FINAL PROJECT REPORT

ON

Crowd Detection Anti-Riot Drone

Project Group Member:-

Md.Azhar Khan(091)
Omji Verma(093)
Pankaj Kumar Singh(094)
Praveen Tiwari(096)

PROJECT GUIDE:-

Er. Atul Agnihotri

Submitted To:

Er. Neeraj Kumar
(Project In-charge)

Department of Electronics and Communication Engineering
University Institute Of Engineering and Technology
CSJM University-Kanpur

2021-22

University Institute of Engineering and Technology

C.S.J.M. University- Kanpur

CERTIFICATE

Dr.VISHAL AWASTHI

This is to certify that ***MD.AZHAR KHAN, OMJI VERMA, PANKAJ KUMAR SINGH and PRAVEEN TIWARI*** of B. Tech. (Electronics & Communication Engineering) 4th year student has delivered his project on "**CROWD DETECTION ANTI-RIOT DRONE** " in a successful and satisfactory manner under the guidance and supervision. This project report is a requirement for the partial fulfilment of his B. Tech. course from

U.I.E.T. , CSJM UNIVERSITY - KANPUR.

Er. ATUL AGNIHOTRI (Project Guide)

Er. NEERAJ KUMAR (Project In-Charge)

(Head of the Department)

ELECTRONICS AND COMMUNICATION ENGINEERING

ACKNOWLEDGEMENT

It is our proud privilege and duty to acknowledge the kind of help and guidance received from several people in preparation of this report. It would not have been possible to prepare this report in this form without their valuable help, cooperation and guidance. First and foremost, We wish to record our sincere gratitude to our project guides for their constant support and encouragement in preparation of this report and interface facilities needed to prepare this report. The Project on "**Crowd Detection Anti-Riot Drone**" was very helpful to us in giving the necessary background information and inspiration in choosing this topic for the seminar.

We are thankful to our project guides **Er. Atul Agnihotri** sir Lecturers in the Electronics and Communication Engineering department for their constant encouragement and valuable guidance.

I am also thankful to **Dr. Vishal Awasthi**, Head of Department , Electronics and communication engineering, for his valuable support.

Date- 24 May 2022

Place- KANPUR

MD.AZHAR KHAN - CSJMA18001390091

OMJI VERMA - CSJMA18001390093

PANKAJ KUMAR SINGH - CSJMA18001390094

PRAVEEN TIWARI - CSJMA18001390096

ABSTRACT

An unmanned aerial vehicle (UAV), commonly known as a drone, is an aircraft without any human pilot, crew or passengers on board. Sometimes it becomes difficult for the armed forces to locate the crowd so that the tear gas can be fired in the right place, it cannot be fired from a long distance, or it becomes difficult to move through the aggressive crowd throwing stones, such to overcome the problem, this Crowd Detection Anti - Riot Drone will help the armed forces in controlling the riot crowd. Thus, activating the tear gas canister which will be dropped on to the crowd using a servo mechanism.

Objective:

Crowd control and monitoring is a difficult task for the police forces if the strength of the crowd is huge. Usually, the police will use tear gas to disperse the crowds in uncontrolled situations but during this process, the police might be injured to avoid such situations. We can use the drone to control the crowds.

With the advancement in technologies, the usage of drones increased a lot. In this drone project, we will propose a solution to this problem.

- Detection of crowd at Initial Stage.
- Area Surveillance
- Launching Teargas Canister without going in a crowd.
- Real-time video image of the situation on the ground.

Content

1. Introduction

An unmanned aerial vehicle (UAV), commonly known as a drone, is an aircraft without any human pilot, crew or passengers on board. UAVs are a component of an unmanned aircraft system (UAS), which include additionally a ground-based controller and a system of communications with the UAV. The flight of UAVs may operate under remote control by a human operator, as remotely-piloted aircraft (RPA), or with various degrees of autonomy, such as autopilot assistance, up to fully autonomous aircraft that have no provision for human intervention.

UAVs were originally developed through the twentieth century for military missions too "dull, dirty or dangerous"for humans, and by the twenty-first they had become essential assets to most militaries. As control technologies improved and costs fell, their use expanded to many non-military applications. These include aerial photography, product deliveries, agriculture, policing and surveillance, infrastructure inspections, science, smuggling,and drone racing.

In the riots, people violently throw stones and set buses on fire. Police and armed forces are appointed to control such situations. They have used sticks or batons, water cannons and tear gas to disperse an unlawful assembly. Sometimes it becomes difficult for the armed forces to locate the crowd so that the tear gas can be fired in the right place, it cannot be fired from a long distance, or it becomes difficult to move through the aggressive crowd throwing stones, such to overcome the problem, this Crowd Detection Anti - Riot Drone will help the armed forces in controlling the riot crowd. The drone is controlled by RC remote; this transmitter sends wireless commands to the receiver by pressing the second button of the remote and the receiver will start the camera and wirelessly stream the live video to the screen. A joystick on the remote controls the drone's direction (forward, backward, left and right). The other joystick is used for altitude control, controlling how high the drone flies. Using one button, from the remote we control the drop of the tear canister. For the detection of human crowd, we will use deep neural networks. The aim of our work is to provide light architectures, as imposed by the computational restrictions of the application, that can effectively distinguish between crowded and uncrowded scenes, captured from drones. When the drone detects the crowd the security person controlling the drone will press a button on the remote which will cause the smoke canister to ignite. Thus, activating the tear gas canister which will be dropped on to the crowd using a servo mechanism.

2. Quadcopter

The quadcopter, also called the quadrotor helicopter or quadrotor, is a multi-colored helicopter that is propelled and driven by four rotors. Quadcopters are classified as rotorcraft, as in airplanes with fixed wings, due to the fact that their lift is produced by a group of rotors (propeller facing upwards).

Quadcopters typically use two pairs of identical propeller propellers; two consecutively like clockwork (CW) and two opposite clocks (CCW). These use independent variations of the speed of the entire rotor to detect control. By changing the speed of all rotors you may be able to produce exactly the amount you want; find the center of the thrust both side by side and length, and create the total torque you want, or the rotating force.

Quadcopters are different from conventional helicopters, which use rotors ready to adjust the height of their blades in a flexible manner as they rotate the rotor hub. In the early days of flight, quadcopters (then called 'quad rotors' or simply 'helicopters') were seen as possible solutions to some of the ongoing problems in direct flight. Control problems caused by torque (as well as performance problems from rotating anti-torque, which does not produce a useful lift) are usually eliminated by rotation, so short blades are much easier to build. Several man-made designs appeared between the 1920's and 1930's. These vehicles were among the most successful heavy-duty and heavy-duty vehicles (VTOL). However, premature prototypes have a malfunctioning function, and recent prototypes require an excessive amount of experimental activity, due to the addition of poor durability and limited control.

