AUTONOMOUS DRONE FOR DEFENCE MACHINERY MAINTENANCE AND SURVEILLANCE

A PROJECT REPORT

Submitted by

RAM V SHRIVATSAV (212715106126)

SANDEEP KUMAR R (212715106140)

P R SRIRAM (212715106157)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

SRI VENKATESWARA COLLEGE OF ENGINEERING

ANNA UNIVERSITY:: CHENNAI 600 025

MARCH 2019

ANNA UNIVERSITY, CHENNAI

BONAFIDE CERTIFICATE

Certified that this project report **"AUTONOMOUS DRONE FOR DEFENCE MACHINERY MAINTENANCE AND SURVEILLANCE"** is the bonafide work of **"RAM V SHRIVATSAV (212715106126), SANDEEP KUMAR R(212715106140)** and **P R SRIRAM(212715106157)"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. S. Muthukumar	Dr. S. Muthukumar
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Professor	Professor
Electronics & Communication Engineering	Electronics & Communication Engineering
Sri Venkateswara College of Engineering	Sri Venkateswara College of Engineering
Pennalur,	Pennalur,
Sriperumbudur – 602 117	Sriperumbudur – 602 117

Submitted to the Project viva-voce Examination held on …………………………

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We would like to thank our principal, **Prof. Dr. S. GANESH VAIDYANATHAN**, for providing us the necessary infrastructure and constant support to complete our project.

We thank **Dr. S. MUTHUKUMAR**, Head of the Department, Department of Electronics and Communication Engineering for his continuous support in the completion of the project.

We wish to thank the project coordinators, **Mrs. S RADHIKA, Mr. S P SIVAGNANA SUBRAMANIAN** and **Mrs. K S SUBHASHINI** for encouraging us throughout our project.

We are extremely grateful to **Mr. VIJAY BABU** Senior Engineer at STARC Defence Research and Development Organization for guiding us throughout the project work. We sincerely express our gratitude to him.

We also extend our sincere thanks to all teaching and non-teaching staff of the Department of Electronics and Communication Engineering for their valuable support.

ABSTRACT

The impact the drone has created on human life has been extraordinary. Vehicles have become smart and autonomous which paves the way to a wide range of applications in space and military vehicles. This proposed research work focuses on the implementation of an autonomous unmanned aerial vehicle which is capable of navigating autonomously without any real-time input from the user. Surveillance and Machinery maintenance are the primary applications that are being focussed in this project which is being used in Defence purposes in the Line of Control and Warzones. Surveillance is done through a high definition GoPro camera which takes images for every two seconds. On arrival at the base station, all the images are transferred to the local server, which then passes these test images to a pretrained object detection Yolov3 model to predict the movements of human and abnormal objects. The machinery maintenance needs a lot of manpower and the autonomous drone handles this with ease as it has an onboard CC2500 RF wireless communication module which communicates with the raspberry pi placed on the machinery which in turn reads out data about its environmental conditions such as temperature, pressure, humidity and accelerometer data. We use machine learning techniques to find patterns in the given data using Random forest classifier to say if the working condition of the machine is safe, moderate or at risk based on the machine's environmental data.

LIST OF FIGURES

LIST OF ABBREVIATIONS

API	Application Programming Interface
CNN	Convolution Neural Network
CPU	Central Processing Unit
DL	Deep Learning
DNN	Deep Neural Network
ARM	Advanced Risc Machines
GPIO	General Purpose Input Output
RAM	Random Access Memory
SOC	System on Chip
V	Volt
A	Ampere
CV	Computer Vision
SSH	Secure Shell Computing
IC	Integrated Circuit
GND	Ground
Vcc	Voltage Collector-to-Collector
py	Python
Vss	Voltage Source-to-Source
CLI	Command Line Interface
YOLO	You only look once

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 UNMANNED AERIAL VEHICLE

The concept of the unmanned aerial vehicle has been on the rise over the past few years. Having a human commandeer an aerial vehicle especially in areas like war zones and places struck by natural calamities can be a very challenging and a hazardous task. By deploying a drone, the task can be carried out with utmost precision and with no human collateral damage. The drones can be used for a wide range of applications ranging from defence to agriculture.

1.2 AUTONOMY OF THE DRONE

One of the key aspects of the drone is its autonomy. The unmanned aerial vehicle is capable of navigating autonomously without any real-time input from the user and also is programmed to follow a specified path. The algorithm enables a control technique by which the quadcopter is facilitated to fly autonomously. The trajectory of the drone, accurate altitude hold performance and its smooth motion can all be monitored autonomously.

1.3 GENERAL ELEMENTS

Although the possibilities are endless when it comes to drone technology and surveillance, the three important elements of our project are:

- Deep Learning for object detection
- Machine learning for machinery data analysis

- Raspberry Pi for data collection from machines

1.3.1 DEEP LEARNING FOR OBJECT DETECTION

It is necessary to analyze the state of the drone's environment in order to recognize an object in a relatively new and unknown environment, that is where our deep learning algorithm is incorporated to recognize the objects. The deep learning algorithm can detect human movements, unidentified objects and other foreign bodies with the aid of data acquired from the camera which is then fed into the local host storage. The images are captured every three seconds through SJCAM 5000x elite camera which is fed to our trained model.

1.3.2 MACHINE LEARNING FOR MACHINERY DATA ANALYSIS

The machinery controller has a raspberry pi sensehat sensor module installed to keep track of the atmospheric conditions like pressure, temperature, and humidity of the machine. These data are sent to drone from machine using the CC2500 radio frequency transceiver module. The machine learning algorithm is then used to analyze the state of the machine from the collected environmental data. The algorithm used in this case is the random forest algorithm which predicts the current state of the machine to be either safe, moderate or at risk.

1.3.4 RASPBERRY PI FOR DATA COLLECTION FROM THE MACHINE

The main function we aim to accomplish using raspberry pi is the monitoring of the various atmospheric conditions around the machine as well as its orientation. The raspberry pi sensehat has various sensor packed on a single board which facilitates the measurements of temperature, humidity, pressure, and orientation. The output can be collected through the I2C protocol in the general purpose input and output pins of raspberry pi microprocessor.

1.4 INTEGRAL PARTS

The autonomous quadcopter drone is controlled using the ardupilot flight controller, which is powered by a Lithium Polymer 2200mAh battery. The body of the drone is made of lightweight S500 frame which has a carbon fiber power distribution board which is used to ensure equal distribution of power to all the four brushless DC motors. The Gyro stabilization technology is the most essential component that makes the drone to maintain a smooth flight even through strong winds and gusts. The drone also has a controller module which stores the message in EEPROM of the Arduino nano which is decoded in the base station and is further analyzed for finding any deviations from the present ranges of values. The SJCAM 5000x elite camera is used for object detection and procuring input data for deep learning.

1.5 NEED FOR DRONE

The main purpose of the drone is to minimize human collateral damage and error. In war zones and places struck by natural calamities, it can be very challenging or even impossible to have human intervention owing to the extreme and dangerous surroundings. At environments like this, it is not advisable to deploy manpower to get the job done. Therefore, to avoid any possible harm to humans and also to maintain the task error-free drones can be deployed. These drones can be quick and efficient when navigating through rough conditions and unpredictable terrain.

CHAPTER 2

LITERATURE SURVEY

Tuton Chandra Mallick, Mohammad Ariful Islam Bhuyan, Mohammed Saifuddin Munna," Design & Implementation of a UAV (Drone) with Flight Data Record [4]". This paper proposes the development of an autonomous unmanned aerial vehicle which is controlled by wireless technology through the graphical user interface. This proposed design is capable to fly autonomously and also capable to track pre-loaded mission automatically. Proposed mathematical model and artificial algorithm control technique by which quadrotor can capable to fly autonomously, trajectory tracking, graceful motion, and accurate altitude hold performance. In this system author used IMU 9DOF (3-axis accelerometer, 3-axis gyroscope & 3-axis magnetometer) which ensure its smooth movement, graceful motion, and trajectory tracking. GPS system and barometric sensor make it more efficient in autonomous mode. Several PID loops designed to get better stability and performance in a different mode. All signals are processed by a powerful high-speed controller board which makes it more efficient and effective. This work aimed to design a quadcopter that will try stable its position according to preferred altitude. Also here stability check has been done with pitch and roll. All data and result discussed at the end of the paper.

Apurv Saha, Akash Kumar, Aishwarya Kumar Sahu," FPV Drone with GPS used for Surveillance in Remote Areas [2]" Paper presents a brief idea about drones, their technology, the process to make that technology energy-efficient and can be used for different purposes. The major aim of our prototype is for surveillance of terrifying notions and hidden activities which can be captured in the camera which gives us an aerial view of objects. Our major aim is to increase the flight time of the drone as much as possible so to get more time for surveillance and to reduce the noise produced by cloaking it. An additional feature which is added to the prototype is the transmission of video in FPV goggles. Our aim is making it available at an affordable rate for the private agencies, institutions and governmental bodies for efficient surveillance. There is a lot of hidden observations and development for this prototype in the future.

Suet-Peng Yong1 and Yoon-Chow Yeong," Human Object Detection in Forest with Deep Learning based on Drone's Vision"[1]. In the past decade, various new and impressive applications have been developed and implemented on drones, for instance, search and rescue, surveillance, traffic monitoring, weather monitoring and so on. The current advances in drone technology provoked significant changes in enabling drones to perform a wide range of missions with increasing level of complexity. Missions such as search and rescue or forest surveillance require a large camera coverage and thus making drone a suitable tool to perform advanced tasks. Meanwhile, the increasing trend of deep learning applications in computer vision provides a remarkable insight into the initiative of this project. This paper presents a technique which allows detecting the existence of a human in forestry environment with human object detection algorithm using deep learning framework. The purpose of detecting human existence in the forestry area is to reduce illegal forestry activities such as illegal entry into a prohibited area and illegal logging activities. Also, the outcome of this project is expected to aggrandize the usage of drone for forest surveillance purpose to save time and cost.

CHAPTER 3

PROPOSED SYSTEM

3.1 PROPOSED SYSTEM ARCHITECTURE

The below diagram depicts the system architecture

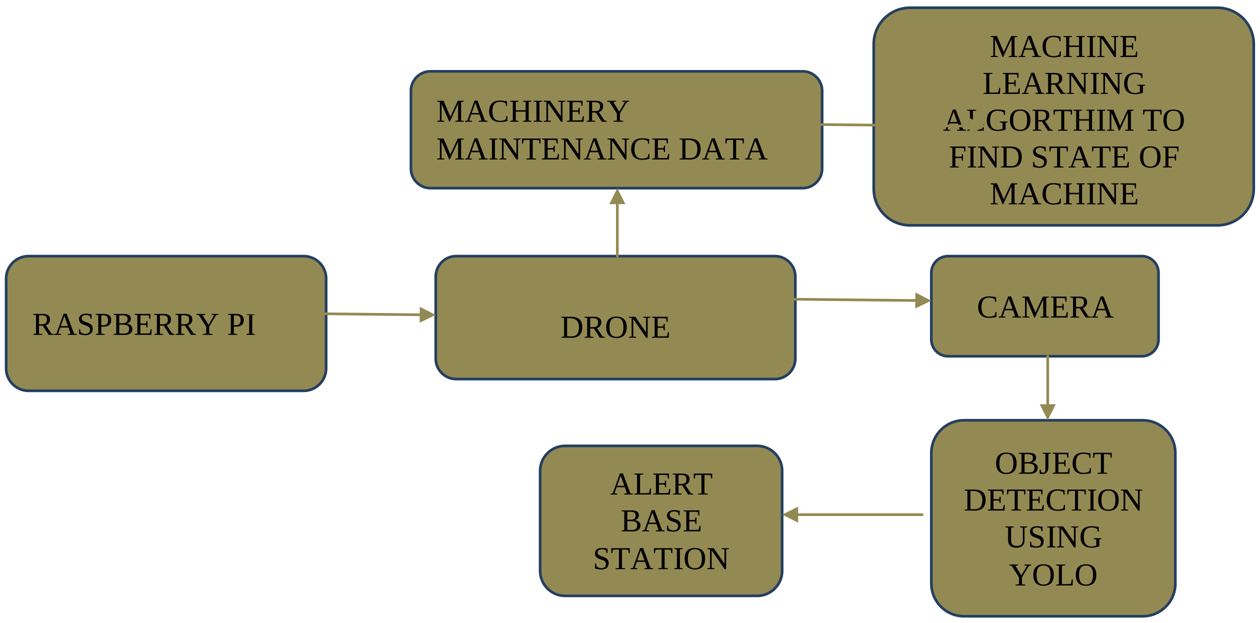

MACHINERY MAINTENANCE DATA
MACHINE LEARNING ALGORTHIM TO FIND STATE OF MACHINE
RASPBERRY PI
DRONE
CAMERA
ALERT BASE STATION
OBJECT DETECTION USING YOLO

Fig 3.1 Proposed system architecture

The drone acts as a medium to collect information for both surveillance and machinery maintenance. The raspberry pi sensors are used to collect environmental data of the machine which is sent to the drone using the CC2500 radio frequency transceiver module.

