A Project Report on

VISION

A Drone for Emergency Operations

Submitted in partial fulfillment of award of

BACHELOR OF TECHNOLOGY
Degree
in
COMPUTER SCIENCE & ENGINEERING

By

ABHISHEK KUMAR - 1608210006
AKSHITA SHARMA - 1608210012
ANKIT AGARWAL - 1608210021
ASHISH - 1608210038

(2016-2020)

Mr. Zubair Iqbal
Assistant Professor

SUPERVISOR

Dept. of Computer Science & Engineering
Moradabad Institute of Technology
Moradabad (U.P.)

A Project Report on

VISION

A Drone for Emergency Operations

Submitted in partial fulfillment of award of

BACHELOR OF TECHNOLOGY

Degree
in

COMPUTER SCIENCE & ENGINEERING

By

ABHISHEK KUMAR-1608210006
AKSHITA SHARMA-1608210012
ANKIT AGARWAL-1608210021
ASHISH – 1608210038

(2016-2020)

Mr. Zubair Iqbal
Assistant Professor

SUPERVISOR

MORADABAD INSTITUTE OF TECHNOLOGY
Mit
IN PURSUIT OF EXCELLENCE

Head
Computer Sci. & Engg. Department
Moradabad Institute of Technology
Moradabad-24400

Deptt. of Computer Science & Engineering

Moradabad Institute of Technology

Moradabad (U.P.)

ABSTRACT

In the present Era, there are too many developments in the field of drones. Besides, the military and commercial applications of drones, there is no doubt in their efficiency in case of supporting emergency disaster operations. It focuses mainly in emergency operations, like activity immediately after the occurrence of a disaster like nuclear accidents, dangerous material releases, floods, earthquakes. For special rescue teams, the drone can help much in a rapid location selection, detecting real time situation, counting number of people or males and females. Floods are typical for a slow onset disaster. In contrast, managing floods is a very complex and difficult task. It requires continuous monitoring of dykes, flooded and threatened areas. Drone can help managers largely keeping an area under observation. Drone can be used for fire detection, intervention monitoring and also for post-fire monitoring. In case of nuclear accident or hazardous material

leakage drone is also a very effective or can be the only one tool for supporting emergency operations.

A quadcopter can achieve vertical flight in a stable manner and be used to monitor or collect data in a specific region such as mapping terrains. Technological advances have reduced the cost and increase the performance of the low power microcontrollers that allowed the general public to develop their own quadcopter. The goal of this project is to build, modify, and improve an existing quadcopter kit to obtain stable flight, gather and store GPS data, and perform auto commands, such as auto-landing. The components used in the project are Quadcopter frame, motors, electronic speed controllers, Arduino Mega development board, Auto Pilot Mode and sensor boards. Batteries, a transmitter, a receiver, a GPS module, and a micro SD card adaptor were interfaced with the kit. Individual components were tested and verified to work properly.

Head
Computer Sci. & Engg. Department
Moradabad Institute of Technology
Moradabad-24400

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B.Tech project undertaken during final year. We owe gratitude to our project guide Mr. Zubair Iqbal, this project would not be able to implement completely without his guidance and support. We are deeply indebted to our project guide for his patience, inspiration, guidance, constant encouragement, moral support, keen interest, and valuable suggestions during preparation of project.

The seniors who continuously encouraged us through the implementation phase of project are Kanika Sharma and Radhika Rathi. We are indebted to them for their constructive support and cooperation and for taking interest in discussing our problem and encouraging us.

Abhishek
Akshita.
Ankit

ABHISHEK KUMAR-1608210006
AKSHITA SHARMA-1608210012
ANKIT AGARWAL-1608210021
ASHISH-1608210038
(COMPUTER SCIENCE AND ENGINEERING)

Head
Computer Sc & Engg. Department
Institute of Technology

CHAPTER 1

INTRODUCTION

When a natural disaster occurs in a populated zone, a fast and effective organization of the disaster management is required to assist the affected population, minimize the number of victims and limit the economic impact. At all phases of disaster management (preparedness, response, recovery and reconstruction) one of the first actions to be taken is to set up a disaster cell for coordination. The detection and the monitoring of the impact of natural disasters on the terrain are mainly performed by space borne and air borne sensors relying on radio and optical instruments. Contrary to limitations in the time window of observation attached to optical, radio observations are available 24/7 and relatively insensitive to atmospheric conditions: These are therefore particularly useful during the “Response phase” of the disaster management cycle when information must be delivered as quickly as possible to the disaster cells.

UAVs (Unmanned Aerial Vehicle) may bring significant improvements with respect to these issues. Depending on the requirements of their potential mission, UAVs can easily be equipped with different kinds of sensors, including optical ones. Ofcourse, their altitude permits a higher quality of images and an under-the-clouds observation. Finally, search and rescue teams may carry UAVs and deploy them based on the site needs. For example, flying drones can be used to explore flooded areas in order to find a practicable path to victims. In this respect, UAVs extend the exploration range of rescue teams while at the same time improving their own safety in areas that may be dangerous. A good example is the sense Fly UAV (Ackerman, 2013), which during the aftermath of 2010 Haiti earthquake, automatically mapped the region, enabling authorities to quickly draw maps of devastated areas. These maps helped the rescue teams and could improve the lives of victims in the aftermath of the earthquake. Of course, UAV also have drawbacks and cannot be operated at all time. First of all, they have to cope with weather conditions at a higher degree than conventional aircrafts: because of their light weight and rather low propulsive power, they are more sensitive to wind gusts, for instance. Secondly, they are less prepared to face difficult environments, such as heavy rain (water-proof conception) or hot temperatures. Therefore, these basic conditions have to be checked prior to operating them.