In the late 2000's, advances in electronics allowed the integration of cheap lightweight airplane controllers, accelerometers (IMU), global landing systems, and cameras. This has led to the quadcopter configuration becoming popular in small unmanned vehicles. With its small size and smooth operation, these quadcopters can fly in and out.

Small in size, quadcopters are cheaper and more durable than traditional helicopters thanks to their mechanical simplicity. Their shorter blades are also advantageous because they have low K.E. which reduces their ability to cause harm. For small scale quadcopters, this makes the vehicles safer for close contact. It is also possible to suit quadcopters with guards that surround the rotors, reducing the potential for damage. However, as size increases, fixed-propeller quadcopters develop disadvantages relative to conventional helicopters. As the blade size increases, their speed increases. This means that it takes longer to change the speed of the blade, which negatively affects the control. Helicopters do not experience this problem because increasing the dimensions of the rotor discs does not have a significant effect on the power to regulate blade pitch.

i. literature survey

Drones are becoming a major part of society. From hobby drones to military equipment to delivery services, more and more drones are being used today. In fact, between 2014 and 2015, drone sales increased by 63% and only continued to rise. But like all types of technology, drones are not flawless. And if you own a drone, it's important to remember the potential problems you may face. So let's go over some common drone problems and how to fix them.

ii. Decreased Battery Life

Having a working battery is one of the most important parts of flying a drone. This is why it is important to properly charge the battery before attempting to fly. When the drone is not in use, it is best to store the battery in a cool, dry place – the battery should not be left inside the drone when not in use. This can help maintain charge and extend the lifespan of the battery. Additionally, it is important not to overcharge the battery, which can significantly reduce its lifespan. Overall, taking excellent care of the battery will ensure that it lasts as long as possible.

iii. Incorrect flight direction

If a drone's compass is not properly calibrated, drone users may experience incorrect or abnormal flight direction during flight. This can also happen if the drone is fitted with a mounted flight controller and therefore the specifications are incorrectly placed. This is why it is important to offer your drone a quick inspection before each flight. Calibrating the compass can usually help solve this problem but sometimes reprogramming the remote control can help. If none of those solutions work, then drone repair services may also be needed to find the basis of those drone problems.

3. Scope of Quadcopter Drone

The standardization of controls and capabilities of drone-driven IoT will occur thanks to industry diffusion, regulation, and economics. This is an enormous opportunity for drone manufacturers who will naturally start to use similar apps, tools, and interfaces. The market for commercial/ civilian drones is expected to grow at a compound annual growth rate (CAGR) of 19% between 2015 and 2020. The following industries may benefit tremendously from this commercialization of drones:

- **Agriculture** – Aerial or orthographic surveillance of land to identify and control potential crop threats, such as pests or fungus infestations, will be possible. Soil irregularities, like water saturation and erosion, might be easily found. Moreover, aerial drones could scan fruit for sugar and temperature variables to seek out potential problems and also identify areas for selecting crops at optimal times.

- **Mining** – Drone-driven IoT can be used to survey and audit various aspects of mining operations, including berm erosion, road analysis, subsidence, directing automated ground vehicles, and security.

- **Construction** – Drone-driven IoT can be used to survey build sites, monitor operations and progress, provide 3D mapping, inspect construction materials and check security.

- **Utilities** – Power lines, turbines, towers, and dams can be inspected by drone-driven IoT. Property surveys, equipment monitoring, and security functions also can be performed by drone-driven IoT.

- **Delivery Services** – Drone deliveries could begin as soon as regulations are set and services are available to expand operations.

- **Film and tv** – Drones are already getting used to mount cameras and take aerial shots which were previously possible only by the utilization of helicopters. Drones have provided a less noisy and vibration-free medium during this sector.

- **Emergency Services** – Drone driving is often used for traffic surveillance and accident assessment. they will even be wont to carry equipment, water, relief packages, and supply other logistical support functions.

- **Riot Control -** Used to disperse crowds.

4. METHODOLOGY

i. Vertical Motion

Drones use rotors for propulsion and control. We can think of the rotor as a fan, as they do much of the same thing. The spinning blades push the air down. All forces come in pairs, which means that as the rotor pushes down on the air, the air pushes up on the rotor. This is the basic idea behind lift, which comes down to controlling the upward and downward force. The faster the rotors spin, the higher the lift, and vice versa.

Now, a drone can do three things in the vertical plane: hover, ascend or descend. To hover, the net thrust of the four rotors pushing the drone upward must be equal to the force of gravity pulling it down. easy. So what about going up, which pilots call climbing? Simply increase the thrust (speed) of the four rotors so that there is a non-zero upward force that exceeds the weight. After that, you can reduce the thrust a bit---but there are now three forces on the drone: weight, thrust and air drag. So, you still need the thrusters to be bigger than with just a hover.

Descending requires doing the exact opposite: simply reduce the rotor thrust (speed) so that the net force is downward.

ii. Turning (Rotating)

Let's say we have a hovering drone that points north and we want to rotate it east

In this configuration, the red rotors are rotating counterclockwise and the green ones clockwise. With two sets of rotors rotating in opposite directions, the total angular momentum is zero. Angular momentum is like linear momentum, and you calculate it by multiplying the angular velocity by the moment of inertia. Stop. What is the moment of inertia? It is similar to mass, except that it is related to rotation. Yes, it gets rather complicated, but all you need to know is that the angular momentum depends on how fast the rotor spins.

If there is no torque on the system (the system is drone here), the total angular momentum must remain constant (zero in this case). To make things easier to understand, I would say that the red counterclockwise rotors have a positive angular momentum and the green clockwise rotors have a negative angular momentum. I would assign a value of +2, +2, -2, -2 to each rotor, which adds up to zero (I omitted the units).

Let's say you want to rotate the drone to the right. Suppose I reduce the angular velocity of rotor 1 such that it now has an angular momentum of -1 instead of -2. If nothing else, the total angular momentum of the drone would now be +1. Of course, this cannot happen. So the drone rotates clockwise so that the body of the drone has an angular momentum of -1.

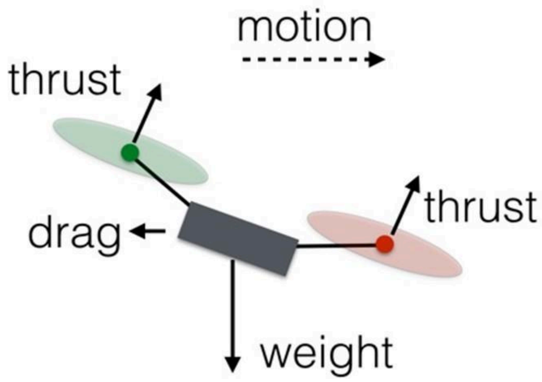
Decreasing the spin of rotor 1 did indeed cause the drone to rotate, but it also decreased the thrust from rotor 1. Now the net upward force does not equal the gravitational force, and the drone descends. Worse, the thrust forces aren't balanced, so the drone tips downward in the direction of rotor 1.