Features of the raspberry pi 3 microprocessor are:

- Quad-Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full-size HDMI
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A

The CC2500 radio frequency module is used to get transmit machinery environmental data such as temperature, pressure, humidity and accelerometer values from raspberry pi to the drone.

Features of the CC2500 Radio Frequency are:

The CC2500 is a low-cost 2.4 GHz transceiver designed for very low-power wireless applications. The circuit is intended for the 2400-2483.5 MHz Industrial, Scientific and Medical and Short Range Device frequency band. The RF transceiver is integrated with a highly configurable baseband modem. The modem supports various modulation formats and has a configurable data rate up to 500 kBaud. CC2500 provides extensive hardware support for packet handling, data buffering, burst transmissions, clear channel assessment, link quality indication, and wake-on-radio. The main operating parameters and the 64-byte transmit/receive FIFOs of CC2500 can be controlled via an SPI interface.

3.2 MODULES

The entire architecture is split into three modules:

- Drone
- Object detection during deep learning
- Raspberry pi and CC2500 Radio Frequency module

CHAPTER 4
AUTONOMOUS DRONE

4.1 INTRODUCTION

Unmanned aerial vehicles are a relatively new technology that has spread its applications recently. In this chapter, the development of these vehicles and their implementation are presented. Unmanned aerial vehicles have always been distinguished by having attributes very different in comparison to the rest of aerial vehicles. One of them is their capability to be potentially smaller than any other vehicle with the same target mission since they do not need to carry pilots on board. Because of this fact, they have been considered throughout history as an excellent opportunity to develop higher versatility and greater performance vehicles, being the military and defence industries their principal investors. We explain the structure of drone and each component of drone in detail.

4.2 WHY DRONES?

The drone has a wide range of applications that have led industries to invest in this technology. Moreover, the eco-friendly possibilities of these products have contributed to these industries to gain the sympathy of the public. Looking at the opinion of business experts, the drones market is considered a great commercial opportunity, since this market is expected to grow in the following years, especially in the civil side (a growth of 19% versus a 5% in the military side). Furthermore, these conclusions are also valid for new firms in this sector, owing to the UAV industry is still young and legacy military drone manufacturer clients do not interfere in the civilian sector. Finally, regarding the geographic zones where it is being experienced the UAV market to grow, it is remarkable that many of the new drone manufacturers are taking place outside the USA, the traditional UAV developer. According to BI studies, Canada with the Aeryon firm, China through DJI firm and Korea with Gryphon firm are examples of firm growth in this market.

4.3 OVERVIEW

The drone has several important sensors and controllers associated with it. Our drone controllers is ardupilot flight controller which sends the control signals to the motor and helps the drone to stabilizes itself through various sensor present in it. Drone's motors and propellers are the reason for thrust generations and lifting which is powered through a Lithium Polymer 2200mAh rechargeable power supply attached on the drone. The drone comes in various shapes such as quadcopter, hexacopter but our drone is a quadcopter which provides stable flight. The drone frame is made up of carbon fiber which provides strong durability even after multiple crashes.

4.4 HARDWARE

An analysis of the most usual components of multicopters has been done, in order to select the ones to use in the quadcopter.

The function of the main structure is to be the framework of the quadcopter, gathering all the components. They are made as light as possible but we also take into account the need to support the stresses generated by the aerodynamic forces produced during the aircraft flight. The usual materials used are aluminum, fiberglass, polymers, and carbon fiber.

4.4.1 MOTORS

The most widely used motor for small multicopters is the Brushless DC motor. It works by the electromotive forces generated by the permanent magnet located at the rotor and the coil arrangement at the stators. Its main advantage is that it does not produce magnetic interferences. This is particularly useful for improving the performance of the compass of the multicopter system. In addition, the higher efficiency of this kind of motors has led to being the motor choice for the quadcopter.

4.4.2 BLADES

Blades are the multicopter element in charge of producing the lifting and propelling forces. The main materials used for this component are plastics, carbon fiber, and wood. The last one has been discarded due to the large rotation momentum generated by its high weight, which would notably an act the aircraft maneuverability. Concerning plastics and carbon fiber, both have similar features but carbon fiber is a bit lighter and stronger which makes it an apt choice for our case.

4.4.3 ELECTRONIC SPEED CONTROLLERS

The aim of the Electronic Speed Controllers (ESC) is adapting the motor velocity according to the PWM (Pulse Wide Modulation) inputs that they are receiving. This is mainly done following two strategies: measuring the electromotive forces created by the movement of the magnet or using magnetic sensors based on the Hall Effect.

4.4.5 BATTERY

Multicopters commonly use LiPo battery as an energy source. The main reasons for this tendency are their lightweight, the large variety of shapes that they can adopt, their great capacities in comparison with their small size and their high discharging rates. Different capacities can be chosen for this type of batteries. Capacity is an indication of how much power can be stored by the battery, and it is measured in mAh (milliamperes per hour). For ensuring flight missions of 15 minutes, LiPo batteries of 2200mAh have been used with maximum current rating and voltage rating of 30A and 7.4V.

4.4.6 RADIO CONTROLLER

It is the device that connects the pilot with the aircraft and it is recommendable to be used in automatic flight for being able to recover the multicopter control in case it is needed. It has a minimum of four channels and it transmits at frequencies in the range of 2.4 GHz. The chosen radio control has two sticks: the left one controls the throttle or vertical acceleration and the yaw and the right one controls the roll and the pitch. The top right toggle stick is used for controlling the flight modes of the multicopter. The radio controller used can select a maximum of three modes. One of them will be reserved for AUTO mode. It will be selected during automatic mission.

4.4.7 TELEMETRY AND GPS

The function of the telemetry is proportionate to flight data at real time. The GPS is used to estimate the global position of the aircraft by measuring the relative positions with respect to several satellites. It has to be remarked that the GPS selected for this project has to include a compass since the IMU autopilot chosen does not include it.

4.5 QUADCOPTER MODEL

The mathematical model for the quadcopter is developed in this section and it is based on a series of assumptions:

1. The quadcopter is considered as a rigid body consisting of the main airframe and four arms. Thus, the links between the arms and the airframe are assumed to be ideal constraints, it means, there is no dissipation on them.
2. The arms meet at the center of the airframe at right angles.
3. The center of gravity of the quadcopter is located at the center of the airframe.
4. Blade's forces can be modeled according to Momentum Theory.

4.5.1 FRAME OF REFERENCE

Three main frames of reference are needed to describe the motion of a quadcopter.

The Earth reference frame - It is the one that is going to be considered inertial. Its origin is located at a static point of the Earth surface, is usually the home position of the aircraft. This reference frame will be notated by the subindex E: [iE; JE; kE].

The body reference frame - It is set in its own quadcopter, and it has its origin in the center of mass of it. The x-axis and the y-axis of this frame are in the direction of the arms 1 and 2 respectively, according to the number criterion as shown in Figure 4.1. The z-axis points the Earth in normal flight attitude conditions. The notation corresponding to this frame is: [iB; JB; kB].

The blade reference frame - It is the one used for describing the propellers motion. Then, four of them must be utilized, each one at the propeller of each arm. Their origin is attached to their blades center of mass.

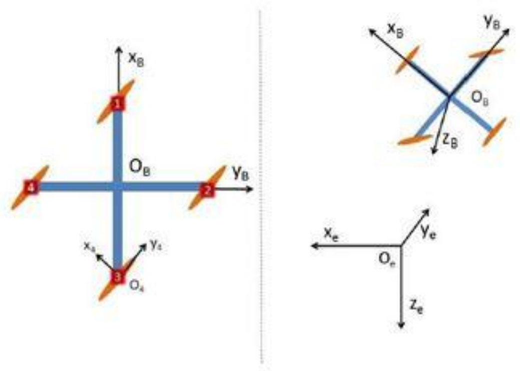


Figure 4.1: Reference frames used in the model development of the quadcopter

Regarding the blade reference frame, two of them (arms 1 and 3 according to Figure 4.1) will be rotating clockwise while the other two will be spinning in the opposite direction in order to cancel the gyroscopic effect and the aerodynamic torques in normal flight attitude.

4.5.2 MASS GEOMETRY

The mass of the quadcopter can be described as the combination of the arms mass, the board and battery masses and the ones of the blades, due to the fact that they are the components with a larger mass contribution. m0 subscript refers to the board and battery masses, which are located on the body axis origin and mb to the blades.

mq = 4 marm + m0 + 4mb (4.1)

Considering the arms and the blades as punctual masses located at the center of the blade reference frame, the inertia matrix of the quadcopter can be calculated. Moreover, due to the quadcopter symmetry, the principal axis of inertia will coincide with the body axis. Therefore, the diagonal moments of inertia of the quadcopter inertial tensor will be zero.

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} = \left(\frac{2}{3}m_{arm}l^2 + 2m_bl^2\right) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad (4.2)$$

4.5.3 KINEMATICS

For converting the vectors between the different reference frames, rotations matrices are needed. The conversion between the Earth reference frame and the Body reference frame is particularly important when developing the equations of motion.

$$\begin{bmatrix} i_B \\ j_B \\ k_B \end{bmatrix} = R_E^B \begin{bmatrix} i_E \\ j_E \\ k_E \end{bmatrix} \quad (4.3)$$

A similar procedure must be carried out for relating the Euler angles with the body rotation rates, which are the ones measured by the gyroscopes.

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} 1 & sin\phi tan\theta & cos\phi tan\theta \\ 0 & cos\phi & -sin\phi \\ 0 & sin\phi sec\theta & cos\phi sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (4.4)$$

4.5.4 FORCE MODELS

Actions affecting the quadcopter include its own weight and the aerodynamic forces produced by its blades. These aerodynamic forces and moments must be modeled using the Momentum Theory.

$$M_j = \frac{1}{2} C_D \rho A \Omega^2$$

$$T_j = C_T \rho \pi R^4 \Omega^2 \qquad (4.5)$$

ρ refers to the air density, **Ω** is the angular velocity of the blade, R the radius of the blade, A the cross-section of the blade, Tj the thrust and Mj the torque produced by the motors. Once the thrust model is established, it can be a model all the forces acting on the quadcopter.