1.1 RELATED WORKS

The numbers of use cases where drones have already been useful in humanitarian settings are numerous; the ones described hereafter are just a small subset of these for illustration purposes. Dan office IT, has a commercial drone solution for disaster response (Pedersen, 2014). It was used in real operation sites such as the typhoon Yolanda in Tacloban, Philippines, where it helped in the identification of the operation site, and on the identification of feasible roads. In the same disaster, the Core Phil DSI team used a fixed wing drone, eBee, to capture aerial imagery of downtown Tacloban. These images were further analysed through crowdsourcing and helped in the generation of the most detailed and up-to-date maps of the region. These maps were afterwards used by different humanitarian organizations and even by the Filipino Government. The control of fleets of drones is also not a new theme. In fact, it is a well-studied subject in the military context. Ofcourse, the purpose is here different; the same goes for the flight control. However, even on military operations the proposed fleet control mechanisms intend, basically, to help humans to control the drones rather than providing a fully autonomous fleet. For example, proposed an automation architecture to help humans on the supervision of a drone fleet, but the drones are not completely autonomous, it is still up to the human operator to decide the drones mission.

1.2 PROPOSED SYSTEM

We are going to develop a drone with some AI features like object detection, face recognition and perform some special tasks like counting people and counting number of males and females with the help of ML and Al techniques which will help in some special or emergency operations so first we need to make a drone. To make a drone we need a frame, four brushless motors, ESC (electronic speed controller), propellers, flight controller, transmitter and receiver, power distribution board, GPS, and battery for power supply and assemble all these parts which will forms a drone which perform all basic drone tasks like take off, move forward, move backward, move left, move right and landing so this is a basic drone now we need to make it smart. With the help of Raspberry Pi and Pi cam we give it an Intelligence from python programming in raspberry pi. To give intelligence to the drone we use open CV for object detection and face recognition which take input from the Pi Cam and GPS and after analyses the data it send data to the system administrator which is a easy task rather than use a manual method. To understand the working of the Vision we took an example. For example, there is a flood in the city and we need to rescue the stuck people in the city and we did not know that how many number of people in the city, so we send our vision drone to the location which have intelligence to do this work efficiently. Then the drone Vision analyses all the situation and send the calculated data to the administrator.

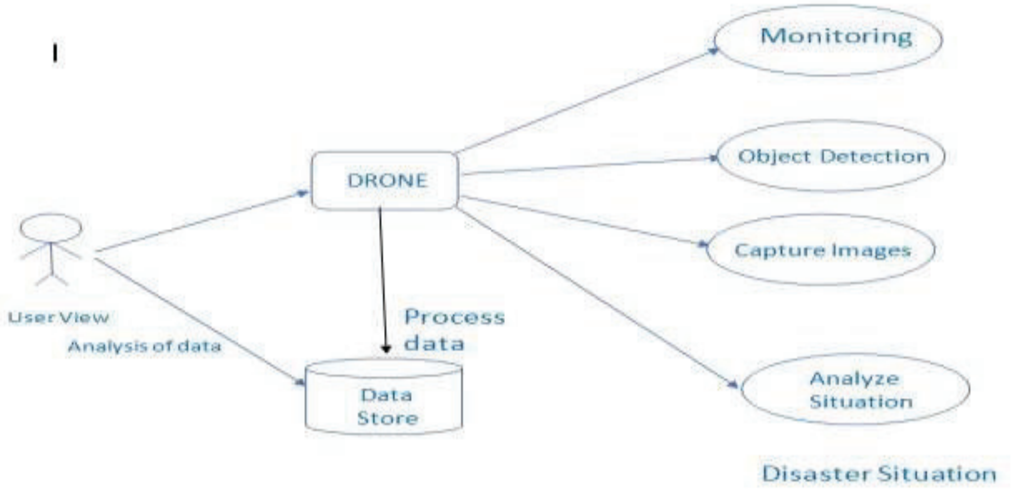


Fig:1.1 Use Case Diagram of Proposed System

1.3 USES OF PROPOSED SYSTEM

- Real time surveillance
- Search and rescue.
- Forest fire detection.
- Oil, gas and mineral exploration and production search and rescue operation.
- Other disaster related rescuing operations.

1.4 OBJECTIVE OF THE SYSTEM

The primary objective of this paper is to evaluate how Drones or unmanned aerial vehicles –UAV's in the present or near future can help survivors in the case of a tsunami, earthquake, flooding, and any natural disaster. Initially, we assume that in any natural disaster always exist the high possibility of damage to the infrastructure, transportation systems, telecommunications systems access, and basic services immediately. This research proposes three areas the uses of unmanned aerial vehicles –UAV's- (or Drones) in the case of natural disasters response and humanitarian relief aid. These are

- Live Streaming and monitoring.
- Save lives.
- Drones minimize the obvious dangers and health risks.

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION

2.1 INTRODUCTION

2.1.1 Purpose

Outlining the specifications of a drone can be remotely controlled for a variety of operations primarily to better predict real-time data. VISION is commonly associated with Disaster management and surveillance. An interactive data will allow the person to contact immediate with Rescue teams available during the time of disaster.

2.1.2 Overview

The planned outcome of this project is a Drone which is a flying robot with an embedded system and work in the conjunction with on board sensors and GPS. It will detect and analysis the disaster and count number of people, their gender and prepare the dataset, also monitor the real time situation of the area.

2.2 GENERAL DESCRIPTION

2.2.1 Product Functions

VISION-A drone for emergency operation project is designed to detect and analyze the disaster and count number of people, their gender and prepare the dataset and detect the real time situation of the area.

2.2.2 User Characteristics

Using the Pi-Cam emergency operations, including Real time surveillance, Search and rescue and other disaster related rescue operations can be monitored. Specific resources will be provided for general accessibility.

2.2.3 User Problem Statement

Following major obstacles are:

- Handling the Weight of the Drone.
- Size of Quadcopter.
- 100% Accuracy in Modules.
- Providing the real time connection for monitoring real time data.