To rotate the drone without creating all those other problems, decrease the spin of rotor 1 and 3 and increase the spin for rotors 2 and 4. The angular momentum of the rotors still doesn't add up to zero, so the drone body must rotate. But the total force remains equal to the gravitational force and the drone continues to hover. Since the lower thrust rotors are diagonally opposite from each other, the drone can still stay balanced.

iii. Forwards and Sideways

What is the difference between moving forward or backward? None, because the drone is symmetrical. an equivalent holds for side-to-side motion. Basically, a quadcopter drone is sort of a car where every side is the front. This suggests that explaining the way to move forward also explains the way to withdraw or to either side.

To fly forward, I want a forward component of thrust from the rotors. Here may be a view (with forces) of a drone moving at a continuing speed.



How do we get the drone into this position? we could increase the rotation rate of rotors 3 and 4 (the rear ones) and decrease the rate of rotors 1 and 2. The total thrust force will remain equal to the weight, so the drone will stay at the same vertical level. Also, since one of the rear rotors is spinning counterclockwise and the other clockwise, the increased rotation of those rotors will still produce zero angular momentum. The same holds true for the front rotors, and so the drone does not rotate. However, the greater force in the back of the drone means it will tilt forward. Now a slight increase in thrust for all rotors will produce a net thrust force that has a component to balance the weight along with a forward motion component.

5. Components Used

- **F450 Quadcopter Frame:** It is a 450mm quad frame built from quality materials. The main frame is glass fiber while the arms are constructed from ultra-durable polyamide nylon. This version of the F450 features integrated PCB connections for direct soldering of your ESCs.This eliminates the need for a power distribution board or messy multi-connectors keeping your electronics layout very tidy. F450 also comes with stronger molded arms, so no more arm breakage at the motor mount on a hard landing.

  Assembly is a breeze with pre-threaded brass sleeves for all of the frame bolts, so no locknuts are required. It utilizes one size of bolt for the entire build, making the hardware very easy to keep in order and only requiring one size of hex wrench to assemble.

- **Motors:** A2212 is a brushless outrunner DC motor specifically made to power Quadcopters and Multirotors. It is a 1400kV motor. It provides high performance, super power and brilliant efficiency. These motors are perfect for medium size quadcopters with 8 inch to 10 inch.

- **ESCs or electronic speed control:** 30A BLDC ESC Electronic Speed Controller is specifically made for quadcopters and multi-rotors and provides faster and better motor speed control giving better flight performance compared to other available ESCs.30A BLDC ESC ELECTRONIC SPEED CONTROLLER can drive motors which consume up to 30A current. It works on 2S-3S LiPo batteries. It has an onboard BEC which provides regulated 5V(2A max draw) to power the flight controller and other onboard modules. It can be used to control our brushless motors with a 2S-3S LiPo (make sure the motor doesn't draw more than 30A).

- **Propellers:** This 1045 propeller can be used with brushless motors with ratings 800-2200 kV. With a low kV motor (800 - 1400 kV), this propeller provides smooth flights with longer flight times for FPV and aerial photography. This propeller with a high kV motor (over 1200 kV) provides fast flights for acrobatic flights.

● **Batteries:**Lithium polymer (LiPo) batteries are a type of battery now used in many consumer electronics devices. They have been gaining popularity in the radio control industry over the past few years and are now the most popular choice for anyone looking for longer time and higher power.

● **Flight Controller:** We have built our own flight controller using the Multiwii open source flight controller platform.

● **Transmitter and receiver :** Radio transmitters and receivers are electronic devices; that manipulate electricity resulting in the transmission of useful information through the atmosphere or space. The transmitter sends a signal over a frequency to the
receiver. The transmitter has a power source that provides the power for the controls and transmission of the signal.

6. Drone Transmitter

A transmitter is an electronic device used in telecommunications to produce radio waves in order to transmit or send data with the aid of an antenna. The transmitter is able to generate a radio frequency alternating current that is then applied to the antenna, which, in turn, radiates this as radio waves. There are many types of transmitters depending on the standard being used and the type of device; for example, many modern devices that have communication capabilities have transmitters such as Wi-Fi, Bluetooth, NFC and cellular.

A transmitter is also known as a radio transmitter.

We have built our own transmitter using the nrf24l01 wireless radio module.

Component used:

- NRF24L01 + PA + LNA
- Toggle Switch x 2
- Joystick x 2
- Arduino Uno
- Li Ion cell 3.7v
- Boost Converter
- On/Off Switch
- Led Indicator
- TP4056 1A Li-Ion Lithium Battery charging Module

i. **Transmitter**

FlySky CT6B 2.4Ghz 6 Channel Transmitter and Receiver(FS-R6B) Remote is the popular 6 Channel Radio CT6B manufactured by FlySky. CT6B FLYSKY 2.4GHZ 6CH TRANSMITTER is an entry level 2.4 GHz radio system offering the reliability of 2.4 GHz signal technology and a receiver with 6 channels. CT6B FLYSKY 2.4GHZ 6CH TRANSMITTER radio is a value for money, entry level 6 channel transmitter, ideal for quadcopters and multicopters that require 6 channel operation.

This radio has very lightweight and handy design with two retract switches and proportional flap dials in easy reach for channels 5 and 6. It can be powered by 8 x AA Size Batteries or a 12V Power Supply. It comes with a trainer port to help beginners learn flying.

This remote is comes with FS-R6B receiver which is one of the best receiver we had in the class in very reasonable cost. we received many happy and satisfactory feedback from our hobbyist buyers for the same.

It can be configured by connecting it to the computer. Use the T6 config software to configure your radio on a computer.

7. FLIGHT CONTROLLER

Every flying drone must have an impact system. This electronic system allows a drone to be stable within the air while flying and processes all the shifts and changes in direction and therefore the wind.

The flight controller is the nerve center of the drone. Drone flight control systems are many and varied. Flown from GPS enabled autopilot systems via a two way telemetry link to basic stabilization systems using hobby grade radio control hardware, there is an open source project for you.

Modern drone flight controllers can trace their roots back to R/C helicopters. Historically, R/C aircraft were controlled directly by the pilot's radio. Modern drone flight controllers can trace their roots back to R/C helicopters. Historically, R/C aircraft were controlled directly by the pilot's radio. Helicopters added a new wrinkle to the mix: tail rotors. Helicopters use their tail (or anti-torque) rotor to counteract the torque of the main rotor which attempts to spin the entire helicopter body. This all works great when the helicopter is hovering, but what happens when the pilot throttles to fly? As the pilot throttles, the torque increases, causing the entire helicopter to pirouette a second or two, until the torque runs out again. The impact has caused more than one beginner pilot to come face-to-face with his R/C helicopter.