Rolling Moments:

Body gyro effect: ˙θψ˙(Iyy − Izz)

Propeller gyro effect: Jr ˙θ**Ω**r

Roll actuators action: l(−T2 + T4)

Pitching Moments:

Body gyro effect: φ˙ψ˙(Izz − Ixx)

Propeller gyro effect: Jrφ˙**Ω**r -

Pitch actuators action: l(T1 − T3)

Yawing Moments:

Body gyro effect: φ˙ ˙θ(Ixx − Iyy)

Propeller gyro effect: JrΩr

Roll actuators action: (−1)i P4 i=1 Qi

Forces Along x Axis:

Actuator action: (sinψsinφ + cosψsinθcosφ) P4 i=1(Ti)

Forces Along y Axis:

Actuator action: (−cosψsinφ + sinψsinθcosφ) P4 i=1(Ti)

Forces Along z-Axis:

Actuator action: cosψcosφP4 i=1(Ti) - Weight: mg

Qi is the drag of blade i, l the arm length, Jr the blade moment of inertia and Ti the thrust of the rotor i

4.5.5 EQUATIONS OF MOTION

The equations of motion, according to Newton-Euler Formalism, are described with the following expression :

$$\begin{bmatrix} mJ & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{v}_B \\ \dot{\omega}_B \end{bmatrix} + \begin{bmatrix} \omega_B \wedge m v_B \\ \omega_B \wedge I\omega \end{bmatrix} = \begin{bmatrix} F_B \\ M_B \end{bmatrix} \qquad (4.6)$$

Time derivatives must be done in the inertial Earth reference frame. Nonetheless, since vectors v and ω are projected in the body reference frame, Coriolis Theorem must be used, which has been taken into account in the second matrix of the equation . Finally, after a series of algebraic operations that can be checked in, this set of equations of motion is achieved.

$$m\ddot{x} = (sin\psi sin\phi + cos\psi sin\theta cos\phi)\sum_{i=1}^{4} T_i$$

$$m\ddot{y} = (-cos\psi sin\phi + sin\psi sin\theta cos\phi)\sum_{i=1}^{4} T_i$$

$$m\ddot{z} = mg - cos\psi cos\phi\sum_{i=1}^{4}(T_i)$$

$$I_{xx}\ddot{\phi} = \dot{\theta}\dot{\psi}(I_{yy} - I_{zz}) + J_r\dot{\theta}\Omega_r + l(-T_2 + T_4)$$

$$I_{yy}\ddot{\theta} = \dot{\phi}\dot{\psi}(I_{zz} - I_{xx}) + J_r\dot{\phi}\Omega_r + l(T_1 - T_3)$$

$$I_{zz}\ddot{\psi} = \dot{\phi}\dot{\theta}(I_{xx} - I_{yy}) + J_r\Omega_r + (-1)^i\sum_{i=1}^{4} Q_i \tag{4.7}$$

4.6 ASSEMBLY

The assembly process of the used quadcopter is going to be presented. It is divided into two sections: the hardware assembling part and the wiring and connections part. Knowing the function and location of each component is useful to quickly localize errors in case of encountering a faulty piece.

The components used inside the quadcopter are the following ones:

- 4 x Quad Frame Arm
- Quad Upper plate
- 24 x M2.5*6 Mounting screws
- 16 x M3*8 Mounting screws
- x Clockwise propellers
- x Counter-clockwise propellers
- x 6mm washer
- APM 2.5. Flight Controller

- 2 x Clockwise brushless motors
- 2 x Counter-clockwise brushless motors
- Radio Receiver
- GPS receiver
- Double-sided tape
- Cushioning sponge
- 2 x Neoprene Tape
- 3.2.1 Hardware assembling

4.6.1 ESCs AND BATTERY SOLDERING

Firstly, the ESC must be welded to the lower board. The red cables must be soldered in the positive (+) plugs and the black cables in the negative (-) ones. Secondly, the battery has to be connected to the board. With this aim, it has to be hooked two cables to the battery and then plugs them in their respective positions. Again, a red cable and a black one must be used. A scheme of this installation is presented in Figure 3.2.

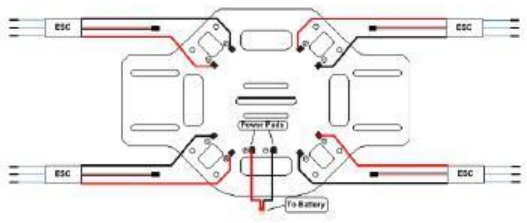


Figure 4.2: Scheme of the ESC and battery soldering

Before proceeding with this installation, it is useful to check that all ESCs work properly. It can be done by providing power to each of them and measuring their voltage. This inspection is also advisable to be done at the end of the soldering, in order to verify that the soldering has been done properly.

4.6.2 QUADCOPTER FRAME ASSEMBLY

The four arms must be attached now to the bottom frame. Four M2.5*6 Mounting screws must be screwed to the frame for each arm by using an Allen wrench. After that, the upper frame must be assembled on the other side of the arms. A picture of this final assembly is shown in the Figure below.

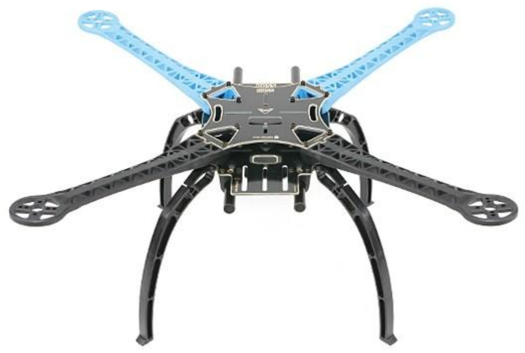


Figure 4.3: Graphical representation of the frame assembly

In this step, it is important to care about the cables of the ESCs, ensuring that they can easily pass through the armhole.

4.6.3 MOTOR ASSEMBLY

The four motors must be assembled at the extreme of the arms using the M3*6 Mounting screws. When placing each motor, it has to be taken into account the motor rotation criterion which was established after developing theoretically the quadcopter model, due to the fact that the spinning of the vehicle around itself is to be avoided. Thus, the motors with the same rotation direction (either clockwise or counter-clockwise) must be located in opposite arms, as it can be appreciated in.

Once the motors are mounted, the blades must be installed. The same criteria than the one used for the motor apply to the blades: the ones which spin clockwise must be screwed on the motors that rotate clockwise and the other ones, which spin counter-clockwise, must be screwed with their respective motor pairs. In order to avoid any kind of confusion, it is presented in Figure 4.4 the shape of both kinds of rotor blades.

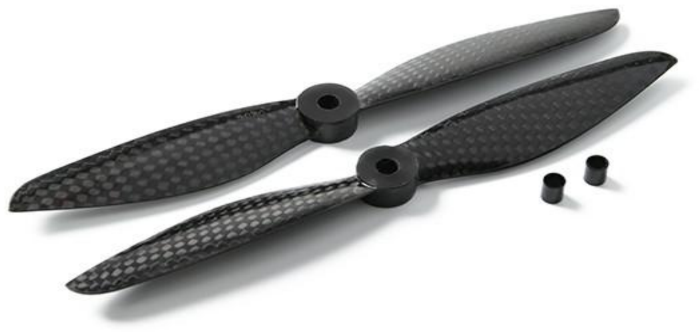


Figure 4.4: Counter-clockwise blade (left) vs Clockwise blade (right)

4.6.3 APM, radio controller, and GPS assembly

Finally, the APM, the radio controller, and the GPS must be gathered in the assembly. As vibrations an act seriously its performance, it is necessary to use some element for dampening these vibrations. Then, a cushioning sponge will be utilized with this aim for assembling the APM, neoprene tapes for the radio controller and double-sided tape for the GPS.

Once the assembly is completely finished, it is recommended to check that the structure is rigid enough.

4.6.4 WIRING AND CONNECTIONS

For the quadcopter to be ready for flight, the only remaining final step is proceeding with the wiring and the connections. The ESC has to be connected to the motor and the APM, and the GPS and the radio receiver to the APM. Each motor has to be connected with its respective ESC. Again, it has to be taken into account if the motor is spinning clockwise or counter-clockwise since the link with the ESC will be different depending on it as shown in Figure 4.5.

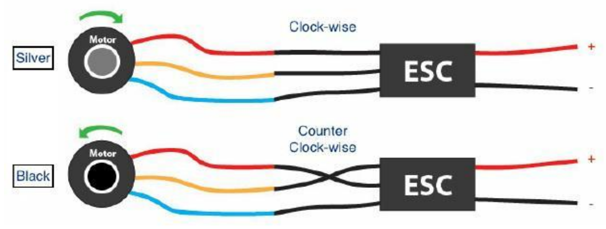


Figure 4.5: ESC connection depending on its spinning direction

Regarding the connection of the ESC with the APM, the direction of rotation of each motor plays a fundamental role. The enumeration of the motors established before (Figure 4.5) shows what is the corresponding output pin for each motor.

Concerning the GPS connection, it must be followed by two steps:

Connect the GPS magnetometer port to the I2C port of the APM. Connect the GPS port to the APM GPS port.

Lastly, the radio controller must be connected. On the one hand, the channels 1 to 6 of the receiver must be linked to the same numbers in the APM inputs pins.. On the other hand, power must be supplied. Then, one wire must connect the positive pin of the receiver with the positive one of the APM and another wire with their respective ground pins.

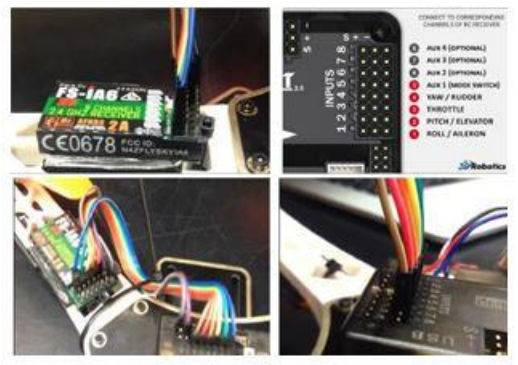


Figure 4.6: Radio receiver connection

It is important to tie all the cables to the body using zip ties, so as to avoid vibrations of the components. In this way, the quadcopter will behave closer to a rigid body.

-

4.7 QUADCOPTER PROPERTIES CHARACTERIZATION

It is necessary to adjust the parameters referring to the quadcopter, such as the blades moments and drag, or the mass and moments of inertia of the aircraft. A series of test bench is used for the experimental description of the blade's performance.

4.7.1 BLADES TEST

The test bench made for the blade characterization consists of acquiring experimentally values of the thrust and the torque generated by the blades for different rotational velocities. Then, it can be linearized with the results when comparing the torque (Mj) and the thrust (Tj) versus the rotational velocity to the power of two, obtaining the mean slope of those curves. The equations 4.6 and 4.7 are result of dividing those results between AR=2 and R4 respectively.

The outcomes of this test bench are:

CD = 3:41347   CT = 2:708042

3.3.2 Mass and moments of Inertia characterization

For obtaining the mass, each individual component of the quadcopter has been weighted. The addition of all the weights let estimate the total weight

Components	Quality	Mass(g)	Total Mass (g)
Upper frame plate	1	36.9	36.9
LOwer frame plate	1	68.6	68.6
Arms	4	49.7	198.8
APM	1	33.1	33.1
Motors	4	53.2	212.8
ESCs	4	23.3	93.2
Receiver	1	8.1	8.1
Propellers	4	7.8	31.2
GPDS	1	83.3	31.2
LiPo baterry	1	191	191

Table 4.1: Mass of the components

Then, adding all the terms:

Final mass =0.957kg

In relation to the inertia moments calculation, it has been used a complete model of the quadcopter made in Solid Edge. This program is capable of estimate the moment of inertia properties of the analyzed model when knowing the individual properties of the components of the mode

It is interesting to point out that the Solid Edge model of the quadcopter is not completely precise. Some geometry of the components has been simpli ed, as long as they have been considered of second order importance. Moreover, the wiring connections have been disregarded. The reasoning for this is the extremely lower mass of the cables respect to the rest of the quadcopter components, which makes their e ect to be negligible.

4.8 CALIBRATION

The calibration of the software of all the quadcopter components is necessary before the flight. Furthermore, the calibrated results parameters will be used also for the later simulation of the quadcopter flights. Mission Planner has been selected as the software for doing the calibration. It is going to be explained in detail the complete calibration process.

