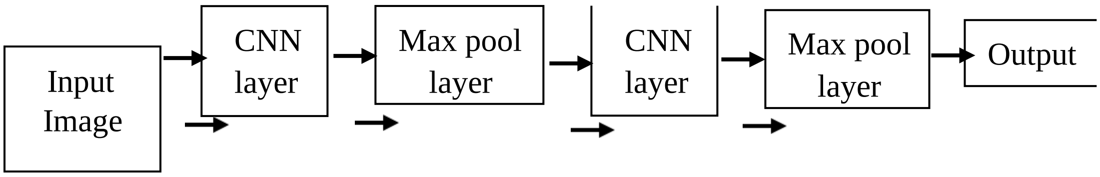


Fig3.6: Training Layers for YOLO model

In order to train our model properly and increase its efficiency, we will run both forward and backward propagation. There can be cases where there will be over lapping of bounding boxes as we have only 3 X 3 grid classification. In order to avoid such confrontation the YOLO algorithm breaks the image into much finer 19 X 19 grids.

There can be still be cases where over-lapping still persists. In order to avoid such thing anchor tags and IOU are used. These techniques will reduce the redundant and other bounding boxes which are showing probabilities lesser than the threshold values

3.4 How to encode Bounding Boxes?

As we have discussed earlier, B_x, B_y, B_h and B_w are calculated relative to the grid cell we are dealing with. We will understand this concept with an example. Consider another grid which contains a chair, as shown in fig 3.7.

Fig 3.7: Bounding box on the chair in a grid

So, B_x, B_y, B_h and B_w will be calculated relative to this grid only. The y label for this grid as shown in table 3.4.

Y =	1
	B_x
	B_y
	B_h
	B_w
	0
	1
	0

Table 3.4: Eight-dimensional coordinate values for center grid

P_c = 1 since there is an object in this grid and since it is a car, C_2 = 1. We will see how to decide B_x, B_y, B_h and B_w. In YOLO, the coordinates assigned to all the grids, as shown in fig 3.8 (0,0)

Fig 3.8: Coordinates assigned for a particular bounding box

B_x, B_y are the x and y coordinates of the midpoint of the object with respect to this grid. In this case, it will be (around) B_x = 0.4 and B_y = 0.5 as shown in fig 3.9.

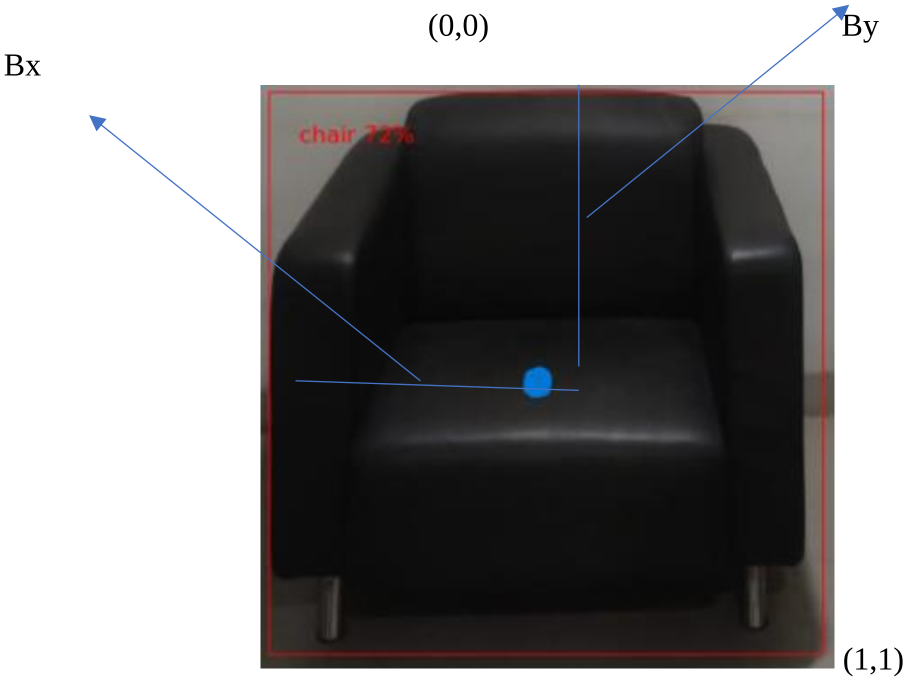


Fig 3.9: B_x and B_y value assigned for a particular bounding box

B_h is the ratio of the height of the bounding box to the height of the corresponding grid cell, which in our case is around 0.8. It can be greater than 1.So, $B_h = 0.8$. B_w is the ratio of the width of the bounding box to the width of the grid cell. So, $B_w = 0.6$ (approximately). The y label for this grid as shown in table 3.5.