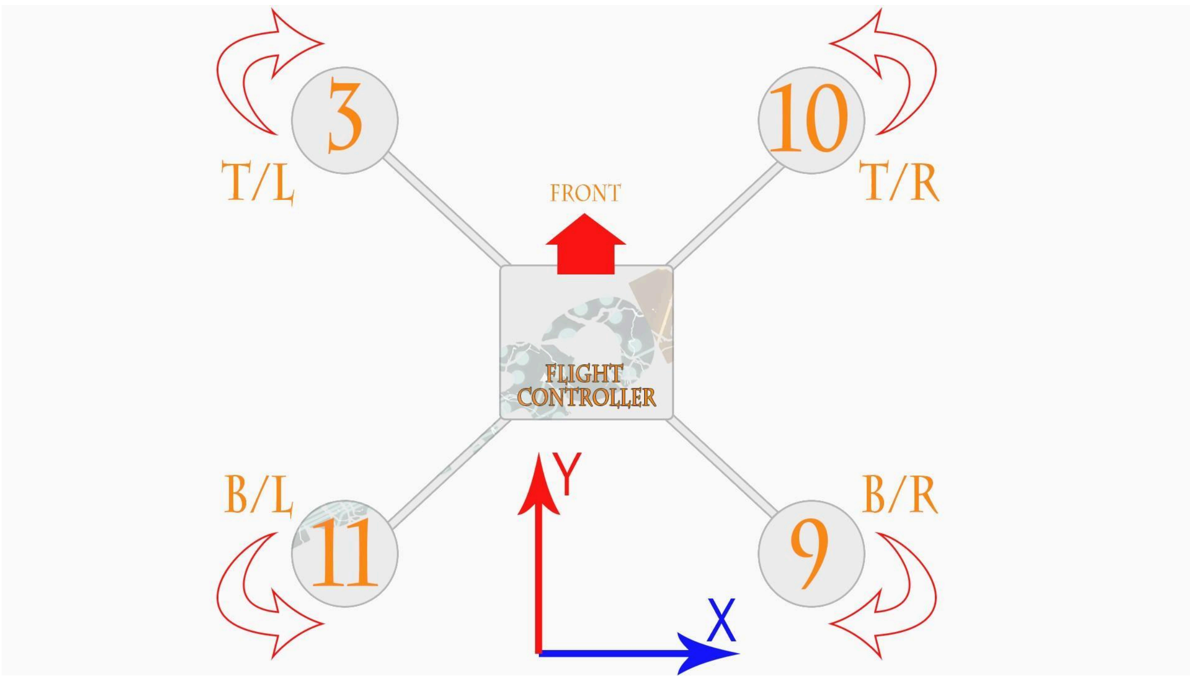
The solution to this problem was the gyroscope, a heavy brass spinning weight that tilted in response to the helicopter's motion. A Hall Effect sensor would detect that tilt and command the tail rotor to counteract the helicopter's rotation. As the years passed, mechanical gyros were replaced by solid state MEMS gyros. Microcontrollers entered the picture and brought with them advanced processing techniques. Heading hold gyros were then introduced. Whereas the older "only rate" gyros would drift, weathervane, and wiggle, the heading hold gyros would close the helicopter's nose until the pilot ordered a turn. These single axle flight controllers were quickly adopted by the R/C helicopter community.

Today’s flight control systems have many sensors available to them – GPS, barometric pressure sensors, airspeed sensors, the list goes on. The major contributors to the flight calculations are still the gyros, coupled with accelerometers. As the name implies, accelerometers measure acceleration – be it due to gravity, a high G turn, or stopping force. Accelerometers aren’t enough though – An accelerometer in free fall will measure 0 G’s. Turning forces will confuse a system trying to operate solely on accelerometer data. That’s where gyros come in. Gyros measure the rate of rotation about an axis. Just as our helicopter example above covered yaw, gyros can be used to measure pitch and roll of an aircraft.

i. So why do we need a gyro?

By the way, the drone itself will not remain upright and it will be impossible to fly it without a flight controller. A gyro module will detect any motion on each of the 3 axes. Usually it passes the rotation/second value back to the microcontroller. Using those values we can calculate the angle of the drone at each moment. If we know the angle we can always power the motors differently to stabilize and counter the unwanted forces.

We have made a quad-X shaped drone, which means it has 4 motors placed in the shape of a crucifix. For that we have to define the front part of the plane. We will have two motors at the front and two at the back as we can see in the picture below.

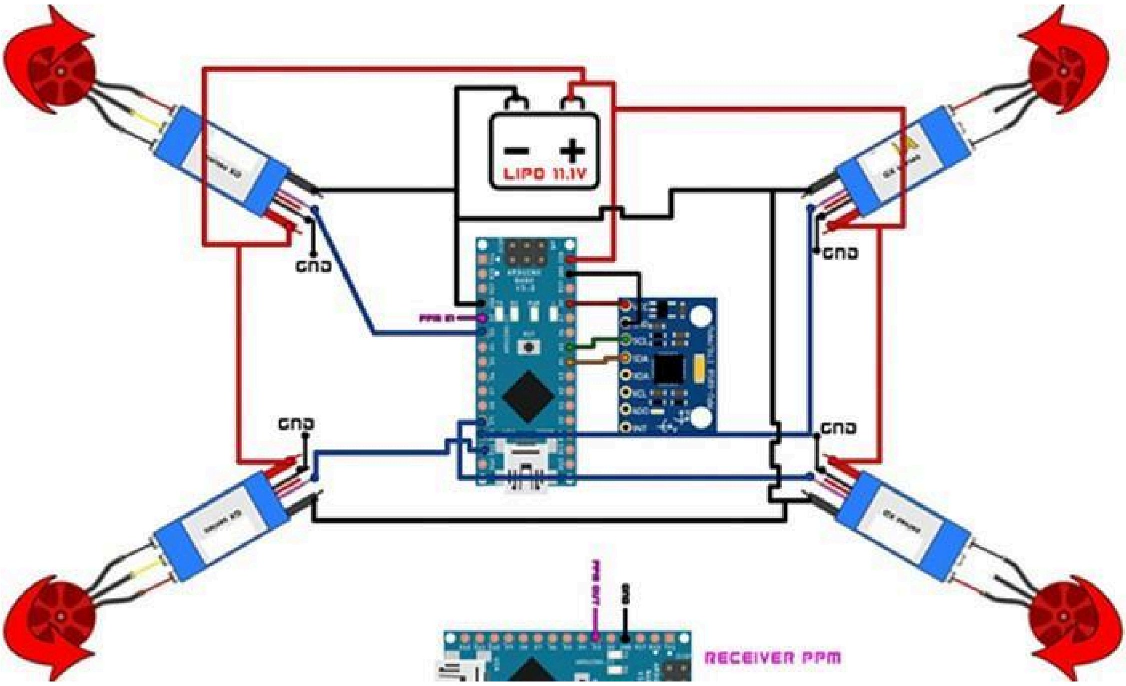


Now that we have defined the front of our drone, we can set its axis. Left to right will be the "X" axis and top to bottom will be the "Y" axis. We have to center the flight controller as much as he can. It will detect the high speed of the drone and adjust the PWM signal for each of the four motors to stabilize the drone.