Firstly, the APM board must be connected to the computer via USB. Then, the APM must be linked to Mission Planner. After initializing Mission Planner, it is done by pressing the Connect button on the top right of the screen. For the APM to be recognized, it must be chosen before pressing the button the communication port. This can be done automatically by Mission Planner if the AUTO option is selected, or by clicking on the COM4 option and setting up the data rate to 115200. It is important to remark that Disconnect button must be used before unplug the APM. Once the APM is connected, it has to be gone to Initial Setup Wizard

4.8.1 FRAME TYPE CONFIGURATION

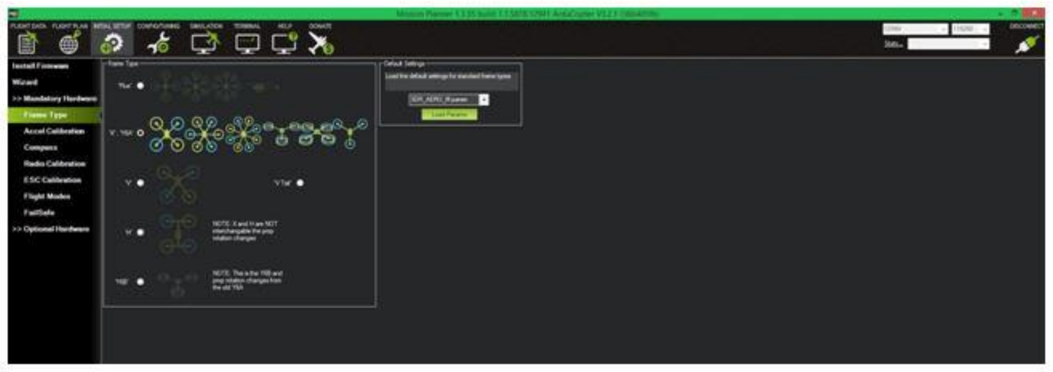


Figure 4.7: Frame type configuration

The frame of the quadcopter used must be selected. Mission Planner screen can be appreciated in Figure 4.7.

4.8.2 ACCELEROMETER CALIBRATION

Two steps must be followed:

- Calibrate Accelerometer: The autopilot will have to be placed on each edge: level, on right side, on left side, nose down, nose up and on its back.
- Calibrate level It requires placing the autopilot horizontal.

After finishing these steps, the accelerometer setting will be saved automatically. It is important to ensure that the autopilot is kept still just after pressing the key for each accelerometer calibration step.

4.8.3 RADIO CONTROL CALIBRATION

The radio controller transmitter and receiver must be calibrated now. It is needed to teach the autopilot to work with it. Select Calibrate Radio button to start the calibration process. Then, both sticks must be moved in the largest possible circle in order to be attained their complete range of motion. The range of set values will be pointed in the screen with a red line. Same procedure must be followed with channel 5 and 6 toggle sticks. Press end calibration to save the calibrated parameters. Note that if the screen bars are moving in the opposite direction to the radio control orders, it must be selected the reverse channel option on the radio transmitter

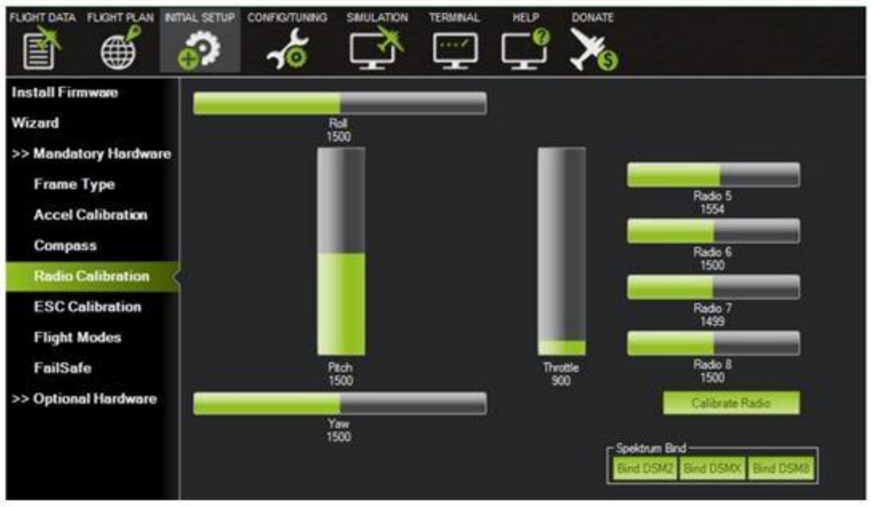


Figure 4.8: Radio control calibration

4.8.4 ESCs CALIBRATION

The minimum and maximum PWM values of the ESCs for maximum throttle radio controller commands. Before starting the ESCs calibration, the autopilot must be disconnected from the computer and the rotor blades must be disassembled. Then, it must be followed the next steps.

Turn on the Radio Controller and put the throttle stick at maximum.

Connect the LiPo battery. The autopilot's red, blue and yellow LEDs should light up in a cyclical pattern. If this condition is succeed, the APM will start in ESC calibration mode the next time the battery is plugged in.

Keeping the throttle stick high, disconnect and connect the reconnect the battery. The ESC calibration should start.

Wait for your ESCs to emit the musical tone. It depends on the battery's cell count, being that number the regular number of beeps that must be emitted (i.e. 3 for 3S, 4 for 4S). Then, additional two beeps to indicate that the maximum throttle has been captured.

Pull the throttle stick down to its minimum position

The ESCs will emit a long tone indicating that the minimum throttle has been captured. Then, the calibration is complete. Check now that the motor spins by raising the throttle a bit and then lowering it again.

Set the throttle to minimum and disconnect the battery. The APM will exit the ESC calibration mode.

4.8.5 FLIGHT MODES

Moving the toggle sticks of the radio controller it can be seen how the mode selected varies. Then, it must be selected the modes that are wanted for the missions. In the radio controller used in this project, up to three different modes can be saved. The ones chosen are the stabilize mode, for controlling manually the quadcopter through the radio controller; the guided mode, which enables the option to send commands to the quadcopter from Mission Planner; and the auto mode, for initializing automatic missions.

4.8.6 PID CALIBRATION

For calibrating the PID controllers, the mode Autotune will be used. It can be selected in the flight modes of the radio transmitter. Then, this mode must be activated once the vehicle is flying. The APM will start then to tuning the PID, lasting around 2 minutes. During this time, short corrections done manually using the radio controller sticks can be done for avoiding the aircraft to go too far.

CHAPTER 5

OBJECT DETECTION USING DEEP LEARNING

5.1 INTRODUCTION TO DEEP LEARNING

Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, semi-supervised or unsupervised.

Deep learning models are loosely related to information processing and communication patterns in a biological nervous system, such as neural coding that attempts to define a relationship between various stimuli and associated neuronal responses in the brain.

Deep learning architectures such as deep neural networks, deep belief networks and recurrent neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics and drug design, where they have produced results comparable to and in some cases superior to human experts.

5.2 DEEP LEARNING AND NEURAL NETWORKS

In machine learning, a convolutional neural network (CNN, or ConvNet) is a class of deep, feed-forward artificial neural networks that have successfully been applied to analyzing visual imagery. CNN's use a variation of multilayer perceptrons designed to require minimal pre-processing. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics.

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field.

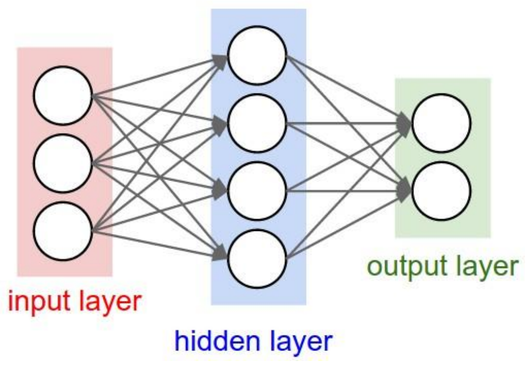


Fig 5.1: Basic CNN layout

CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage. They have applications in image and video recognition, recommender systems and natural language processing. There are four main operations in the ConvNets:

- Convolution
- Non-Linearity (ReLU)
- Pooling or Sub Sampling

- Classification (Fully Connected Layer)

These operations are the basic building blocks of *every* Convolutional Neural Network, so understanding how this work is an important step to developing a sound understanding of ConvNets.

5.2.1 CONVOLUTION

ConvNets derive their name from the “convolution” operator. The primary purpose of Convolution in case of a ConvNet is to extract features from the input image. Convolution preserves the spatial relationship between pixels by learning image features using small squares of input data. Convolution is similar to cross-correlation. For discrete real valued signals, they differ only in a time reversal in one of the signals. For continuous signals, the cross-correlation operator is the adjoint operator of the convolution operator.

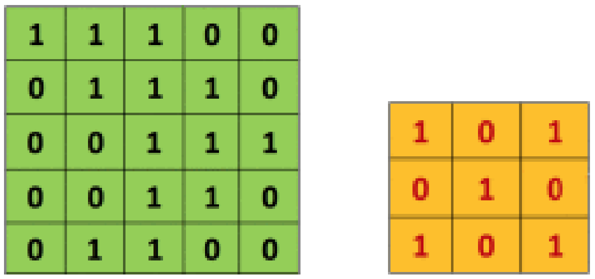


Fig 5.2: Example matrices

We slide the orange matrix over our original image (green) by 1 pixel (also called ‘stride’) and for every position, we compute element wise multiplication (between the two matrices) and add the multiplication outputs to get the final integer which forms a single element of the output matrix. Note that the 3×3 matrix “sees” only a part of the input image in each stride. In CNN terminology, the 3×3 matrix is called a ‘filter’ or ‘kernel’ or ‘feature detector’ and the matrix formed by sliding the filter over the image and computing the dot product is called the ‘Convolved Feature’ or ‘Activation Map’ or the ‘Feature Map‘. It is important to note that filters act as feature detectors from the original input image.

5.2.2 RELU

An additional operation called ReLU has been used after every Convolution operation. ReLU stands for Rectified Linear Unit and is a non-linear operation. ReLU is an element wise operation (applied per pixel) and replaces all negative pixel values in the feature map by zero. The purpose of ReLU is to introduce non-linearity in our ConvNet, since most of the real-world data we would want our ConvNet to learn would be non-linear (Convolution is a linear operation – element wise matrix multiplication and addition, so we account for non-linearity by introducing a non-linear function like ReLU). Other non-linear functions such as tanh or sigmoid can also be used instead of ReLU, but ReLU has been found to perform better in most situations.

5.2.3 POOLING

Spatial Pooling (also called subsampling or down sampling) reduces the dimensionality of each feature map but retains the most valuable information.

Spatial Pooling can be of several types: Max, Average, Sum etc. In the case of Max Pooling, we define a spatial neighborhood (for example, a 2×2 window) and take the largest element from the rectified feature map within that window. Instead of taking the largest element we could also take the average (Average Pooling) or the sum of all elements in that window. In practice, Max

Pooling has been shown to work better.

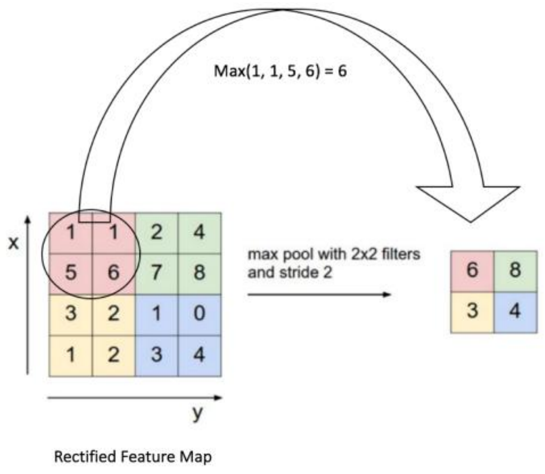


Fig 5.3: Max Pooling operation

We slide our 2 x 2 window by 2 cells (also called 'stride') and take the maximum value in each region.

5.2.4 CLASSIFICATION

The Fully Connected layer is a traditional Multi-Layer Perceptron that uses a SoftMax activation function in the output layer. The term "Fully Connected" implies that every neuron in the previous layer is connected to every neuron on the next layer. The output from the convolutional and pooling layers represent high-level features of the input image. The purpose of the Fully Connected layer is to use these features for classifying the input image into various classes based on the training dataset.