2.2.4 User Objectives

- Capable of real time face recognition.
- Real time gender analysis.
- Can easily recognize an object can along its path.
- Ability of live streaming and monitoring.
- Capable to fly upto the range of 100 metres.
- Drone can be handle using the tx^n and rx^n with the help of remote.

2.2.5 General Constraints

Constraints faced were related to the weight and balancing of the drone if the weight increases then it is quite difficult to fly the drone.

2.3 FUNCTIONAL REQUIREMENTS

2.3.1 Direction Search

- **Description**

  A flying robot used to detect and analyze the disaster and count number of people, their gender and prepare the dataset and detect the real time situation of the area.

- **Technical Issues**

  Algorithm should be easy to implement. Most difficult portion of requirement will be accuracy of the algorithms and binding together the part of IoT as well as Machine Learning algorithms.

2.4 INTERFACE REQUIREMENTS

2.4.1 User Interfaces

- **Graphical User Interface**

  The user interface will consist of a view page where a person handling a drone can monitor the real time situation where a person can detect where the disaster is and with the help of that he can inform to the rescue team about the disaster and can save millions of life. The interface will be database driven, and scaled to be available across different platforms.

2.5 PERFORMANCE REQUIREMENTS

2.5.1 User Performance Requirements

The user's performance will be dependent on the speed of their Internet connection, gps location as well as the performance of the network. The project will be designed to be compatible with a medium connectivity of service providers, so as to be widely accessible.

2.6 OTHER NON-FUNCTIONAL ATTRIBUTES

2.6.1 Security

No security measures are expected to be involved excepting that there will be no obstacle coming along the path, which will be dangerous for the frame to fly off. The database will be protected.

2.6.2 Reliability

Based on up-time of server on which data monitoring resides. Regular backup of database, may be required to ensure up-time and reliability. Also the power controlling systems and gps must be maintained perfectly.

2.6.3 Maintainability

- Always Keep a Flight Log.
- Avoid crash landing the drone.
- Avoid no-flying zones.
- Avoid stopping the drone while going in full speed.
- Keep your remote controller batteries charged.
- Take care not to bump the drone when transporting it, better yet, get a hard case for transporting the drone.

2.6.4 Portability

The defining feature of the Air is its portability. The entire package, from the drone itself, to its batteries new remote are all designed for easy storage and transport.

2.7 WORK FLOW DIAGRAM

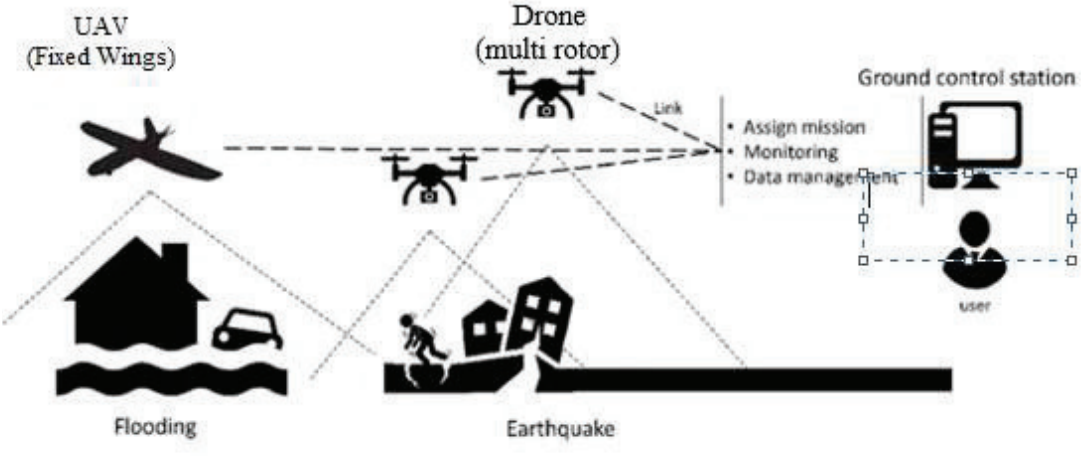


Fig.2.1: Work Flow Diagram of the Application

CHAPTER 3

LITERATURE SURVEY

3.1 PROJECT MODULES

- Face Recognition.
- Gender Recognition.
- Object Recognition.
- Creation of Drone.

3.2 MODULES DESCRIPTION

3.2.1 Face Recognition

A facial recognition system is a technology capable of identifying or verifying a person from a digital image or a video frame from a video source. There are multiple methods in which facial recognition systems work, but in general, they work by comparing selected facial features from given image with faces within a database. It is also described as a Biometric Artificial Intelligence based application that can uniquely identify a person by analyzing patterns based on the person's facial textures and shape.

While initially a form of computer application, it has seen wider uses in recent times on mobile platforms and in other forms of technology, such as robotics. It is typically used as access control in security systems and can be compared to other biometrics such as fingerprint or eye iris recognition systems. Although the accuracy of facial recognition system as a biometric technology is lower than iris recognition and fingerprint recognition, it is widely adopted due to its contactless and non-invasive process. Recently, it has also become popular as a commercial identification and marketing tool. Other applications include advanced human-computer interaction, video surveillance, automatic indexing of images, and video database, among others. Some face recognition algorithms identify facial features by extracting landmarks, or features, from an image of the subject's face[1]. For example, an algorithm may analyze the relative position, size, and/or shape of the eyes, nose, cheekbones, and jaw. These features are then used to search for other images with matching features.

Other algorithms normalize a gallery of face images and then compress the face data, only saving the data in the image that is useful for face recognition. A probe image is then compared with the face data. One of the earliest successful systems is based on template matching techniques applied to a set of salient facial features, providing a sort of compressed face representation.