We can see that each motor has a number. That number corresponds to the digital pin of the arduino. Each motor has a signal pin for the ESC. We can also see that each motor spins in a different way to create a particular air vortex. Both the rear and front pair of soft motors swivel towards the interior of the drone.

ii. SCHEMATIC OF FLIGHT CONTROLLER WITH PPM INPUT

With this configuration our drone will be able to fly without any problem. We can add more modules like magnetometer or barometer but with gyro and accelerometer it is fine for now. So once again we use the lipo battery of the drone to supply the flight controller. Connect both ground and 11.V directly to the Vin pin. Arduino Nano already has 5 and 3.3 voltage regulators. We connect pins D3, D9, D10 and D11 to each of the four motors ESCs. Each esc also needs to share ground with the arduino to understand the pwm signals.



Also supply 5 volts to the MPU6050 module and ground. Connect the SDA (A4) and SCL (A5) pins to the Arduino analog pins A4 and A5 and we are done. Also remember to share the ground between the flight controller and the radio receiver. You can see that the IMU MPU6050 also has "X" and "Y" axes. Center the module as it can on the drone and respect those pivots.

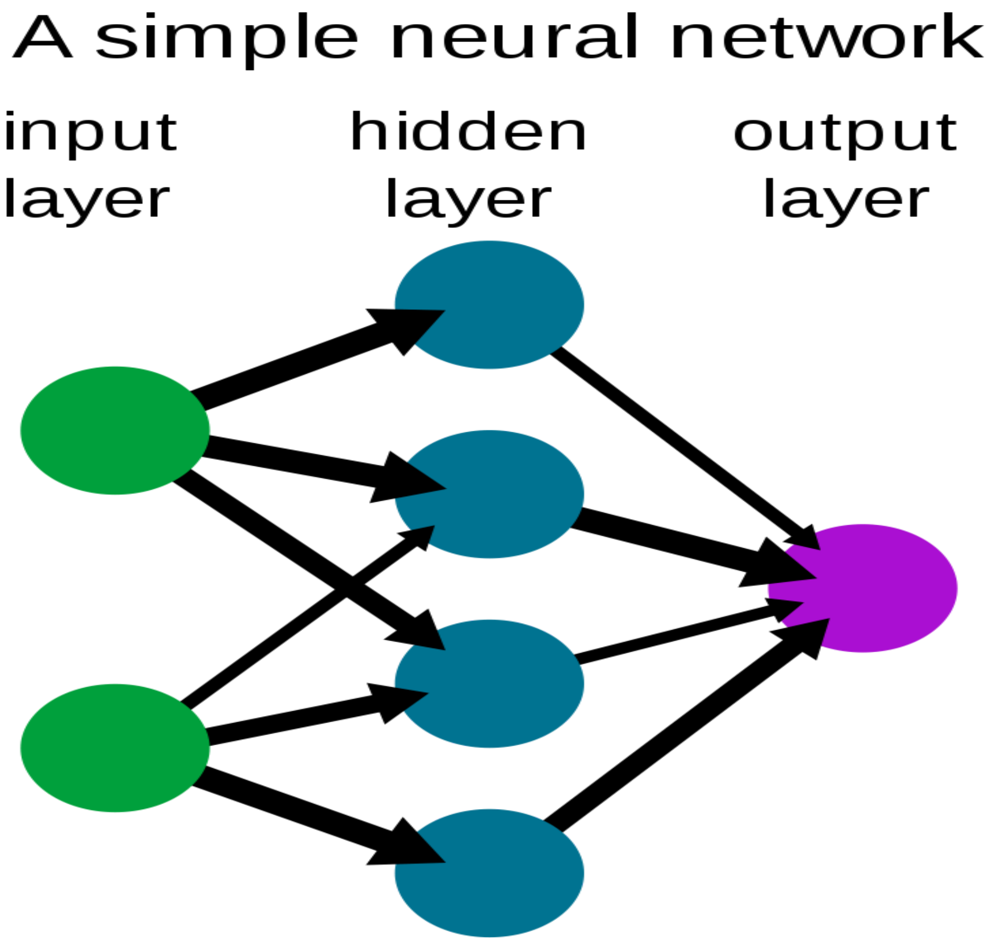
Flight Controller Firmware

We used multiwii as a flight controller firmware.

MultiWii is an open source software project aiming to provide the brain of a RC controlled multi rotor flying platform. It is compatible with several hardware boards and sensors.

Neural Network

A neural network is a network or circuit of biological neurons, or in a modern sense, an artificial neural network, composed of artificial neurons or nodes. Thus a neural network is either a biological neural network, made up of biological neurons, or an artificial neural network, for solving artificial intelligence (AI) problems. The connections of the biological neuron are modeled in artificial neural networks as weights between nodes. A positive weight reflects an excitatory connection, while negative values mean inhibitory connections. All inputs are modified by a weight and summed. This activity is referred to as a linear combination. Finally, an activation function controls the amplitude of the output. For example, an acceptable range of output is usually between 0 and 1, or it could be −1 and 1.



These artificial networks may be used for predictive modeling, adaptive control and applications where they can be trained via a dataset. Self-learning resulting from experience can occur within networks, which can derive conclusions from a complex and seemingly unrelated set of information

Overview

A biological neural network is composed of a groups of chemically connected or functionally associated neurons. A single neuron may be connected to many other neurons and the total number of neurons and connections in a network may be extensive. Connections, called synapses, are usually formed from axons to dendrites, though dendrodendritic synapses[3] and other connections are possible. Apart from the electrical signaling, there are other forms of signaling that arise from neurotransmitter diffusion.

Artificial intelligence, cognitive modeling, and neural networks are information processing paradigms inspired by the way biological neural systems process data. Artificial intelligence and cognitive modeling try to simulate some properties of biological neural networks. In the artificial intelligence field, artificial neural networks have been applied successfully to speech recognition, image analysis and adaptive control, in order to construct software agents (in computer and video games) or autonomous robots.

Historically, digital computers evolved from the von Neumann model, and operate via the execution of explicit instructions via access to memory by a number of processors. On the other hand, the origins of neural networks are based on efforts to model information processing in biological systems. Unlike the von Neumann model, neural network computing does not separate memory and processing.