	1
Y =	0.4
	0.5
	0.8
	0.6
	0
	1
	0

Table 3.5: Eight-dimensional coordinate values for center grid

In the next section, we will look at more ideas that can potentially help us in making this algorithm's performance even better.

3.5 Intersection Over Union & Non-Max Suppression

Now there can be cases where different bounding boxes will predict same object. But we have to select the best bounding box which best probabilities. It can be through the concept of IoU which means Intersection Over Union. It checks area of intersection between actual bounding box and predicted bounding box. Higher the intersection better the closeness to actual bounding box. Consider the actual bounding boxes for a stop sign as shown in fig 3.10.

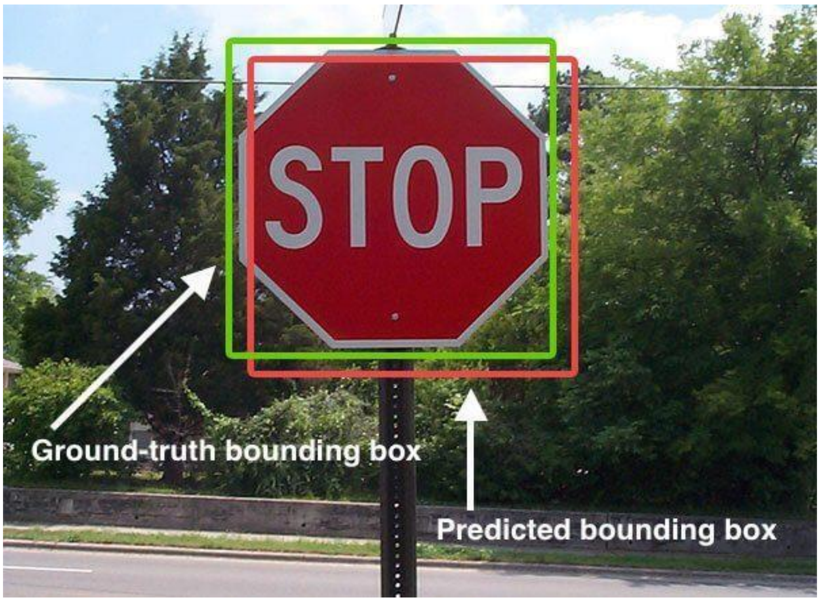


Fig 3.10: Intersection of actual bounding box and predicted box

Here, the green box is the actual bounding box and the red box is the predicted one. A good prediction will be case of higher intersection between these green and red box .IoU will use this concept as shown in fig 3.11.

.

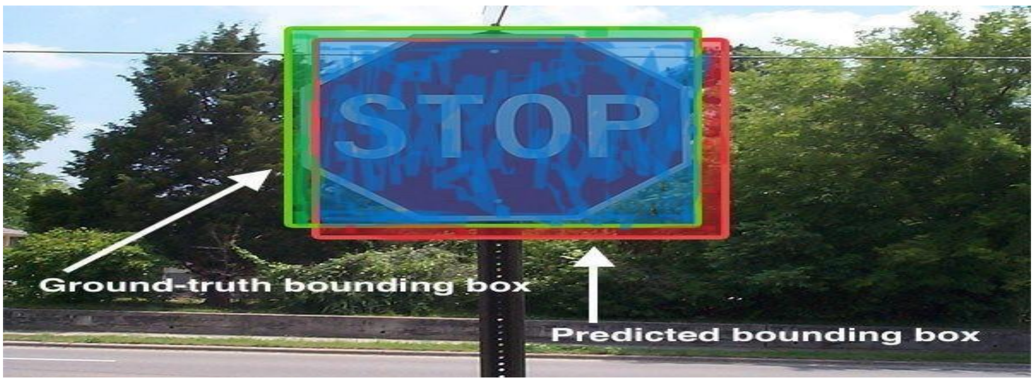


Fig 3.11: Blue region is area of intersection

IoU = Area of the intersection / Area of the union, i.e.