Recognition algorithms can be divided into two main approaches: geometric, which looks at distinguishing features, or photometric, which is a statistical approach that distills an image into values and compares the values with templates to eliminate variances. Some classify these algorithms into two broad categories: holistic and feature-based models. The former attempts to recognize the face in its entirety while the feature-based subdivide into components such as according to features and analyze each as well as its spatial location with respect to other features.

Popular recognition algorithms include principal component analysis using eigenfaces, linear discriminant analysis, elastic bunch graph matching using the Fisher face algorithm, the hidden Markov model, the multilinear subspace learning using tensor representation, and the neuronal motivated dynamic link matching.

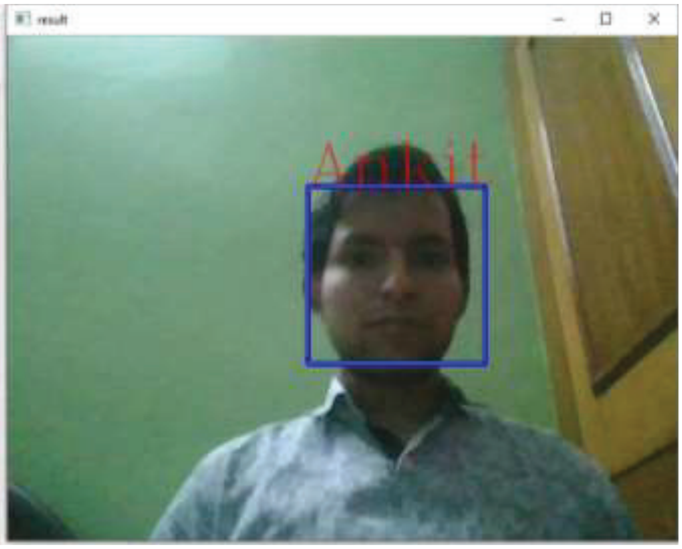
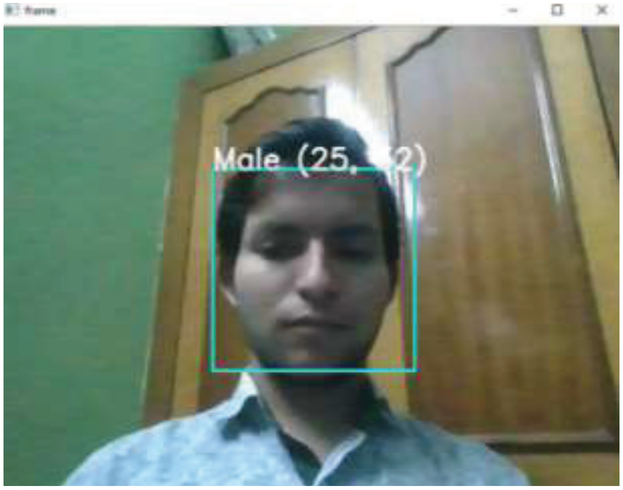


Fig. 3.1: Face Recognition

3.2.2 Gender Recognition

It is used to detect age and gender of users who passes by based on online face analyses. Gender identity is the personal sense of one's own gender. Gender identity can correlate with a person's assigned sex at birth or can differ from it. Gender expression typically reflects a person's gender identity, but this is not always the case. While a person may express behaviors, attitudes, and appearances consistent with a particular gender role, such expression may not necessarily reflect their gender identity. The term gender identity was originally coined by Robert J. Stoller in 1964.

Gender recognition using OpenCV's fisherfaces implementation is quite popular and some of you may have tried or read about it also. But, in this example, I will be using a different approach to recognize gender. This method was introduced by two Israel researchers, Gil Levi and Tal Hassner in 2015. I have used the CNN models trained by them in this example. We are going to use the OpenCV's dnn package which stands for "Deep Neural Networks". In the package, Open CV has provided a class called Net which can be used to populate a neural network. Furthermore, these packages support importing neural network models from well known deep learning frameworks like caffe, tensorflow and torch. The researchers I had mentioned above have published their CNN models as caffe models. Therefore, we will be using the CaffeImporter import that model into our application.



. **Fig 3.2 Gender Recognition**

3.2.3 Object Recognition

Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class in digital images and videos. Well-researched domains of object detection include face detection and pedestrian detection. Object recognition is a key technology behind driverless cars, enabling them to recognize a stop sign or to distinguish a pedestrian from a lamppost. It is also useful in a variety of applications such as disease identification in bioimaging, industrial inspection, and robotic vision.