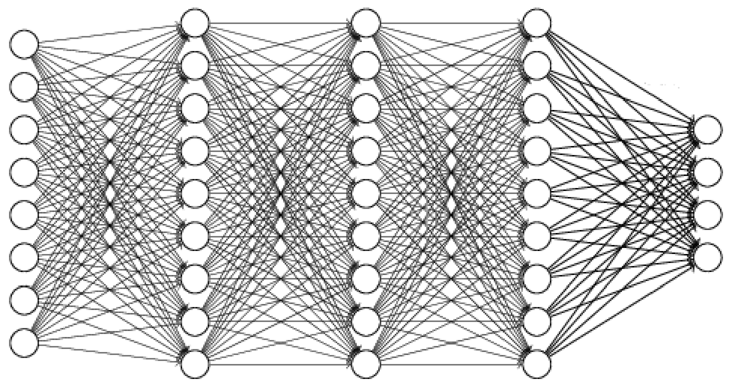


Fig 5.4: A fully connected network

Apart from classification, adding a fully-connected layer is also a way of learning non-linear combinations of these features. Most of the features from convolutional and pooling layers may be good for the classification task, but combinations of those features might be even better.

5.3 INTRODUCTION TO TENSORFLOW

TensorFlow is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows users to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API.

5.3.1 KEY FEATURES OF TENSORFLOW

- Efficiently works with mathematical expressions involving multi-dimensional arrays
- Good support of deep neural networks and machine learning concepts

- GPU/CPU computing where the same code can be executed on both architectures
- High scalability of computation across machines and huge data sets

Together, these features make TensorFlow the perfect framework for machine intelligence at a production scale.

5.4 INTRODUCTION TO YOLO OBJECT DETECTIONS

Object detection is a domain that has benefited immensely from the recent developments in deep learning. Recent years have seen people develop many algorithms for object detection, some of which include YOLO, SSD, Mask RCNN and RetinaNet [1].

YOLO stands for You Only Look Once is a famous algorithm used in object detection. It's an object detector that uses features learned by a deep convolutional neural network to detect an object [7].

5.5 ARCHITECTURE OF YOLO

YOLO makes use of only convolutional layers, making it a fully convolutional network (FCN). It has 75 convolutional layers, with skip connections and upsampling layers. No form of pooling is used, and a convolutional layer with stride 2 is used to downsample the feature maps. This helps in preventing loss of low-level features often attributed to pooling.

Being a fully connected layer, YOLO is invariant to the size of the input image. However, in practice, we might want to stick to a constant input size due to various problems that when we are implementing the algorithm.

A big one amongst these problems is that if we want to process our images in batches (images in batches can be processed in parallel by the GPU, leading to speed boosts), we need to have all images of fixed height and width. This is needed to concatenate multiple images into a large batch (concatenating many PyTorch tensors into one)

The network downsamples the image by a factor called the stride of the network. For example, if the stride of the network is 32, then an input image of size 416 x 416 will yield an output of size 13 x 13. Generally, the stride of any layer in the network is equal to the factor by which the output of the layer is smaller than the input image to the network.

Typically, (as is the case for all object detectors) the features learned by the convolutional layers are passed onto a classifier/regressor which makes the detection prediction (coordinates of the bounding boxes, the class label.. etc).

In YOLO, the prediction is done by using a convolutional layer which uses 1 x 1 convolutions.The first thing to notice is our output is a feature map. Since we have used 1 x 1 convolution, the size of the prediction map is exactly the size of the feature map before it. In YOLO v3 (and it's descendants), the way you interpret this prediction map is that each cell can predict a fixed number of bounding boxes.

5.6 IMPORTANT FEATURES OF YOLO

The three important features of YOLO are

- Detection at three scales
- Detecting Smaller objects
- More bounding boxes per images

5.6.1 DETECTION AT THREE SCALES

The newer architecture boasts of residual skip connections and upsampling. The most salient feature of v3 is that it makes detections at three different scales. YOLO is a fully convolutional network and its eventual output is generated by applying a 1 x 1 kernel on a feature map. In YOLO v3, the detection is done by applying 1 x 1 detection kernels on feature maps of three different sizes at three different places in the network.

The shape of the detection kernel is 1 x 1 x (B x (5 + C)). Here B is the number of bounding boxes a cell on the feature map can predict, “5” is for the 4 bounding box attributes and one object confidence, and C is the number of classes. In YOLO v3 trained on COCO, B = 3 and C = 80, so the kernel size is 1 x 1 x 255. The feature map produced by this kernel has identical height and width of the previous feature map and has detection attributes along the depth as described above.

Assume we have an input image of size 416 x 416.YOLO v3 makes a prediction at three scales, which are precisely given by downsampling the dimensions of the input image by 32, 16 and 8 respectively.

The first detection is made by the 82nd layer. For the first 81 layers, the image is downsampled by the network, such that the 81st layer has a stride of 32. If we have an image of 416 x 416, the resultant feature map would be of size 13 x 13. One detection is made here using the 1 x 1 detection kernel, giving us a detection feature map of 13 x 13 x 255.

Then, the feature map from layer 79 is subjected to a few convolutional layers before being up sampled by 2x to dimensions of 26 x 26. This feature map is then depth concatenated with the feature map from layer 61. Then the combined feature maps are again subjected a few 1 x 1 convolutional layers to fuse the features from the earlier layer (61). Then, the second detection is made by the 94th layer, yielding a detection feature map of 26 x 26 x 255.

A similar procedure is followed again, where the feature map from layer 91 is subjected to few convolutional layers before being depth concatenated with a feature map from layer 36. Like before, a few 1 x 1 convolutional layers follow to fuse the information from the previous layer (36). We make the final of the 3 at 106th layer, yielding feature map of size 52 x 52 x 255.

5.6.2 DETECTING SMALLER OBJECTS

Detections at different layers help address the issue of detecting small objects, a frequent complaint with YOLO v2. The upsampled layers concatenated with the previous layers help preserve the fine-grained features which help in detecting small objects.

The 13 x 13 layer is responsible for detecting large objects, whereas the 52 x 52 layer detects the smaller objects, with the 26 x 26 layer detecting medium objects. Here is a comparative analysis of different objects picked in the same object by different layers.

5.6.3 MORE BOUNDING BOXES PER IMAGE

For an input image of the same size, YOLO v3 predicts more bounding boxes than YOLO v2. For instance, at it's native resolution of 416 x 416, YOLO v2 predicted 13 x 13 x 5 = 845 boxes. At each grid cell, 5 boxes were detected using 5 anchors.On the other hand, YOLO v3 predicts boxes at 3 different scales. For the same image of 416 x 416, the number of predicted boxes are 10,647. This means that YOLO v3 predicts 10x the number of boxes predicted by YOLO v2 which makes it slower compared to YOLO v2. At each scale, every grid can predict 3 boxes using 3 anchors. Since there are three scales, the number of anchor boxes used in total are 9, 3 for each scale.

5.7 PRETRAINED MODEL

Tensorflow applications are deep learning models that are made available alongside pre-trained weights. These models can be used for prediction, feature extraction, and fine-tuning. Weights are downloaded automatically when instantiating a model. Here we are using the common object in context dataset to train the YOLO model and using them as our pre-trained weights during inference.

CHAPTER 6

DATA ANALYSIS USING MACHINE LEARNING

6.1 INTRODUCTION TO MACHINE LEARNING

Machine learning is a field of computer science that gives computer systems the ability to "learn" (i.e., progressively improve performance on a specific task) with data, without being explicitly programmed. Machine learning is employed in a range of computing tasks where designing and programming explicit algorithms with good performance is difficult or infeasible; example applications include email filtering, detection of network intruders or malicious insiders working towards a data breach, optical character recognition (OCR), learning to rank, and computer vision.

6.2 MACHINE LEARNING TASKS

Machine learning tasks are typically classified into two broad categories, depending on whether there is a learning "signal" or "feedback" available to a learning system:

- Supervised learning: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs. As special cases, the input signal can be only partially available, or restricted to special feedback.
- Semi-supervised learning: The computer is given only an incomplete training signal: a training set with some (often many) of the target outputs missing.
- Active learning: The computer can only obtain training labels for a limited

set of instances (based on a budget), and also has to optimize its choice of objects to acquire labels for. When used interactively, these can be presented to the user for labeling.

- Reinforcement learning: The training data (in form of rewards and punishments) is given only as feedback to the program's actions in a dynamic environment, such as driving a vehicle or playing a game against an opponent.
- Unsupervised learning: No labels are given to the learning algorithm, leaving it on its own to find a structure in its input.

6.3 MACHINE LEARNING APPLICATIONS

Categorization of machine learning tasks arises when one considers the desired output of a machine-learned system:

- In classification, inputs are divided into two or more classes, and the learner must produce a model that assigns unseen inputs to one or more (multi-label classification) of these classes. This is typically tackled in a supervised way. Spam filtering is an example of classification, where the inputs are email (or other) messages and the classes are "spam" and "not spam".
- In regression, also a supervised problem, the outputs are continuous rather than discrete.
- In clustering, a set of inputs is to be divided into groups. Unlike in classification, the groups are not known beforehand, making this typically an unsupervised task.
- Density estimation finds the distribution of inputs in some space.
- Dimensionality reduction simplifies inputs by mapping them into a lower-

dimensional space. Topic modeling is a related problem, where a program is given a list of human language documents and is tasked to find out which documents cover related topics.

6.4 DATA VISUALISATION

The information collected from machinery using the drone is analyzed to predict the state of the machine. Our dataset contains machines environmental data such as temperature, pressure, humidity, and machine orientation motion data. These data are transferred from machine to drone using the CC2500 module which is collected at the base station.

The plot below shows us the variation of temperature vs humidity

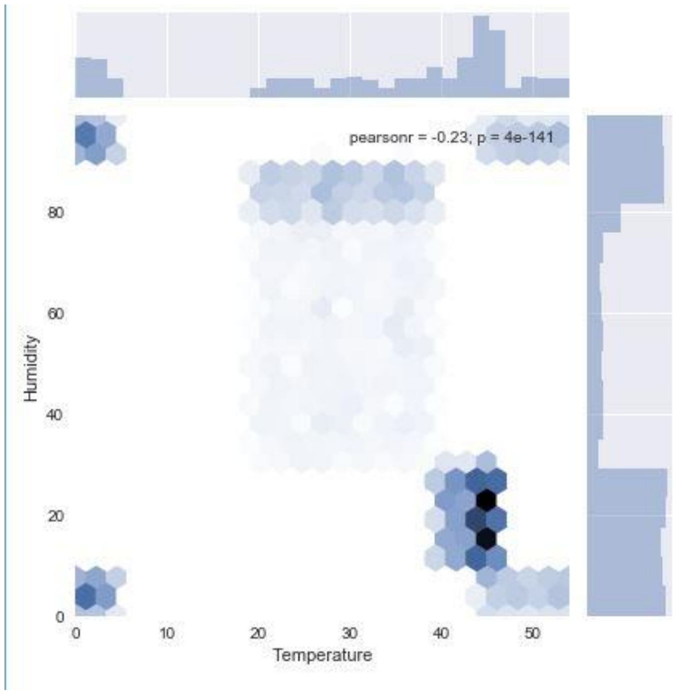


Fig 6.1 Temperature vs Humidity

6.5 DATA PREPROCESSING

The obtained raw data from machine needs to be pre-processed before we pass it to our machine learning model. Data normalization and cross-validation are those preprocessing steps used on this data.

6.5.1 DATA NORMALIZATION

Our data needs to be normalized between 0 to 1 in order reduce variance between input features and also to reduce the complexity of mathematical operations that will be performed on the data thereby reducing the training time. We use the scikit-learn StandardScaler pre-processing class to do normalization on all our input features.

6.5.2 CROSS-VALIDATION

In order to validate our model results, we split our training data into 75% of training and 25% of validation. Our accuracy and other metrics on the validation set are taken as our model output. The reason for incorporating this method is to check that our model has generalized well to any data and does not overfit and produce a high error rate.

6.6 RANDOM FOREST ALGORITHM

Random Forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms because it’s simplicity and the fact that it can be used for both classification and regression tasks.

Random Forest is a supervised learning algorithm. The forest it builds, is an ensemble of Decision Trees, most of the time trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result.