IoU = Area of blue box / Area of blue + red + green shaded part

Higher the IoU value, better the prediction. For conventional purposes 0.5 is considered as threshold value. It can be changed according to task at hand. Higher the threshold value, better the prediction.

Now in order to improve it more significantly we can use one more algorithm which is non-max suppression.

Now it is a common problem that same object can be identified multiple times rather than detecting it one time. This problem can be shown in fig 3.12 .

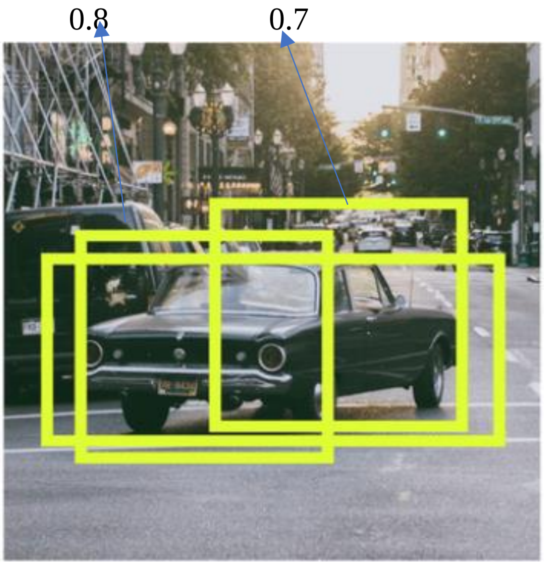


Fig 3.12: Predicted bounding boxes

From the above image we can clearly see same car has been detected more than once. The Non-Max suppression can be used for removing the extra bounding boxes. The steps are as follows:

a. It will take the bounding box which have highest probability. Like in above image (fig 3.12), maximum probability is 0.8, so the corresponding box will be selected.
b. Now, it will go through all the other bounding boxes and select the box which the highIoU are compressed. Like in our example, bounding boxes with 0.8 and 0.7 are compressed/ selected.
c. After suppressing the bounding boxes, it will select the bounding box which have the next larger probability.
d. Then again it will check IoU of surrounding boxes and select boxes which have higher IoU.
e. We will repeat all the above steps until we get the until we get the final output as shownin fig 3.13.

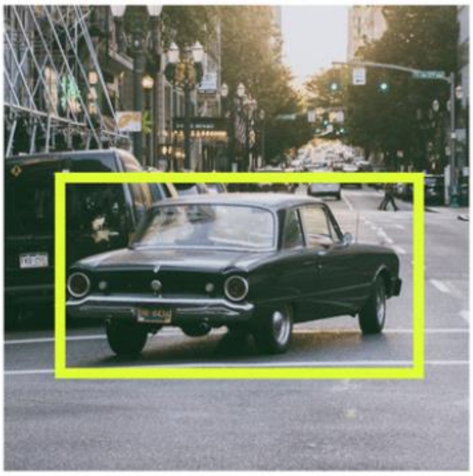


Fig 3.13: Final bounding boxes

Non-Max suppression simply selects the bounding with max probability and suppressing the close-by boxes with non-max probabilities. We can broadly classify our points into following points about Non-Max suppression algorithm:

1. Discard all bounding boxes which have Pc values less than 0.5 .
2. Now the remaining boxes:
    a. Select the box with the highest probability and it will be our output prediction.
    b. Deselect any other box which has IoU greater than the threshold in comparison to output box from above step.
    c. Now keep on repeating the same pattern from step 2 until a final output is achieved .

3.6 Implementation of YOLO

Currently we know three implementations of YOLO each of them has advantages and disadvantages but we using Darknet for our implementation :

- **Darknet**: This was created by same researchers who created the YOLO algorithm. It was written in C with CUDA enabled, hence it does support GPU computation. It is a complete neural network framework, so it really can be used for other tasks besides YOLO detection. The disadvantage is that, since it is written from the scratch (not based on established neural network framework) it will be more difficult to find answers for errors we might find.
- **AlexeyAB/darknet**: It is simply a part of Darknet to support Windows and Linux environment. I have also used this one to check its compatibility with my system and also for the variation in outputs, I have checked the README many times, it is an excellent source to find tips and recommendations about YOLO in general, how to prepare you training set, how to train the network etc .