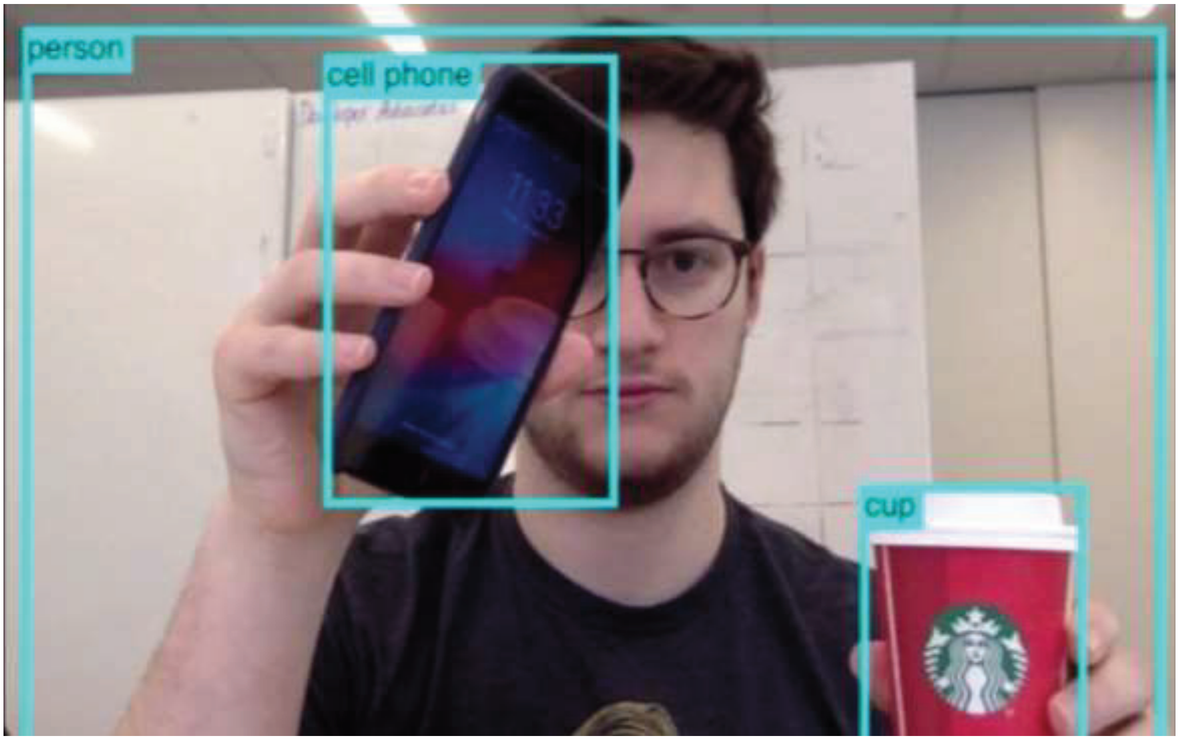


Fig 3.3 Object Recognition

3.2.4 How Object Recognition Works?

Deep learning techniques have become a popular method for doing object recognition. Deep learning models such as convolutional neural networks, or CNNs, are used to automatically learn an object’s inherent features in order to identify that object. For example, a CNN can learn to identify differences between cats and dogs by analyzing thousands of training images and learning the features that make cats and dogs different.

3.2.5 Object Recognition Techniques

(a) Object Recognition Using Deep Learning

There are two approaches to performing object recognition using deep learning:

- **Training a model from scratch**
  To train a deep network from scratch, you gather a very large labeled dataset and design a network architecture that will learn the features and build the model. The results can be impressive, but this approach requires a large amount of training data, and you need to set up the layers and weights in the CNN.

- **Using a pretrained deep learning model**
  Most deep learning applications use the transfer learning approach, a process that involves fine-tuning a pretrained model. You start with an existing network, such as AlexNet or GoogLeNet, and feed in new data containing previously unknown classes. This method is less time-consuming and can provide a faster outcome because the model has already been trained on thousands or millions of images.

Deep learning offers a high level of accuracy but requires a large amount of data to make accurate predictions.

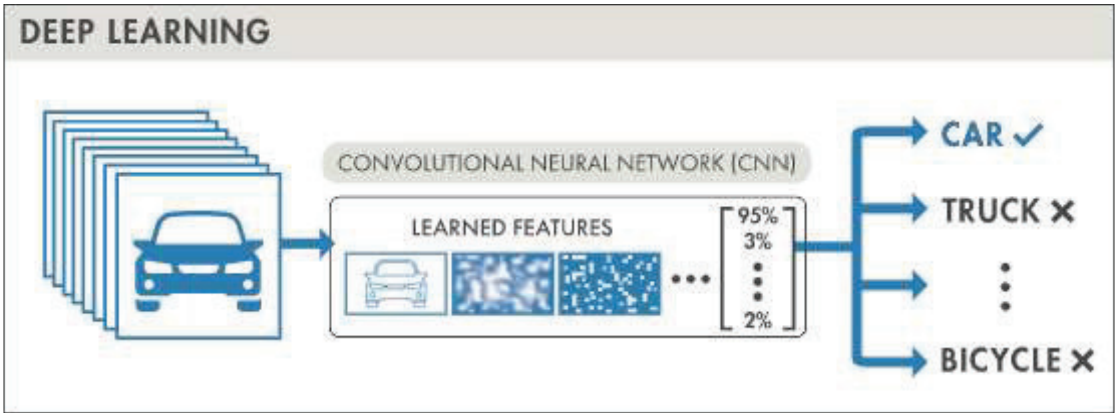


Fig 3.4: Deep learning techniques for object recognition.