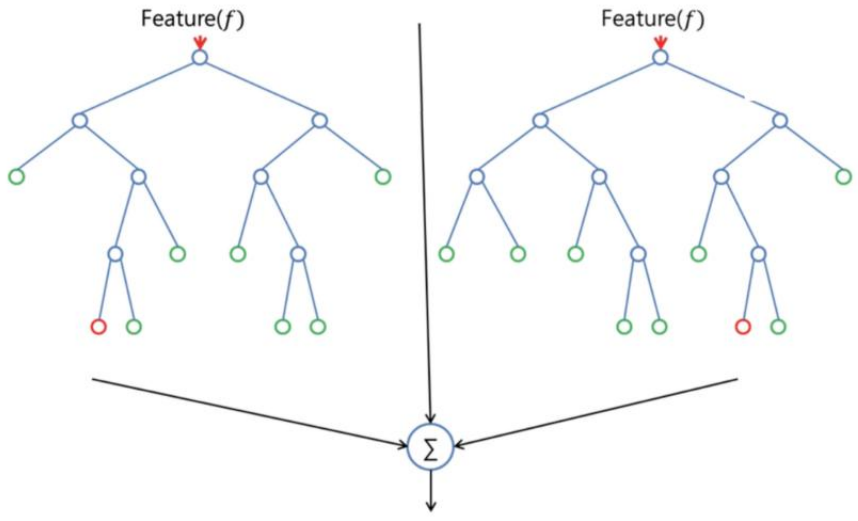


Fig 6.2 Random Forest Algorithm diagram

Random Forest adds additional randomness to the model while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

Therefore, in Random Forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. You can even make trees more random, by additionally using random thresholds for each feature rather than searching for the best possible thresholds (like a normal decision tree does). We have used Random forest as our machine learning model to which we pass our input features to predict the state of the machine to be either safe, moderate or at risk.

CHAPTER 7

HARDWARE AND SOFTWARE REQUIREMENTS

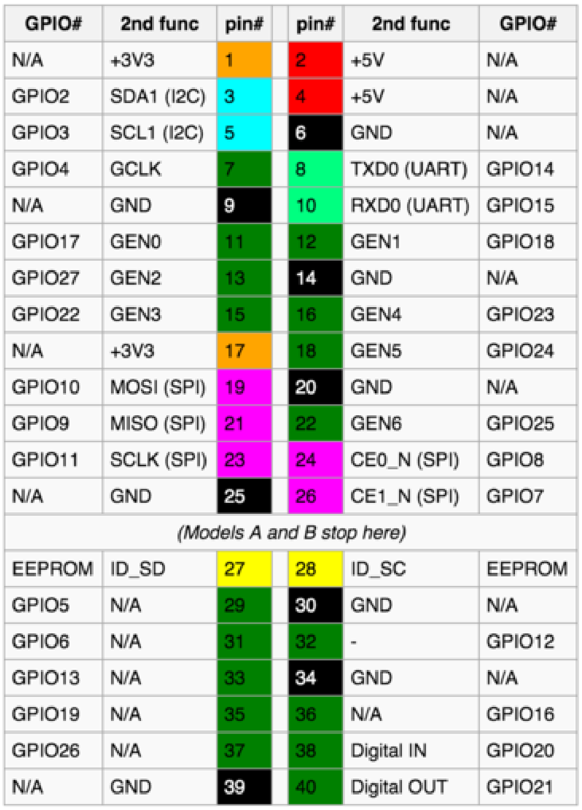
7.1 HARDWARE .

7.1.1 CC2500 RADIO FREQUENCY MODULE

The CC2500 is a low-cost 2.4 GHz transceiver designed for very low-power wireless applications. The circuit is intended for the 2400- 2483.5 MHz Industrial, Scientific and Medical and Short Range Device frequency band. The RF transceiver is integrated with a highly configurable baseband modem. The modem supports various modulation formats and has a configurable data rate up to 500 kBaud. CC2500 provides extensive hardware support for packet handling, data buffering, burst transmissions, clear channel assessment, link quality indication, and wake-on-radio. The main operating parameters and the 64- byte transmit/receive FIFOs of CC2500 can be controlled via an SPI interface.

7. 1. 2 RASPBERRY PI 3

The Raspberry Pi is a series of credit card-sized single-board computers developed in the England by the Raspberry Pi Foundation. Preloaded SD cards will be available from the RPi Shop. GPIO connector RPi A+, B+, 2B, 3 and 3B GPIO J8 40-pin pinout as shown in Table 7.1



GPIO#	2nd func	pin#		pin#	2nd func	GPIO#
N/A	+3V3	1		2	+5V	N/A
GPIO2	SDA1 (I2C)	3		4	+5V	N/A
GPIO3	SCL1 (I2C)	5		6	GND	N/A
GPIO4	GCLK	7		8	TXD0 (UART)	GPIO14
N/A	GND	9		10	RXD0 (UART)	GPIO15
GPIO17	GEN0	11		12	GEN1	GPIO18
GPIO27	GEN2	13		14	GND	N/A
GPIO22	GEN3	15		16	GEN4	GPIO23
N/A	+3V3	17		18	GEN5	GPIO24
GPIO10	MOSI (SPI)	19		20	GND	N/A
GPIO9	MISO (SPI)	21		22	GEN6	GPIO25
GPIO11	SCLK (SPI)	23		24	CE0_N (SPI)	GPIO8
N/A	GND	25		26	CE1_N (SPI)	GPIO7
(Models A and B stop here)						
EEPROM	ID_SD	27		28	ID_SC	EEPROM
GPIO5	N/A	29		30	GND	N/A
GPIO6	N/A	31		32	-	GPIO12
GPIO13	N/A	33		34	GND	N/A
GPIO19	N/A	35		36	N/A	GPIO16
GPIO26	N/A	37		38	Digital IN	GPIO20
N/A	GND	39		40	Digital OUT	GPIO21

Fig 7.1: Raspberry Pi GPIO Connector Pins

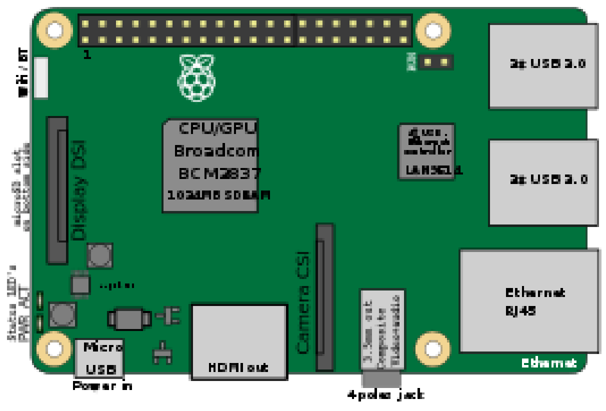


Fig 7.2: Description of the Raspberry Pi Kit

7.1.2.1 PRODUCT DESCRIPTION

The Raspberry Pi 3 Model B incorporates a few enhancements and new features. Improved power consumption, increased connectivity, and greater IO are among the improvements to this, small and lightweight ARM-based computer.

Chip: Broadcom BCM2836 SoC

Core architecture: ARMv8

CPU: A 1.2GHz 64-bit quad-core ARMv8 CPU

GPU: Broadcom VideoCore IV® Multimedia Co-Processor Provides Open GLES 2.0, hardware-accelerated OpenVG, and1080p30 H.264 high-profile decode Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure.

Memory: 1GB SD RAM

Operating System: Raspbian Jessie with Pixel

Dimensions: 85 x 56 x 17mm

Power: Micro USB socket 5V, 2.4A

Ethernet:10/100 Base T Ethernet socket

Wireless: 802.11n Wireless LAN and Bluetooth 4.1

Video Output: HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC)

Audio Output: 3.5mm jack, HDMI

USB: 4 x USB 2.0 Connector

GPIO Connector: 40-pin 2.54 mm (100 mil) expansion header: 2x20 strip Providing 27 GPIO pins as well as +3.3 V, +5V and GND
supply lines.

Camera Connector: 15-pin MIPI Camera Serial Interface (CSI-2)

JTAG: Not populated

Memory Card Slot: 32GB Class 10 Micro SD Card

7.2 SOFTWARE

7.2.1 Raspbian Stretch with PIXEL

Raspbian is the Foundation’s official supported operating system.Raspbian comes pre-installed with plenty of software for education, programming, and general use. It has Python, Scratch, Sonic Pi, Java, Mathematica and more.

Raspbian uses PIXEL, Pi Improved Xwindows Environment, Lightweight as its main desktop environment as of the latest update. It is composed of a modified LXDE desktop environment and the Open box stacking window manager with a new theme and few other changes.

7.2.1.1 Installation

Download the Image

- Official images for recommended operating systems are available to download from the Raspberry Pi Website Downloads page.
- After downloading the .zip file, unzip it to get the image file (.img) for writing to your SD card.

Bootable MicroSD Card

- Use win32diskimager for Windows or Terminal for Mac and write the unzipped image onto the formatted SD card.
- Double check by reinserting the SD card and check if all files have been copied without any error.

Procedure

Once booted the default Username: Pi and Password: raspberry is entered and logged in using the GUI. The SU- password is then set for further use. We connect to the Internet either by WiFi or by connecting the Ethernet Cable.
The system is updated to the latest firmware using the following commands:

- sudo apt-get upgrade
- sudo apt-get update

7.2.2 OpenCV

OpenCV (*Open Source Computer Vision*) is a library of programming functions mainly aimed at real-time computer vision. The library is cross-platform and free for use under the open-source BSD license.
OpenCV's application areas include:

- 2D and 3D feature toolkits, Ego motion estimation
- Facial recognition system,Gesture recognition
- Human–computer interaction (HCI), Mobile robotics
- Motion understanding, Object identification
- Segmentation and recognition, Stereopsis stereo vision: depth perception from 2 cameras.
- Structure from motion (SFM), Motion tracking, Augmented reality

To support some of the above areas, OpenCV includes a statistical machine learning Library that contains:

- Boosting, Decision tree learning, Gradient boosting trees
- Expectation-maximization algorithm
- k-nearest neighbour algorithm, Naive Bayes classifier

- Artificial neural networks
- Random forest,Support vector machine (SVM)

OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. There are bindings in Python, Java and MATLAB/OCTAVE. The API for these interfaces can be found in the online documentation. Wrappers in other languages such as C#, Perl, Ch, Haskell and Ruby have been developed to encourage adoption by a wider audience. All the new developments and algorithms in OpenCV are now developed in the C++ interface. OpenCV runs on a variety of platforms. Desktop: Windows, Linux, macOS, FreeBSD, NetBSD, OpenBSD; Mobile: Android, iOS, Maemo, BlackBerry 10.

7.2.2.1 Step by step Installation

Installing OpenCV in Raspberry PI includes the following steps:

1: Configuring your Raspberry Pi by installing required packages and libraries

Step 1: sudo apt-get update

Step 2: sudo apt-get upgrade

Step 3: sudo rpi-update

Step 4: sudo apt-get install build-essential git cmake pkg-config

Step 5: sudo apt-get install libjpeg8-dev libtiff4-dev libjasper-dev libpng12-dev

Step 6: sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev

Step 7: sudo apt-get install libgtk2.0-dev

Step 8: sudo apt-get install libgtk2.0-dev

Step 9: cd ~

Step 10: git clone https://github.com/Itseez/opencv.git

Step 11: cd opencv

Step 12: git checkout 2.7.2

2: Compiling OpenCV 3.0 with Python 2.7+ support

Step1: sudo apt-get install python2.7-dev

Step 2: wget https://bootstrap.pypa.io/get-pip.py

Step 3: sudo python get-pip.py

Step 4: sudo pip install virtualenv virtualenvwrapper

Step 5: sudo rm -rf ~/.cache/pip

Step 6: # virtualenv and virtualenvwrapper

Step 7: export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python2.7

Step 8: export WORKON_HOME=$HOME/.virtualenvs

Step 9: source /usr/local/bin/virtualenvwrapper.sh

Step 10: source ~/.profile

Step 11: mkvirtualenv cv3

Step 12: workon cv3

Step 13:pip install numpy

Step 14: sudo rm -rf ~/.cache/pip/

Step 15:pip install numpy

Step 16: cd ~/opencv

Step 17: mkdir build

Step 18: cd build

Step 19: cmake -D CMAKE_BUILD_TYPE=RELEASE \

Step 20:-D CMAKE_INSTALL_PREFIX=/usr/local \-D INSTALL_C_EXAMPLES=ON \-D INSTALL_PYTHON_EXAMPLES=ON \-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules \-D

BUILD_EXAMPLES=ON

Step 21: make -j4

Step 22: sudo make install

Step 23: sudo ldconfig

3: Verifying your OpenCV 3.0 install

Step 1: workon cv

Step 2: python

```
>>> import cv2
>>> cv2.__version__
```

7.2.3 KERAS 2.0

7.2.3.1 INSTALLATION

Below is a step by step installation & verification of Keras with TensorFlow running on the backend.