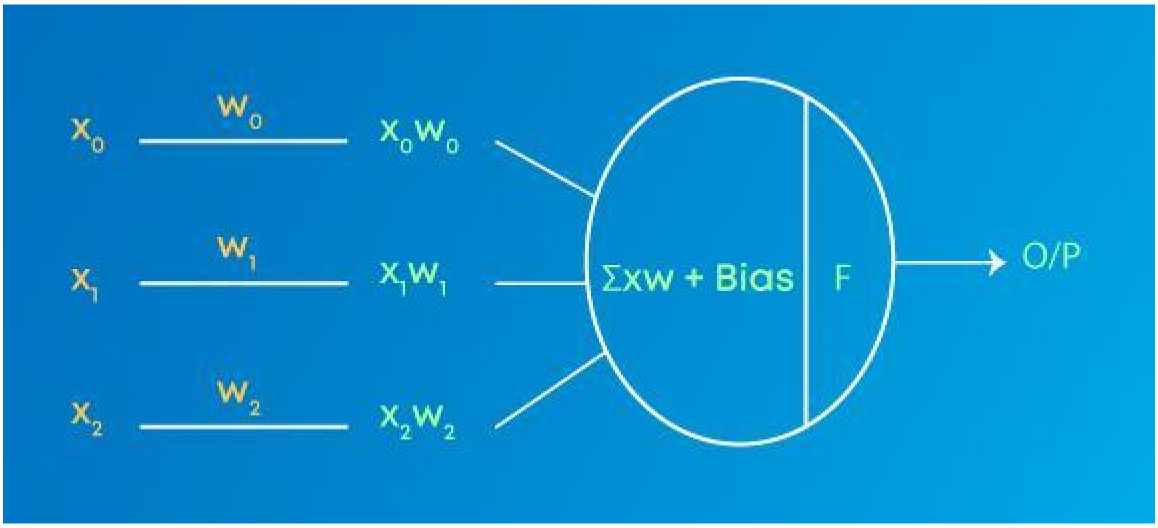
Neural network theory has served both to better identify how the neurons in the brain function and to provide the basis for efforts to create artificial intelligence.

An Introduction to Artificial Neural Network

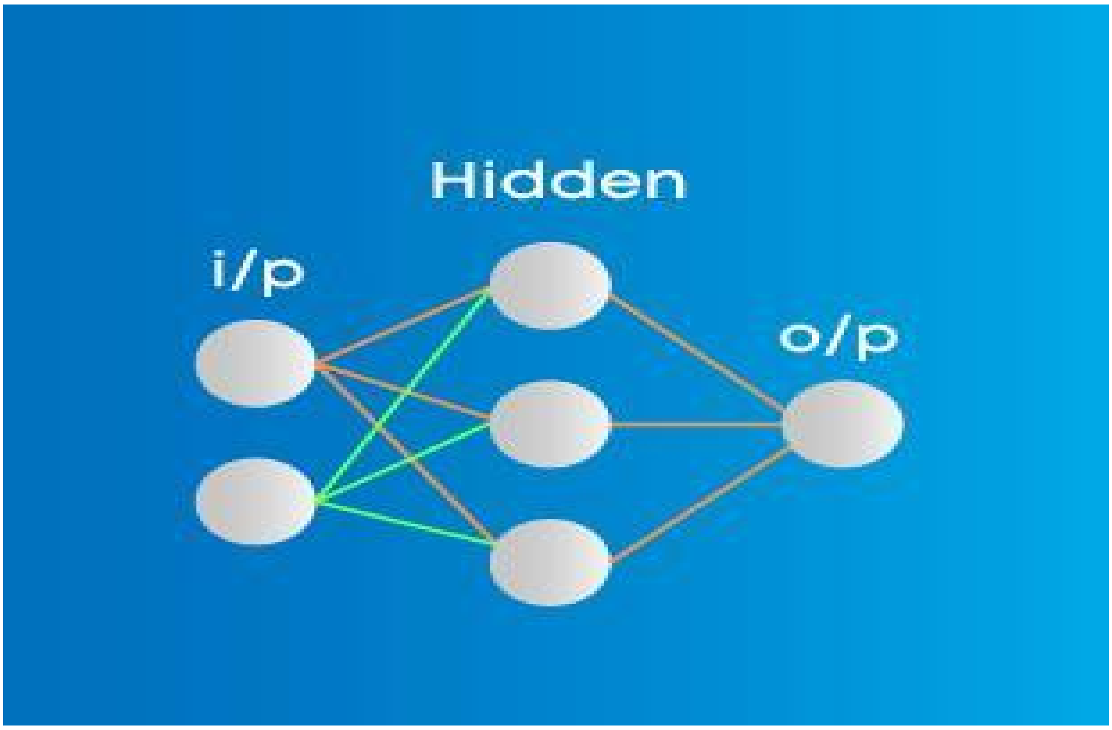
Neural networks represent deep learning using artificial intelligence. Certain application scenarios are too heavy or out of scope for traditional machine learning algorithms to handle. As they are commonly known, Neural Network pitches in such scenarios and fills the gap. Also, enrol for neural networks and deep learning course

Artificial neural networks are inspired from the biological neurons within the human body which activate under certain circumstances resulting in a related action performed by the body in response. Artificial neural nets consist of various layers of interconnected artificial neurons powered by activation functions which help in switching them ON/OFF. Like traditional machine algorithms, here too, there are certain values that neural nets learn in the training phase.

Briefly, each neuron receives a multiplied version of inputs and random weights which is then added with static bias value (unique to each neuron layer), this is then passed to an appropriate activation function which decides the final value to be given out of the neuron. There are various activation functions available as per the nature of input values. Once the output is generated from the final neural net layer, loss function (input vs output)is calculated and backpropagation is performed where the weights are adjusted to make the loss minimum. Finding optimal values of weights is what the overall operation is focusing around. Please refer to the following for better understanding-



Weights are numeric values which are multiplied with inputs. In backpropagation, they are modified to reduce the loss. In simple words, weights are machine learnt values from Neural Networks. They self-adjust depending on the difference between predicted outputs vs training inputs. **Activation Function** is a mathematical formula which helps the neuron to switch ON/OFF.



- **Input layer** represents dimensions of the input vector.