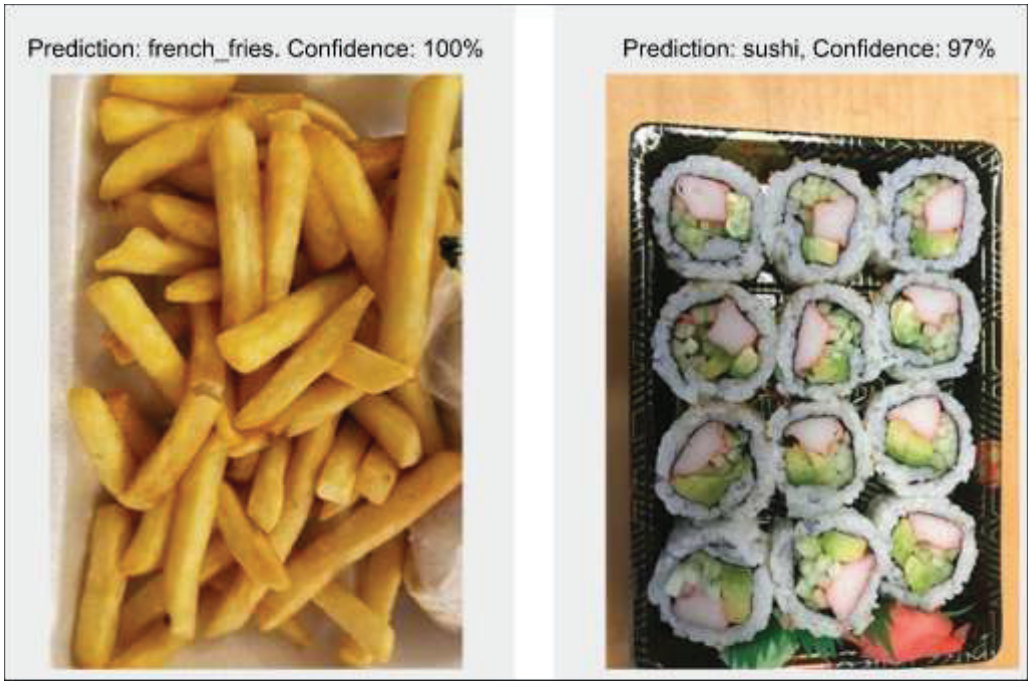

Prediction: french_fries. Confidence: 100%
Prediction: sushi, Confidence: 97%

Fig 3.5: Deep learning application showing object recognition of restaurant food

(b) Object Recognition Using Deep Learning

Machine learning techniques are also popular for object recognition and offer different approaches than deep learning. Common examples of machine learning techniques are:

- HOG feature extraction with an SVM machine learning model
- Bag-of-words models with features such as SURF and MSER
- The Viola-Jones algorithm, which can be used to recognize a variety of objects, including faces and upper bodies

To perform object recognition using a standard machine learning approach, you start with a collection of images (or video), and select the relevant features in each image. For example, a feature extraction algorithm might extract edge or corner features that can be used to differentiate between classes in your data .These features are added to a machine learning model, which will separate these features into their distinct categories, and then use this information when analyzing and classifying new objects. We can use a variety of machine learning algorithms and feature extraction methods, which offer many combinations to create an accurate object recognition model.

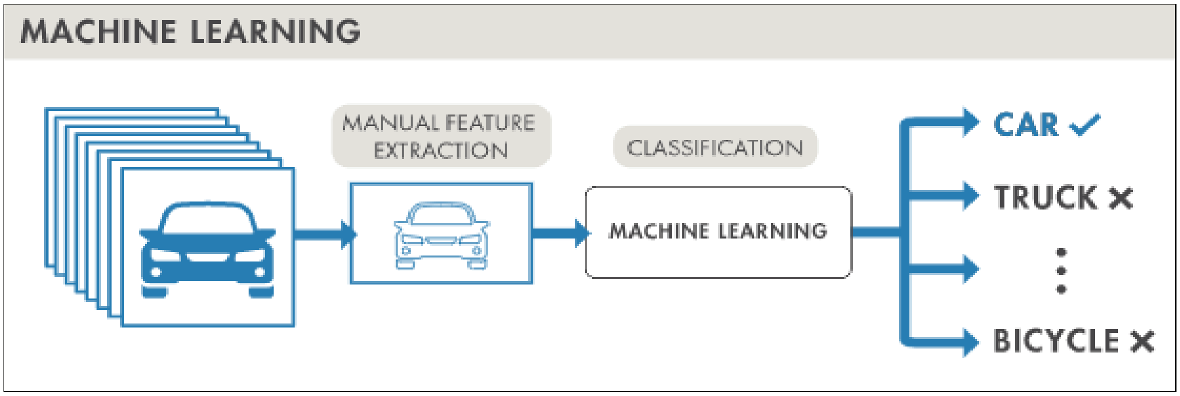


Figure 3.6: Machine learning techniques for object recognition.

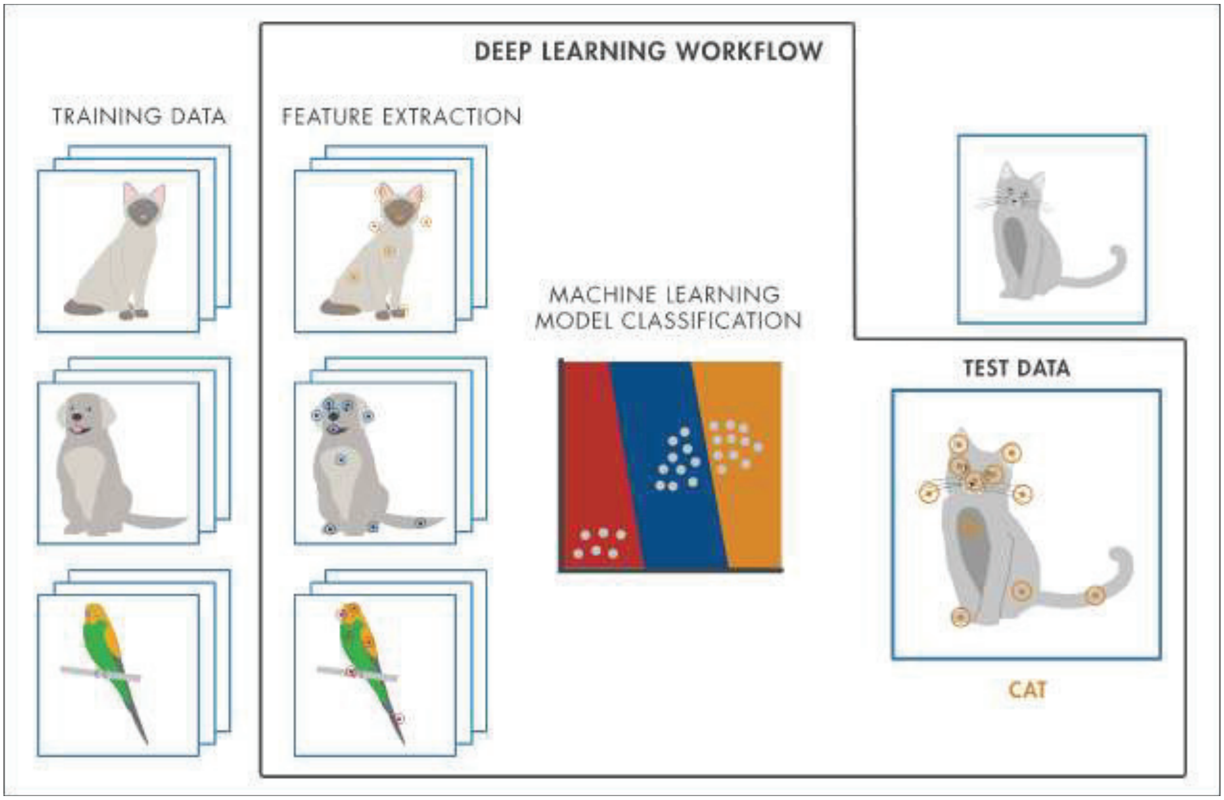


Figure 3.7: Machine learning workflow for object recognition.