We can directly implement the YOLO algorithm on these implementations as these come “ready to use”. It simply means we need to download and install them and start putting images for detection. It will detect it with the help of weights file and it will be limited to classes contained in datasets used to obtain these weights.

3.7 Results

We have successfully achieved object detection in images and videos. These are some of the outputs we have got with YOLO algorithm. Some of the input videos and images have taken from our newly built drone.

.

(b) (a)

Fig 3.14: (b) and (a) are the inputs for object detection

Fig 3.15: Object Detection output for (a)

Fig 3.16: Object Detection output for (b)

Fig 3.17 Image taken from drone

Fig 3.18 Image taken from drone

Chapter 4: Object Tracking

4.1 Introduction

We are going to implement video object tracking on a target given by the user. Yes, this can be achieved by the algorithm we are going to use by making a bounding box on the target, and we can effectively track the target in real-time video.

In this chapter, we are ready give focus on object detection part what we actually do in this two stage object tracking method given by Prediction. We first track the object in time t and after that track the object the same object in time (t+1) in next frame without seeing it. Based on the prediction of the algorithm we received prediction output. After that further localize as well segment the outputs of input in t+1 time frame. this re-define track helps us to track the redefined segmentation output which helps us back to batter prediction.

Object Tracking is the process of locating an object by creating a bounding box on it, and the bounding box will follow the trajectory of the target given by the user. It has number of uses – some of which are human-computer interaction, traffic control, surveillance etc. We are going to implement the object tracking algorithm with our newly designed drone.

4.2 Approach

To continuously follow the trajectory of the target given by user, we are going to implement a fully convolutional neural network algorithm called SiamMask [48]. In video object tracking, most methods don't consider the time continuity of object motion. In other words, most methods predict a zero-velocity-object .

To make the SiamMask algorithm suitable for tracking task. We have used the VOT-2016 [53] dataset for the automatic bounding box generation as it offers the highest IOU as reported in [52].

4.3 Experiments

In this Module we will discuss the various approaches we used to object tracking (VOT 2016 and VOT 2018) . as we can see that our method does not depend on selection of tracking object, for better efficiency of our method we can adopt SiamMask[52] with pretrained model.
We will see the algorithms that we have tested for our object tracking purpose as follows:

4.3.1 SiamFC

Siam FC [54] converts the tracking problem into a similarity learning problem, we can use fully convolutional Siamese network is tarin to locate an image with larger search image. The network compares an image to a candidate image and if the similar size image is appears it will result the high score otherwise low score.

Below is the neural network architecture of SiamFC [54] as shown in fig 4.1.

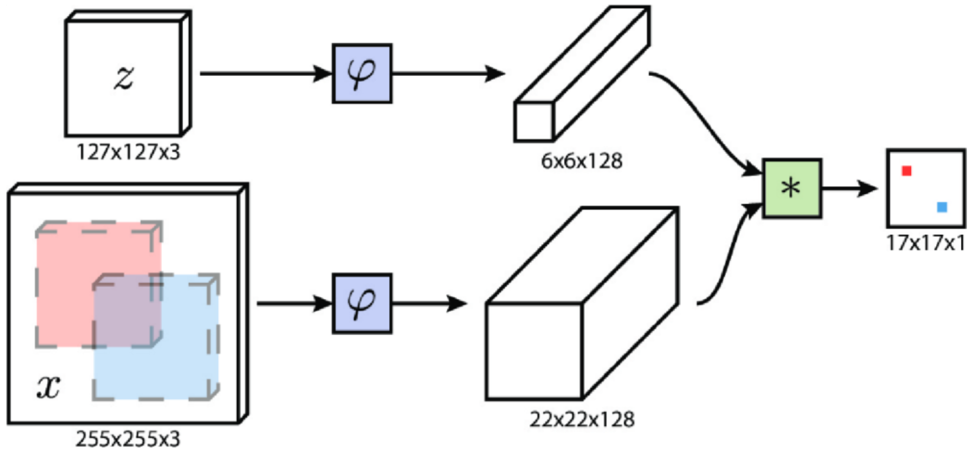


Fig 4.1: Architecture of SiamFC

4.3.2 SiamRPN

SiamRPN [55] focuses on the speed of the neural network-based tracking algorithm. It consists of Siamese framework for feature extraction and region proposal subnetwork including the template and detection branch. The proposed framework is formulated as a local one-shot object detection task, where the bounding box in the first frame is the only example.