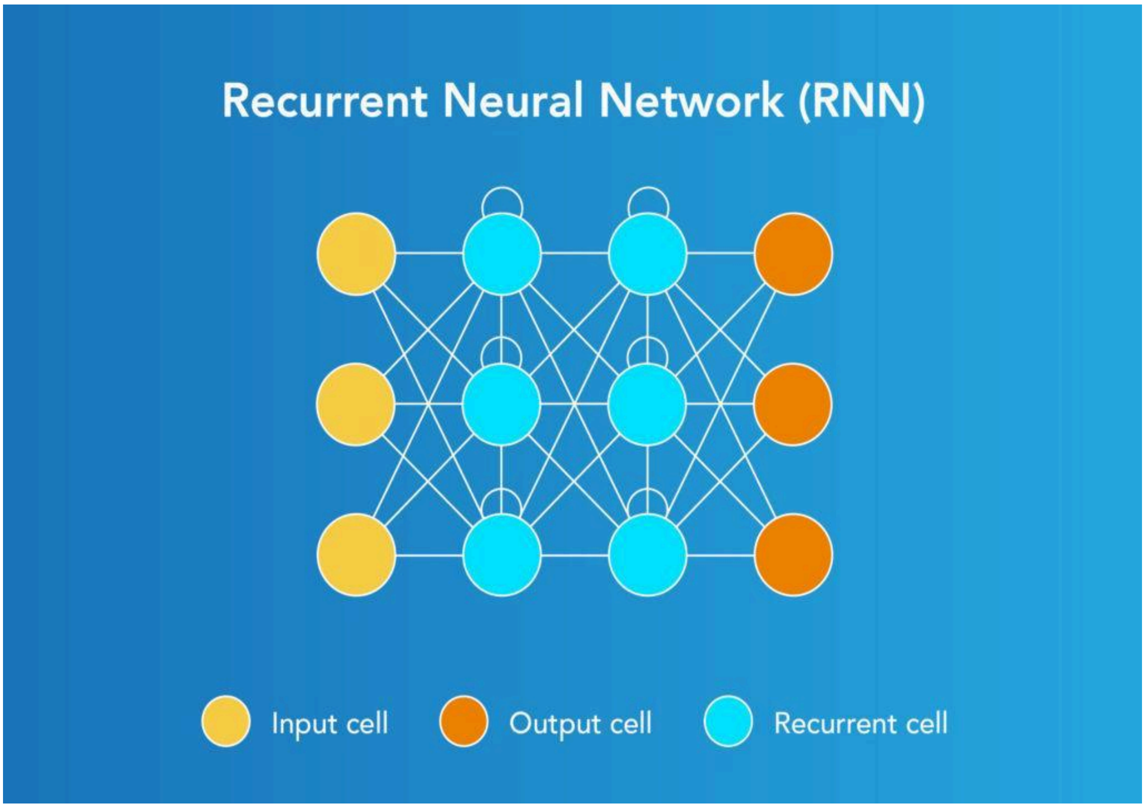
- **Hidden layer** represents the intermediary nodes that divide the input space into regions with (soft) boundaries. It takes in a set of weighted input and produces output through an activation function.
- **Output layer** represents the output of the neural network.

Types of Neural Network

There are many types of neural networks available or that might be in the development stage. They can be classified depending on their: Structure, Data flow, Neurons used and their density, Layers and their depth activation filters etc.

- Perceptron
- Feed Forward Neural Network
- Multilayer Perceptron
- Convolutional Neural Network
- Radial Basis Functional Neural Network
- Recurrent Neural Network
- LSTM – Long Short-Term Memory
- Sequence to Sequence Models
- Modular Neural Network

Recurrent Neural Network



Applications of Recurrent Neural Networks

- **Text processing like auto suggest, grammar checks, etc.**
- **Text to speech processing**
- **Image tagger**
- **Sentiment Analysis**
- **Translation**

Designed to save the output of a layer, Recurrent Neural Network is fed back to the input to help in predicting the outcome of the layer. The first layer is typically a feed forward neural network followed by recurrent neural network layer where some information it had in the previous time-step is remembered by a memory function. Forward propagation is implemented in this case. It stores information required for it's future use. If the prediction is wrong, the learning rate is employed to make small changes. Hence, making it gradually increase towards making the right prediction during the backpropagation.

Advantages of Recurrent Neural Networks

1. Model sequential data where each sample can be assumed to be dependent on historical ones is one of the advantage.

2. Used with convolution layers to extend the pixel effectiveness.

Disadvantages of Recurrent Neural Networks

1. Gradient vanishing and exploding problems
2. Training recurrent neural nets could be a difficult task
3. Difficult to process long sequential data using ReLU as an activation function.

Applications of Neural Network

- Real time translation (Google translation)
- Radars
- Facial recognition
- Medical Science
- Automated driving vehicles
- Handwriting recognition (pen to print)

Advantages of Neural Network

- A neural network can perform tasks that a linear program can not.
- When an element of the neural network fails,it can continue without any problem by their parallel nature.
- A neural network learns and does not need to be reprogrammed.
- It can be implemented in any application.
- It can be implemented without any problem

Crowd Detection Code for Python

```
import cv2
import numpy as np
import time
import requests
```

```
from urllib.request import urlretrieve
from datetime import datetime

print("Welcome to drone project")

while True:

    print("Please select option \n press 1. for capture photo \n
press              2.for detected photo \n press 3. Exit")
    cap = int(input("Enter your choice: "))

    if cap == 1:

        now = datetime.now()
        now = str(now).replace(' ', '_')
        now = now.replace('.', '_')
        now = now.replace(':', '_')
        now = now.replace('-', '_')

        requests.get('http://192.168.84.253/capture')
        print('Wait Capturing Photo')
        time.sleep(10)
        print('Photo Captured')
        photo_name = now+'.jpg'

urlretrieve('http://192.168.84.253/saved-photo','captured/'+
photo_name)
    elif cap == 2:
        CONFIDENCE = 0.5
        SCORE_THRESHOLD = 0.5
        IOU_THRESHOLD = 0.5
        config_path = "darknet-master/cfg/yolov3.cfg"
        weights_path = "yolov3.weights"
```

```
        labels = open("darknet-master/data/coco.names").read()
.strip().split("\n")

        # generating colors for each object for later plotting
        colors = np.random.randint(0, 255, size=(len(labels), 3),
dtype="uint8")

        # load the YOLO network
        net = cv2.dnn.readNetFromDarknet(config_path,
weights_path)
        image = cv2.imread('captured/'+photo_name)
        h, w = image.shape[:2]

        # create 4D blob
        blob = cv2.dnn.blobFromImage(image, 1 / 255.0, (416, 416),
swapRB=True, crop=False)  # mean,size

        # sets the blob as the input of the network
        net.setInput(blob)

        # get all the layer names
        ln = net.getLayerNames()
        ln = [ln[i - 1] for i in net.getUnconnectedOutLayers()]

        # feed forward (inference) and get the network output
        # measure how much it took in seconds
        start = time.perf_counter()
        layer_outputs = net.forward(ln)
        time_took = time.perf_counter() - start
        print(f"Time took: {time_took:.2f}s")
        boxes, confidences, class_ids = [], [], []

        # loop over each of the layer outputs
        for output in layer_outputs:
            # loop over each of the object detections
            for detection in output:
                # extract the class id (label) and confidence (as
a probability) of
```

```
                # the current object detection
                scores = detection[5:]
                class_id = np.argmax(scores)
                confidence = scores[class_id]

                # discard weak predictions by ensuring the
detected
                # probability is greater than the minimum
probability
                if confidence > CONFIDENCE:
                    box = detection[:4] * np.array([w, h, w, h])
                    (centerX,centerY,width,height)
=box.astype("int")

                    # use the center(x,y)-coordinates to derive
the top and
                    # and left corner of the bounding box
                    x = int(centerX - (width / 2))
                    y = int(centerY - (height / 2))

                    # update our list of bounding box coordinates,
confidences,
                    # and class IDs
                    boxes.append([x, y, int(width), int(height)])
                    confidences.append(float(confidence))
                    class_ids.append(class_id)

        # perform the non maximum suppression given the scores
defined before
        idxs = cv2.dnn.NMSBoxes(boxes, confidences,
SCORE_THRESHOLD, IOU_THRESHOLD)

        font_scale = 1
        thickness = 1
        j = 0

        # ensure at least one detection exists
        if len(idxs) > 0:
            # loop over the indexes we are keeping
```