3.2.6 Creation of Drone

A Flying robot or a vehicle without a human pilot on a board and a type of unmanned vehicle. UAVs are a component of an unmanned aircraft system; which include a UAV, a ground-based controller, and a system of communications between the two. We have created a Drone which is a flying robot with an embedded system and work in the conjunction with on board sensors and GPS. It enables Object Detection & Face Recognition to which, we have named it as VISION - A Drone for Emergency Operations. It is used to detect and analysis the disaster and count number of people, their gender and prepare the dataset, also detect the real time situation of the area

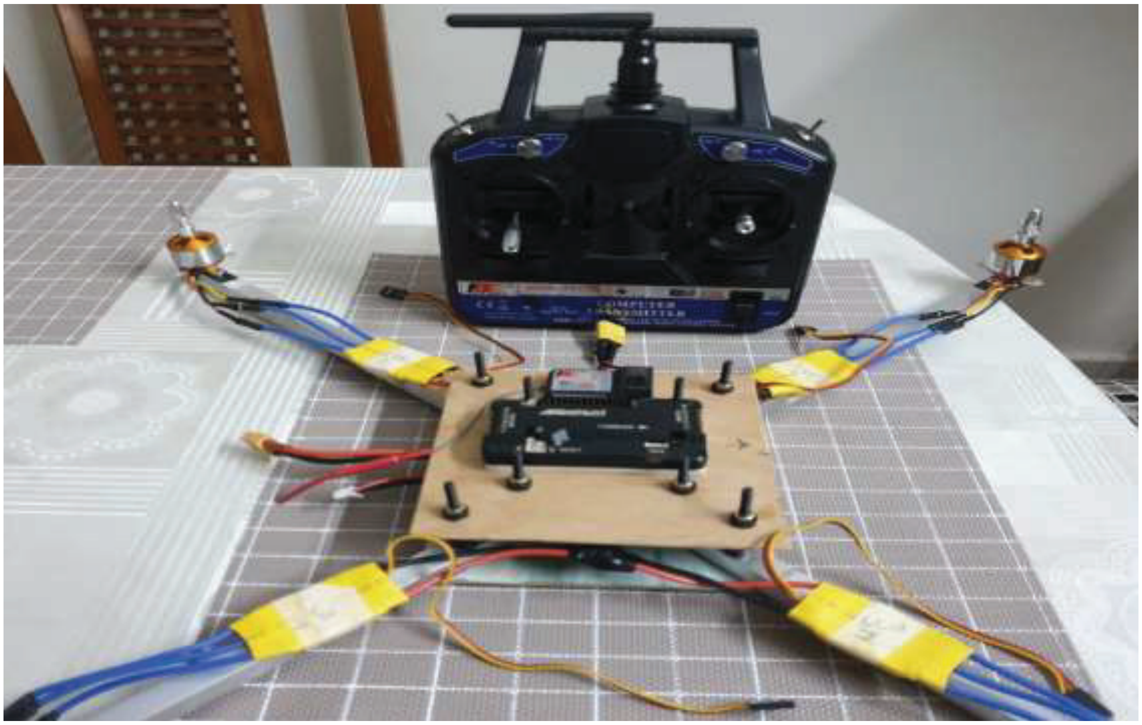


Fig 3.8 Outline of Drone

CHAPTER 4

TECHNICAL SPECIFICATIONS

4.1 HARDWARE REQUIREMENTS

- 2 GB of raspberry pi.
- Pi-Cam.
- 16 gb memory card.
- Usb Cable.
- Apm.
- Quadcopter frame
- Tx^n and rx^n.
- Dual core 64 bit of processor.

4.2 SOFTWARE REQUIREMENTS

- Python 3.6 or above

- Anaconda (Jupyter Notebook or Spyder).
- Raspbian (OS for raspberry pi)