Similar as Faster-RCNN, the template branch predicts background and the regression branch compute the bounding box offset for each of the k frames. The template branch of the Siamese subnetwork is pre-computed and the correlation layers (denoted as star) as trivial convolution layers is used to perform online tracking.

Below is the neural network architecture of SiamRPN as shown in fig 4.2.

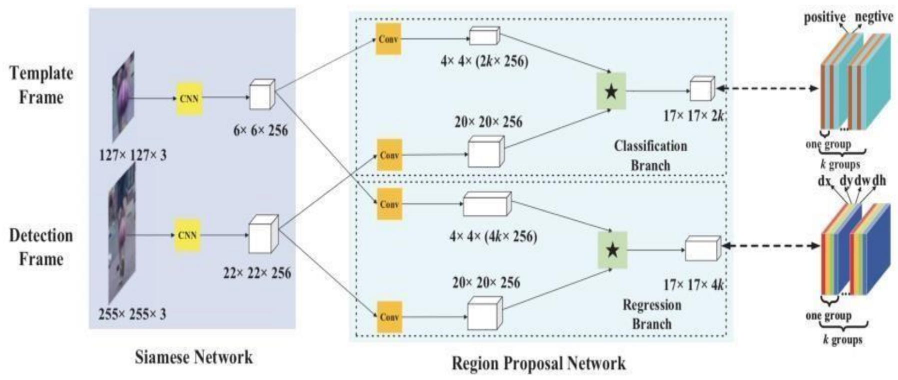


Fig 4.2: Architecture of SiamRPN

4.3.3 SiamMask

SiamMask [52] calculates the problems it faces during the visual tracking and visual object segmentation as a joint learning of three tasks:

1)To track the measure of similarity between the multiple images we take vs the target images we use sliding window fashion.

2)Bounding box regression using a Region Proposal Network.

3)Binary labels are not required in online tracking it only required in offline tracking

Once trained, SiamMask solely relies on the single bounding box of the initialisation, operates online without the updates and the produces rotated bounding boxes at fifty-five frames per second.

Below is the neural network architecture of SiamMask as shown in fig 4.3.

4.4 Implementation of SiamMask

We are going to implement the SiamMask algorithm in Linux environment. To run this algorithm, we need a highly configured computer system with GPU and CUDA enabled, as it is based on real time video object tracking.
Following are the steps to implement SiamMask:

1. Install Anaconda we get

https://repo.continuum.io/archive/Anaconda3-5.3.0-Linuxx86_64.sh

bash Anaconda3-5.3.0-Linux-x86_64.shsource

~/.profile

2. Clone the repository and build the environment and benchmark toolkits git clone

   https://github.com/xl-sr/THOR.git

   cd THOR

   conda env create -f environment.ymlconda activate THOR bash benchmark/make_toolkits.sh