1. Setup:

Step 1: mkvirtualenv tf_keras

Step 2: workon <virtual env name> (workon tf_tkeras)

Step 3: pip install numpy scipy

Step 4: pip install scikit-learn

Step 5: pip install pillow

Step 6: pip install h5py

Step 7: export TF_BINARY_URL=
https://storage.googleapis.com/tensorflow/mac/cpu/tensorflow-0.11.0rc2-py2-none-any.whl

Step 8: pip install --upgrade $TF_BINARY_URL

Step 9: pip install keras

2.Verifying proper install:

Step 1: workon tf_keras (Virtual Environment)

Step 2: python

```
        >>> import tensorflow
        >>> import keras
        >>> quit()
```

7.2.4 VNC

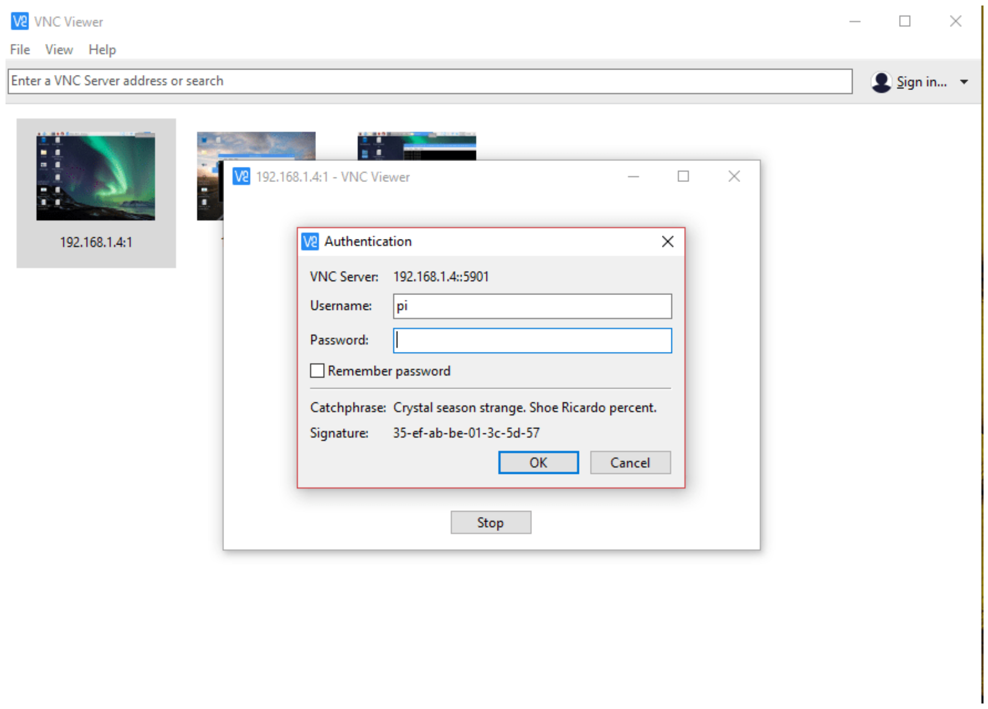
RealVNC is a company that provides remote access software. The software consists of a server and client application for the Virtual Network Computing (VNC) protocol to control another computer's screen remotely. For a desktop-to-desktop connection RealVNC runs on Windows, on Mac OS X, and on many Unix-like operating systems. A RealVNC client also runs on the Java platform and on the Apple iPhone, iPod touch and iPad and Google Android devices. A Windows-only client, VNC Viewer Plus is now available, designed to interface to the embedded server on Intel AMT chipsets found on Intel vPro motherboards. For remote access to view one computer desktop on another, RealVNC comes in one of three editions:

1. Open Edition (formerly "Free Edition")

2. Personal Edition

3. Enterprise Edition

Release 4.6 included features such as HTTP proxy support, chat, an address book, remote printing, Unicode support, and connection notification. Users must activate each of the server versions. In November 2016, RealVNC released the updated version of their software, now called VNC Connect (version 6.0). The new version introduces a cloud connection option using a subscription-based pricing model. Home and Professional subscriptions are cloud connections only. Now, we can access the Raspberry Pi from any client by punching in this unique IP address and port number to gain access to the Raspberry Pi. We would require entering the Username and Password to take control of the device. RealVNC even has the option to chat with the server and FTP services as well.


VNC Viewer
File View Help
Enter a VNC Server address or search
Sign in...
192.168.1.4:1
192.168.1.4:1 - VNC Viewer
Authentication

VNC Server: 192.168.1.4::5901
Username: pi
Password:
Remember password
Catchphrase: Crystal season strange. Shoe Ricardo percent.
Signature: 35-ef-ab-be-01-3c-5d-57

Fig 7.3: VNC Viewer login page

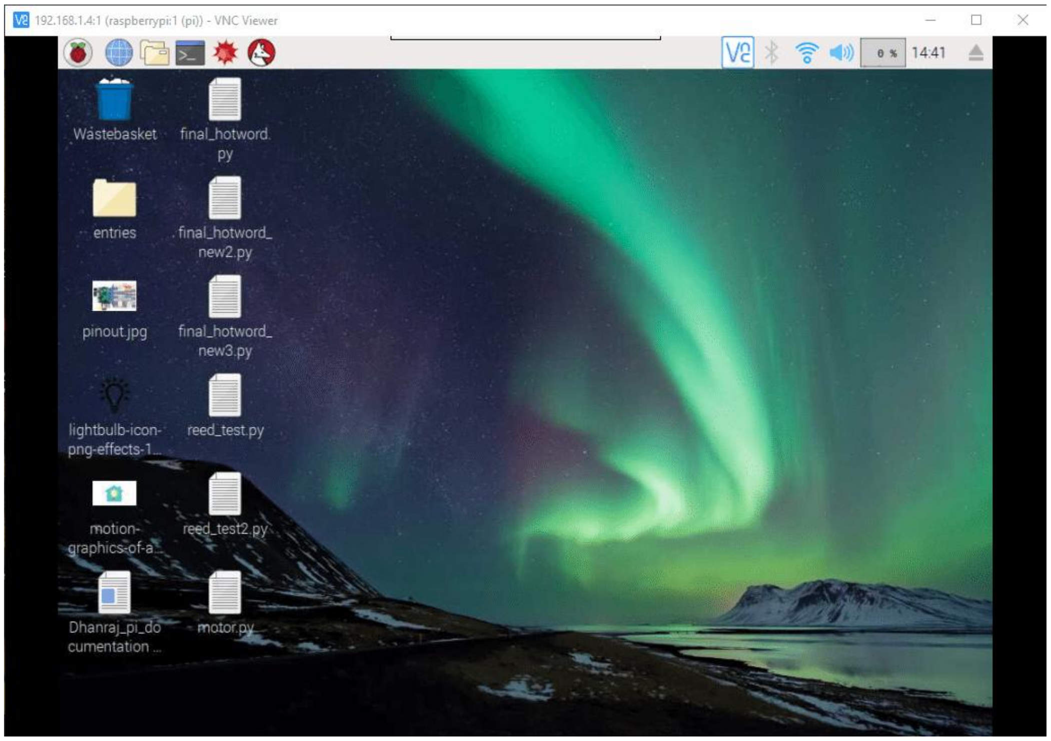


Fig 7.4: Raspberry Pi on the VNC Viewer

7.2.6 PuTTY

PuTTY is a source terminal, serial console and network file transfer application. It supports several network protocols, including SCP, SSH, Telnet, rlogin, and raw socket connection. It can also connect to a serial port.

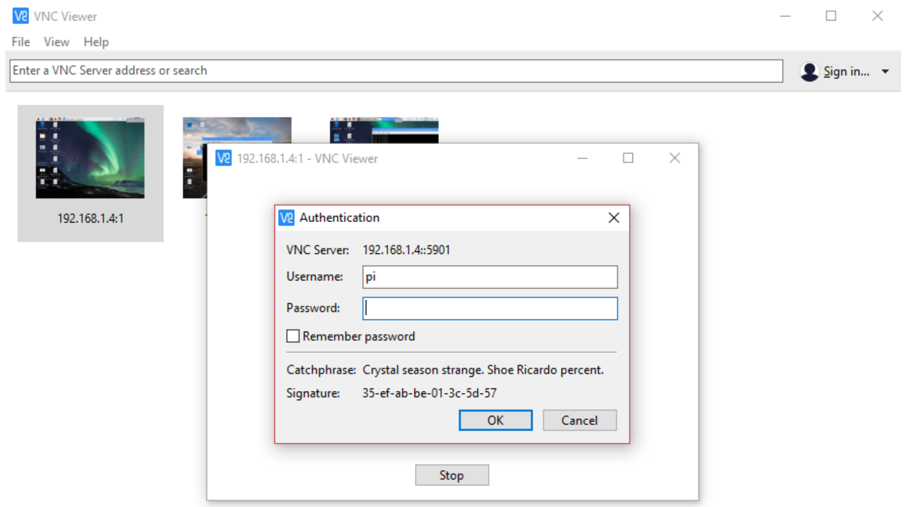
PuTTY comes bundled with command-line SCP and SFTP clients, called "pscp" and "psftp" respectively, and plink, a command-line connection tool, used for non-interactive sessions.

PuTTY consists of several components:

- PuTTY: the Telnet, rlogin, and SSH client itself, which can also connect to a seria lport
- PSCP: an SCP client, i.e. command-line secure file copy
- PSFTP: SFTP client, i.e. general file transfer sessions much like FTP
- PuTTYtel: A Telnet-only client

- Plink: a command-line interface to the PuTTY back ends
- Pageant: an SSH authentication agent for PuTTY, PSCP and Plink
- PuTTYgen: an RSA and DSA key generation utility
- Pterm: a standalone terminal emulator

Connecting to the terminal of the Raspberry Pi via PuTTY is useful and would be used in a window based system majorly. As in Mac the SSH is built into the Terminal of the computer. We first need to find out the IP Address of the Raspberry Pi after Installing the Operating System and pairing up with the internet. After which this IP Address is entered the PuTTY portal with the respective Username and Password. This would give access to the terminal of the remote computer or Raspberry Pi in this case the only condition is that the server and the client is required to be in the same network for PuTTY to access the server from the client.



VNC Server: 192.168.1.4::5901
Username: pi
Password:
Catchphrase: Crystal season strange. Shoe Ricardo percent.
Signature: 35-ef-ab-be-01-3c-5d-57

Fig 7.5: PuTTY Home Page

7.2.7 PYTHON

Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. Python supports multiple programming paradigms, including object oriented,imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.

Python interpreters are available for installation on many operating systems, allowing Python code execution on a wide variety of systems. Using third-party tools, such as Py2exe or Pyinstaller, Python code can be packaged into standalone executable programs for some of the most popular operating systems, allowing the distribution of Python-based software for use on those environments without requiring the installation of a Python interpreter. CPython, the reference implementation of Python, is free and open-source software and has a community-based development model, as do nearly all of its alternative implementations. CPython is managed by the non-profit Python Software Foundation.

7.2.7.1 FEATURES

Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. An important feature of Python is dynamic name resolution (late binding), which binds method and variable names during program execution. Rather than requiring all desired functionality to be built into the language's core, Python was designed to be highly extensible.

7.2.7.2 TYPING

Python uses duck typing and has typed objects but untyped variable names. Type constraints are not checked at compile time; rather, operations on an object may fail, signifying that the given object is not of a suitable type. Python allows programmers to define their own types using classes, which are most often used for object-oriented programming. New instances of classes are constructed by calling the class and the classes themselves are instances of the metaclass type (itself an instance of itself), allowing and reflection.