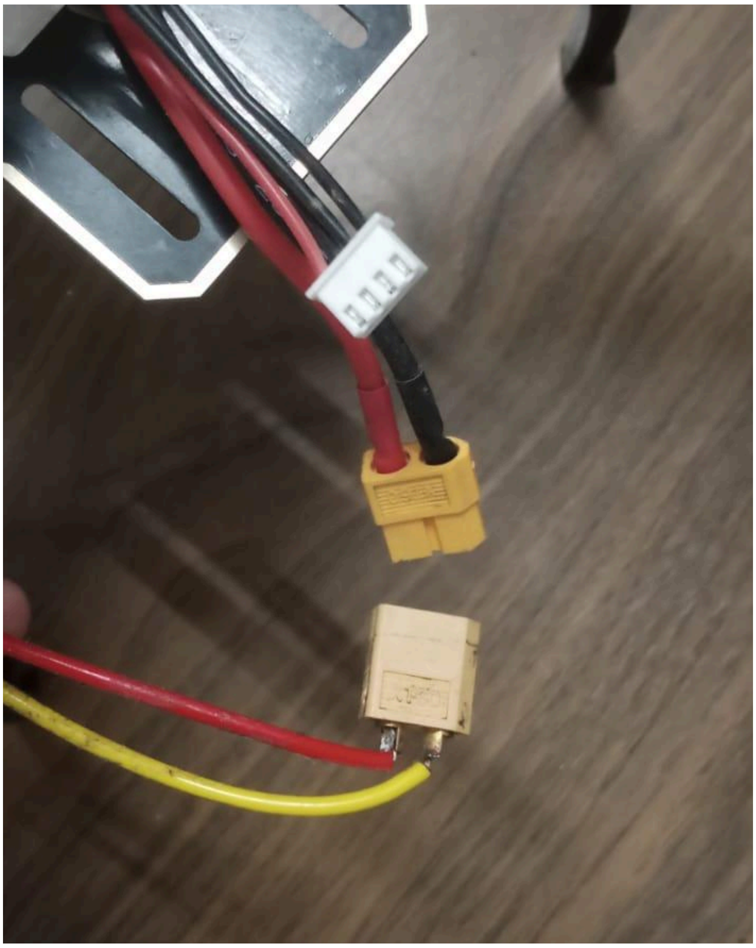
```
            for i in idxs.flatten():

                # extract the bounding box coordinates
                x, y = boxes[i][0], boxes[i][1]
                w, h = boxes[i][2], boxes[i][3]

                # draw bounding box rectangle and label on the
image
                color = [int(c) for c in colors[class_ids[i]]]
                cv2.rectangle(image,(x, y),(x + w, y +
h),color=color, thickness=thickness)
                text =f"{labels[class_ids[i]]}:
{confidences[i]:.2f}"

                # calculate text width & height to draw
transparent boxes as background of the text
                (text_width, text_height) = \
                    cv2.getTextSize(text,
cv2.FONT_HERSHEY_SIMPLEX, fontScale=font_scale,
thickness=thickness)[0]
                text_offset_x = x
                text_offset_y = y - 5
                box_coords = (
                (text_offset_x, text_offset_y), (text_offset_x +
text_width + 2, text_offset_y - text_height))
                overlay = image.copy()
                cv2.rectangle(overlay, box_coords[0],
box_coords[1], color=color, thickness=cv2.FILLED)

                # add opacity (transparency to the box)
                image = cv2.addWeighted(overlay, 0.6, image, 0.4,
0)

                # now put the text (label: confidence %)
                cv2.putText(image, text, (x, y - 5),
cv2.FONT_HERSHEY_SIMPLEX,
                            fontScale=font_scale, color=(0, 0, 0),
thickness=thickness)
```

```
                # person counting
                for k in [labels[class_ids[i]]]:
                    if k == labels[class_ids[1]]:
                        j += 1
        print(f"Detected crowed : {j}")

        scale_percent = 60  # percent of original size
        width = int(image.shape[1] * scale_percent / 100)
        height = int(image.shape[0] * scale_percent / 100)
        dim = (width, height)

        resized = cv2.resize(image,dim,
interpolation=cv2.INTER_AREA)

        (cX, cY) = (width // 2, height // 2)
        M = cv2.getRotationMatrix2D((cX, cY), 90, 1.0)
        rotated = cv2.warpAffine(resized, M, (w, h))


        # resize image

        cv2.imshow(f'Detected Crowd :{j}', rotated)
        cv2.imwrite('detected/'+photo_name, rotated)
        if cv2.waitKey(0) == ord("q"):
            pass


    elif cap==3:
        break
    else:
        print("Invalid choice")
        break
```

Transmitter Controls

Procedure for Operating Drone

1: Place the drone on plane surface
2: Connect the Battery XT6 male Socket with the Drone XT6 Female socket.



3: Power on the transmitter

4: To arm the drone bring throttle to 0 and yaw to maximum.

4: To Disarm the drone, bring yaw and throttle to minimum.

Procedure For Connecting Drone Camera:

1: First make a wifi hotspot with name “dronecam” and Password “dronecame”
2: Drone Camera will automatically connect with this hotspot.
3: Connect your laptop to the same hotspot.
4: Now your Camera is ready to capture image.

Procedure for Capturing Image and Crowd Detection:
1: After connecting drone camera and laptop to the hotspot we are ready detect crowd and capture picture.

Final Layout of Drone

8. Result

1. Drone frames and parts have been successfully assembled.

2. Transmitter with a charging system is mounted in a wooden box.
3. The receiver is made on the PCB.
4. The flight controller is built on a PCB and is calibrated according to the structure of the frame.
5. Successfully established connection between Transmitter and receiver.
6. Transmitter and receiver work well in the 500m range.
7. We have successfully flight tested our drone in an open area with good results.
8. The quadcopter was ready to fly over 15 km/hr. meeting the prototype requirement, but the utmost speed wasn’t tested out of concern for a crash that would occur during this test.

9. Conclusion

According to the proposed plan, the outcome of this semester proposed work results in the event of a quadcopter that features a stable flight. This project is implemented using arduino, a frame where everything is mounted, motors, and propellers for the movement of the quadcopter, ESC to regulate the motors. This results in a really stable flight. The entire system helps in various applications like surveillance and rescue missions. Longer flight times are often achieved by adjusting the trade-off between two variables, the battery capacity (weight) the efficiency of the thrust developed by the motors. The efficiency of the thrust has two factors which are the efficiency of the motor itself and therefore the propeller design.

Now we will implement the anti-riot capabilities to this drone and this will be the proposed work for the next semester