3. Download the SiamMask model we get

   http://www.robots.ox.ac.uk/~qwang/SiamMask_VOT.pth mv

   SiamMask_VOT.pth trackers/SiamMask/model.pth

4. Get the datasets cd data/ bash get_test_data.sh

5. Run your webcam python webcam_demo.py --tracker SiamRPN

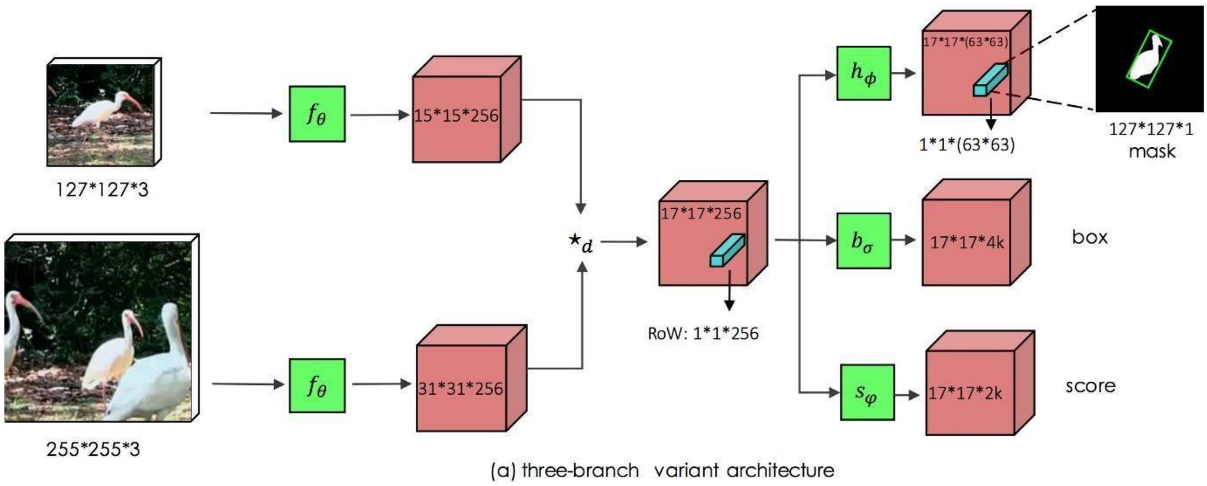


Fig 4.3: Architecture of SiamMask

4.5 Results

We have successfully achieved video object tracking on real-time videos. These are some of the outputs we have got with SiamMask algorithm. All the input videos are taken from our newly designed drone.

Chapter 5: Conclusions

In this report, we achieved following targets:

- We were able to design a drone from scratch and learn all its mechanism and its dynamics.
- We were able to make the drone fly and achieve communication from ground through radio telemetry. We were achieve altitude hold and various other motions such as forward ,backward ,right ,left, upward and downward movement.
- We were able effectively communicate with Gimbal and achieve its 2 -axis movement mounted on our drone for maintaining the center of the frame of camera.
- Salient features of our drone are:
  - It can communicate from ground through RC transmitter.
  - It can do various standard movement.
  - Movement of the camera through a gimbal which is connected to receiver.
  - Effective telemetry air and ground module communication.
  - Object detection and tracking.
- Drone can send real time video and image with compatible FPV transmitter and receiver.
- The received image is acted upon by machine learning algorithms for object detection and tracking.

References And Image Sources

1. https://en.wikipedia.org/wiki/History_of_unmanned_aerial_vehicles
2. https://en.wikipedia.org/wiki/Kettering_Bug
3. https://en.wikipedia.org/wiki/%C3%89tienne_Oehmichen
4. https://en.wikipedia.org/wiki/De_Bothezat_helicopter
5. https://www.dehavillandmuseum.co.uk/aircraft/de-havilland-dh82b-queen-bee/
6. Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-Learning-Detection. In IEEE transactions on pattern analysis and machine learning, VOL. 6, NO. 1
7. Dorin Comaniciu. Mean Shift: A Robust Approach Towards Features Space Analysis. In IEEE transactions on pattern analysis and machine learning, VOL. 24, NO. 5, May 2002
8. https://en.wikipedia.org/wiki/Ku%E1%B9%AD%E1%B9%ADaka
9. http://www4.comp.polyu.edu.hk/~cslzhang/CT/eccv_ct_camera.pdf
10. Design of All-Terrain Rover Quadcopter for Military Engineering Services Shantanu Chaudhary, Arka Prava, Neha Nidhi, Vijay Nath. Lecture Notes in Electrical Engineering book series (LNEE, volume 511)
11. Building a quadcopter: An approach for an Autonomous Quadcopter. VNV Aditya Sharma ; M Rajesh. 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)
12. Self-Navigating Quadcopter. O Guneshwor Singh / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (3), 2015 13. http://ardupilot.org/copter/docs/common-leds-pixhawk.html

14. https://www.getfpv.com/motors/what-is-a-drone-motor.html
15. Stabilizing and control of tilting-rotor Quadcopter in case of a propeller failure. Alireza Nemati, Rumit Kumar, Manish Kumar. Proceedings of the ASME Dynamic Systems and Control Division DSCC 2016

16. A low cost fully autonomous GPS (Global Positioning System) based quad copter for disaster management. HimadriNath Saha, Srijita Basu, Supratim Auddy, Ratul Dey. Arnab Nandy, Debjit Pal, Nirjhar Roy, Subhadeep Jasu, Ankita Saha, SoummyoPriyo Chattopadhyay, Tamanna Maity. 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)

17. https://www.getfpv.com/learn/new-to-fpv/all-about-multirotor-fpv-drone-radio- transmitter-and-receiver/

18. Experimental Validation of an Altitude Control for Quadcopter. Zaki Mustapa, Shakir Saat, A. M. Darsono and H. H. Yusof. ARPN Journal of Engineering and Applied Sciences.