7.2.7.3 LIBRARIES

Modules for creating graphical user interfaces, connecting to relational databases, pseudorandom number generators, arithmetic with arbitrary precision decimals, manipulating regular expressions, and doing unit testing are also included. Python Package Index contains more than 72,000 packages offering a wide range of functionality, including:

- Graphical user interfaces, web frameworks, multimedia, databases, networking, and communications
- Test frameworks, automation and web scraping, documentation tools, system administration

7.2.7.4 USES

The Raspberry Pi single-board computer project has adopted Python as its principal user-programming language.

CHAPTER 8

IMPLEMENTATIONS AND RESULT

8.1 REQUIREMENTS FOR EACH MODULE

The project entirely is split into three modules

- Object Detection
- Machine State classification
- Communication between Raspberry Pi and drone

8.1.1 MODULE 1: OBJECT DETECTION

Input, Output, and Hardware required:

The input to this module is the image captured from the camera attached in the drone.

The output of this module is the object detected inside the bounding box with the class label and the probability it belongs to this class. The algorithm can detect multiple objects if present in an image.

The code is optimized to run on a laptop or desktop CPU without the need of GPU.

8.1.2 MODULE 2: MACHINE STATE CLASSIFICATION

Input, Output, and Hardware required:

The input to this module is the environment data captured from the raspberry pi senehat sensor present in the machine which is sent to drone using the CC2500 radio frequency module.

The output of this module is the present state of the machine that is whether the machine is in safe, moderate or danger state.

The hardware required is a local laptop CPU for inference and CC2500 RF module to send data from machine to drone.

8.2 OBJECT DETECTION - OUTPUT

The drone view image is taken as a test image and the output is the bounding box of the object with class name and class probabilities as shown below.

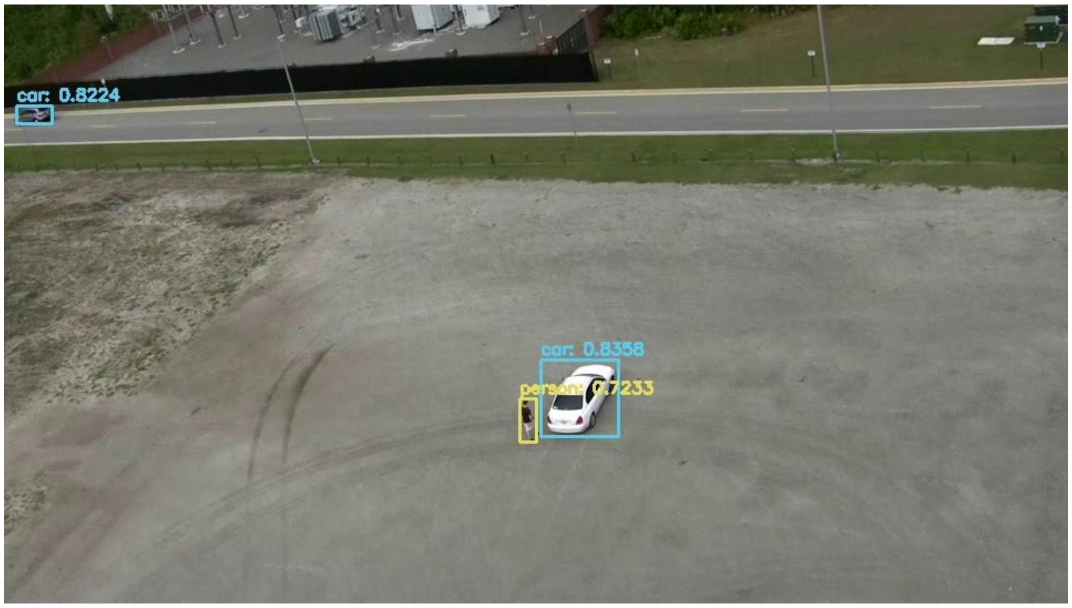
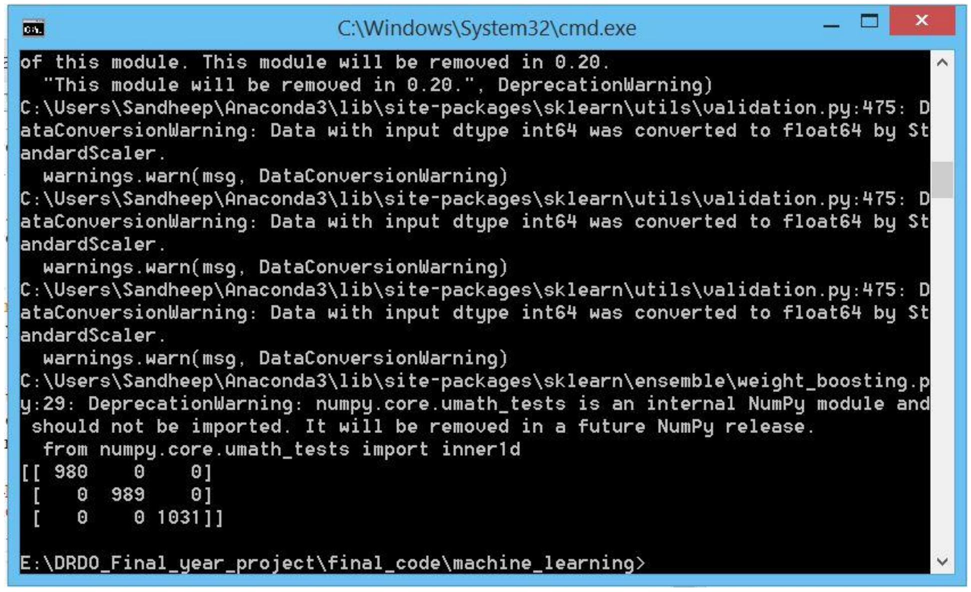


Fig 8.1: Deep learning output

8.3 MACHINE LEARNING - OUTPUT

The confusion matrix of the test data is chosen as a metric to evaluate the model.



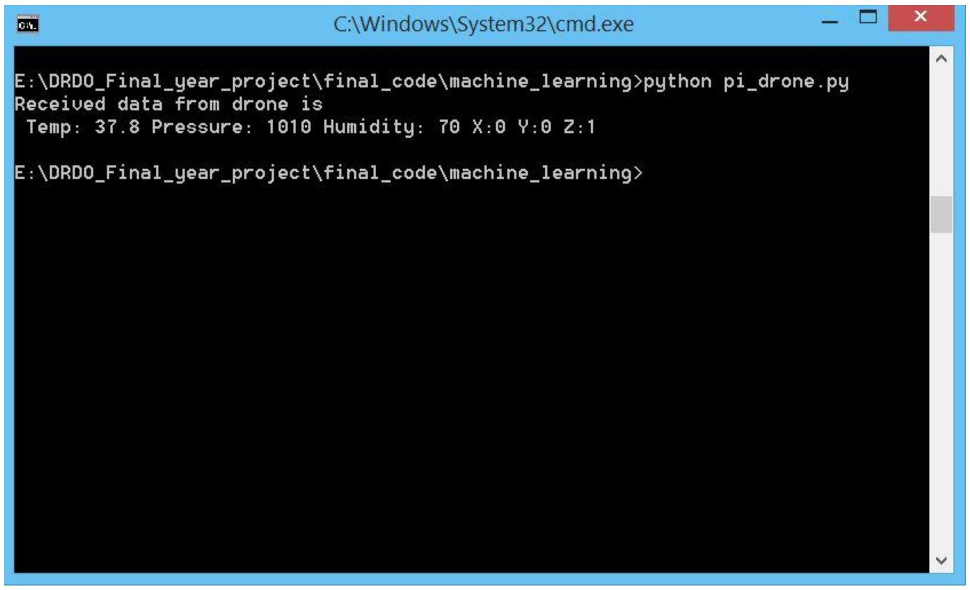
```
of this module. This module will be removed in 0.20.
  "This module will be removed in 0.20.", DeprecationWarning)
C:\Users\Sandheep\Anaconda3\lib\site-packages\sklearn\utils\validation.py:475: D
ataConversionWarning: Data with input dtype int64 was converted to float64 by St
andardScaler.
  warnings.warn(msg, DataConversionWarning)
C:\Users\Sandheep\Anaconda3\lib\site-packages\sklearn\utils\validation.py:475: D
ataConversionWarning: Data with input dtype int64 was converted to float64 by St
andardScaler.
  warnings.warn(msg, DataConversionWarning)
C:\Users\Sandheep\Anaconda3\lib\site-packages\sklearn\utils\validation.py:475: D
ataConversionWarning: Data with input dtype int64 was converted to float64 by St
andardScaler.
  warnings.warn(msg, DataConversionWarning)
C:\Users\Sandheep\Anaconda3\lib\site-packages\sklearn\ensemble\weight_boosting.p
y:29: DeprecationWarning: numpy.core.umath_tests is an internal NumPy module and
 should not be imported. It will be removed in a future NumPy release.
  from numpy.core.umath_tests import inner1d
[[ 980    0    0]
 [   0  989    0]
 [   0    0 1031]]

E:\DRDO_Final_year_project\final_code\machine_learning>
```

Fig 8.1: Machine learning Output

8.4 RASPBERRY PI - DRONE COMMUNICATION - OUTPUT

The received data from a drone is printed on the terminal



```
E:\DRDO_Final_year_project\final_code\machine_learning>python pi_drone.py
Received data from drone is
 Temp: 37.8 Pressure: 1010 Humidity: 70 X:0 Y:0 Z:1

E:\DRDO_Final_year_project\final_code\machine_learning>
```

Fig 8.3: Raspberry Pi Communication with drone

CHAPTER 9

CONCLUSION AND FUTURE SCOPE

The outcome of this project is to propose an innovative addition to the pre-existing UAVs and drones by equipping it with cutting edge autonomy. These changes that we proposed will display significant elevation in terms of performance and efficiency when compared to its predecessors. The deep learning aspect of the drone is where the heart of the innovation lies. The YOLOv3 algorithm is the feature that makes this UAV stand out when compared to the existing models. The autonomous operation would open doors to widespread applications in Defence and Military regions that are life-threatening to manned soldiers to access. The high sensitive borders and latest generation military machinery can be put under high accuracy top-notch surveillance using these Autonomous Unmanned Aerial Vehicles.

As discussed earlier the main application of the drone is surveillance. However, over the years, the structure of the drones can be enhanced in order to support payloads of greater weight for a longer duration and flight path. The future scope for drone involvement is vast. The defence applications of this drone make it such a valuable piece of technology as the war zones are very hostile and unpredictable to rely on human intervention for machinery maintenance. Drones are also deployed to facilitate the regular monitoring and maintenance of machinery in line of control.

-

REFERENCES

[1] Suet-Peng Yong, Yoon-Chow Yeong, Human Object Detection in Forest with Deep Learning based on Drones Vision4th International Conference on Computer and Information Sciences (ICCOINS) Year: 2018 Page s: 1 - 5.

[2] Sunyoung Cho, Jihun Park, Young Sook Shin, Sang-ho Lee, Recognizing Human-Object Interactions via Target LocalizationControl Automation and Systems (ICCAS) 2018 18th International Conference on, pp. 836-840, 2018.

[3] Apurv Saha, Akash Kumar, Aishwarya Kumar Sahu," FPV Drone with GPS used for Surveillance in Remote Areas 10th International Symposium on Advanced Topics in Electrical Engineering (ATEE), Bucharest, 2017,

[4] Tuton Chandra Mallick, Mohammad Ariful Islam Bhuyan, Mohammed Saifuddin Munna Design & Implementation of a UAV (Drone) with Flight Data Record

[5] E. Mingkhwan and W. Khawsuk Digital image stabilization technique for the fixed camera on small size drone, Third Asian Conference on Defence Technology (ACDT), Phuket, 2017, pp.12-19

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, You Only Look Once: Unified, real-time object detection, Proceedings of CVPR, 2016.

[7] D. Mellinger, M. Shomin, and V. Kumar, Control of quadrotors for robust perching and landing, International Powered Lift Conference, October 5-7, 2010, 2010.