19. Simple GUI Wireless Controller of Quadcopter Dirman Hanafi, Mongkhun Qetkeaw, Rozaimi Ghazali, Mohd Nor Mohd Than, Wahyu Mulyo Utomo, Rosli Omar. Int. J. Communications, Network and System Sciences, 2013

20. Visual tracking and control of a quadcopter using a stereo camera system and inertial sensors. 2009 International Conference on Mechatronics and Automation.

21. https://www.dronezon.com/learn-about-drones-quadcopters/how-a-quadcopter-works- with-propellers-and-motors-direction-design-explained/ 22. https://emissarydrones.com/what-is-roll-pitch-and-yaw

23. https://www.youtube.com/watch?v=I8ugqKsMIpE

24. Design of Control System for Quadcopter using Complementary Filter and PID Controller. V.P. Kodgirwar, Vivek Kumar, Manish Shegokar, SushantSawant. International Journal of Engineering Research & Technology (IJERT), 2014

25. Modelling and Simulation of Quadcopter using PID Controller. Viswanadhapalli Praveen* and Anju S. Pillai. IJCTA, 9(15), 2016

26. http://ardupilot.org/copter/docs/common-sik-telemetry-radio.html
27. http://ardupilot.org/dev/docs/mavlink-basics.html

28. http://ardupilot.org/plane/docs/flight-modes.html

29. http://ardupilot.org/copter/docs/esc-calibration.html?

30. https://robu.in/what-is-oneshot-and-multishot-in-esc-difference-between-oneshot-and- multishot-esc-esc-calibration-protocol/

31. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection 32. https://pjreddie.com/darknet/yolo/

33. Rich feature hierarchies for accurate object detection and semantic segmentation. Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik. Cornell University. CVPR 2014. 34.

Ross Girshick. Fast R-CNN. Cornell University

35. Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Cornell University

36. https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088

37. Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, Silvio Savarese. Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression. Cornell University.

38. https://www.vision.ee.ethz.ch/publications/papers/proceedings/eth_biwi_01126.pdf

39. https://github.com/AlexeyAB/darknet

40. https://developer.nvidia.com/cuda-downloads

41. https://github.com/tzutalin/labelImg

42. https://pjreddie.com/darknet

43. Fast Online Object Tracking and Segmentation: A Unifying Approach

44. Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In European conference on computer vision, pages 850–865. Springer, 2016.

45. Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8971–8980, 2018.

46. Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In European Conference on Computer Vision, pages 103–119. Springer, 2018.

47. Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. arXiv preprint arXiv:1812.05050, 2018.

48. Matej Kristan, Jiři Matas, Aleš Leonardis, Michael Felsberg, Gustavo Fernández, and et al. The visual object tracking vot2016 challenge results. In Proceedings of the European Conference on Computer Vision Workshop, pages 777–823. Springer International Publishing, 2016.

49. https://www.robots.ox.ac.uk/~luca/siamese-fc.html

50. https://www.aljazeera.com/news/2022/10/20/mass-drones-are-a-worry-for-the-future-ofwarfare#:~:text=The%20aircraft%20are%20called%20kamikaze,Ukraine%20was%20a%20month%20ago.

51. http://openaccess.thecvf.com/content_CVPR_2019/papers/Li_SiamRPN_Evolution_of_Siamese_Visual_Tracking_With_Very_Deep_Networks_CVPR_2019_paper.pdf

52. https://www.youtube.com/watch?v=WK_2bpWj35A&t=1400s

53. https://developer.nvidia.com/rdp/cudnn-archive

54. http://ardupilot.org/copter/docs/common-sounds-pixhawkpx4.html#common-soundspixhawkpx4

Rahul_Gupta_BTP

ORIGINALITY REPORT

14%	14%	5%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.analyticsvidhya.com Internet Source	6%
2	medium.com Internet Source	2%
3	deepai.org Internet Source	1%
4	www.slideshare.net Internet Source	1%
5	www.ambientflight.com Internet Source	1%
6	github.com Internet Source	1%
7	internetofthingsagenda.techtarget.com Internet Source	<1%
8	www.zhengzhu.net Internet Source	<1%
9	probots.co.in Internet Source	<1%