4.3 OTHER SPECIFICATIONS

- **Quadcopter Frame**

  It is the “backbone” of the quadcopter. The frame is what keeps all the parts of the helicopter together. It has to be sturdy, but on the other hand, it also has to be light so that the motors and the batteries don’t struggle to keep it in the air[2].A quadcopter, also called a quadrotor helicopter or quadrotor, is a multirotor helicopter that is lifted and propelled by four rotors. Quadcopters are classified as rotorcraft, as opposed to fixed-wing aircraft, because their lift is generated by a set of rotors (vertically oriented propellers). Quadcopters generally use two pairs of identical fixed pitch propellers;two clockwise (CW) and two counterclockwise (CCW). These use independent variation of the speed of each rotor to achieve control. By changing the speed of each rotor it is possible to specifically generate a desired total thrust; to locate for the centre of thrust both laterally and longitudinally; and to create a desired total torque, or turning force. Quadcopters differ from conventional Helicopters, which use rotors that are able to vary the pitch of their blades dynamically as they move around the rotor hub. In the early days of flight, quadcopters (then referred to either as 'quadrotors' or simply as 'helicopters') were seen as possible solutions to some of the persistent problems in vertical flight. Torque-induced control issues (as well as efficiency issues originating from the tail rotor, which generates no useful lift) can be eliminated by counter-rotation, and the relatively short blades are much easier to construct. A number of manned designs appeared in the 1920s and 1930s. These vehicles were among the first successful heavier-than-air vertical take off and landing (VTOL) . However, early prototypes suffered from poor performance, and latter prototypes required too much pilot work load, due to poor stability augmentation and limited control authority. Around 2005 to 2010, advances in electronics allowed the production of cheap lightweight flight controllers, accelerometers (IMU), global positioning system and cameras. This resulted in the quadcopter configuration becoming popular for small unmanned aerial vehicles. With their small size and maneuverability, these quadcopters can be flown indoors as well as outdoors. At a small size, quadcopters are cheaper and more durable than conventional helicopters due to their mechanical simplicity. Their smaller blades are also advantageous because they possess less kinetic energy, reducing their ability to cause damage. For small-scale quadcopters, this makes the vehicles safer for close interaction. It is also possible to fit quadcopters with guards that enclose the rotors, further reducing the potential for damage. However, as size increases, fixed propeller quadcopters develop disadvantages relative to conventional helicopters. Increasing blade size increases their momentum. This means that changes in blade speed take longer, which negatively impacts control. Helicopters do not experience this problem as increasing the size of the rotor disk does not significantly impact the ability to control blade pitch[4]. Due to their ease of construction and control, quadcopter aircraft are frequently used as amateur model aircraft projects.

- **Motors**

  The thrust that allows the Quadcopter to get airborne is provided by Brushless DC motors and each of them is separately controlled by an electronic speed controller or ESC. A brushless DC electric motor (BLDC motor or BL motor), also known as electronically commutated motor (ECM or EC motor) and synchronous DC motors, are synchronous motors powered by direct current (DC) electricity via an inverter or switching power supply which produces electricity in the form of alternating current (AC) to drive each phase of the motor via a closed loop controller. The controller provides pulses of current to the motor windings that control the speed and torque of the motor. The construction of a brushless motor system is typically similar to a permanent magnet synchronous motor (PMSM), but can also be a switched reluctance motor, or an induction (asynchronous) motor. They may also use neodymium magnets and be outrunners (the stator is surrounded by the rotor) or (the rotor is surrounded by the stator)[5]. The advantages of a brushless motor over brushed motors are high power-to-weight ratio, high speed, electronic control, and low maintenance. Brushless motors find applications in such places as computer peripherals (disk drives, printers), hand-held power tools, and vehicles ranging from model aircraft to automobiles.

Fig 4.1: Brushless Motor

- **ESCs**

  Electronic Speed Controller is like a nerve that delivers the movement information from the brain (flight controller) to the arm or leg muscles (motors). It regulates how much power the motors get, which determines the speed and direction changes of the quad[6]. An electronic speed control follows a speed reference signal (derived from a throttle lever, joystick, or other manual input) and varies the switching rate of a network of field effect transistors (FETs)[1]. By adjusting the duty cycle or switching frequency of the transistors, the speed of the motor is changed. The rapid switching of the transistors is what causes the motor itself to emit its characteristic high-pitched whine, especially noticeable at lower speeds. Different types of speed controls are required for brushed DC motors and brushless DC motors. A brushed motor can have its speed controlled by varying the voltage on its armature. (Industrially, motors with electromagnet field windings instead of permanent magnets can also have their speed controlled by adjusting the strength of the motor field current.) A brushless motor requires a different operating principle. The speed of the motor is varied by adjusting the timing of pulses of current delivered to the several windings of the motor.

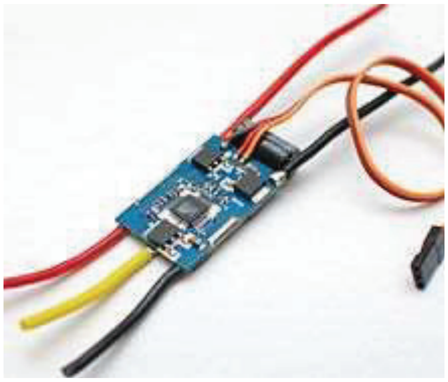


Fig 4.2: generic ESC module rated at 35 amperes with an integrated BEC

Brushless ESC systems basically create three-phase AC power, like a VFD variable frequency drive, to run brushless motors. Brushless motors are popular with radio controlled airplane hobbyists because of their efficiency, power, longevity and light weight in comparison to traditional brushed motors. Brushless DC motor controllers are much more complicated than brushed motor controllers. The correct phase varies with the motor rotation, which is to be taken into account by the ESC: Usually, back EMF from the motor is used to detect this rotation, but variations exist that use magnetic (Hall effect) or optical detectors. Computer-programmable speed controls generally have user-specified options which allow setting low voltage cut-off limits, timing, acceleration, braking and direction of rotation. Reversing the motor's direction may also be accomplished by switching any two of the three leads from the ESC to the motor. Electronic Speed Controllers (ESC) are an essential component of modern quadcopters (and all multirotors) that offer high power, high frequency, high resolution 3-phase AC power to the motors in an extremely compact miniature package. These craft depend entirely on the variable speed of the motors driving the propellers. This wide variation and fine RPM control in motor/prop speed gives all of the control necessary for a quadcopter (and all multirotors) to fly. Quadcopter ESCs usually can use a faster update rate compared to the standard 50 Hz signal used in most other RC applications. A variety of ESC protocols beyond PWM are utilized for modern-day multirotors, including, Oneshot42, Oneshot125, Multishot, and DShot. DShot is a digital protocol that offers a certain advantages over classical analog control, such as higher resolution, CRC checksums, and a lack of oscillator drift (removing the need for calibration). Modern day ESCs protocols can communicate at speeds of 37.5KHz or greater[6].

- **IMU**

  A board that is basically a sum of various sensors that help your quad know where it is and how to level itself. An inertial measurement unit works by detecting the current rate of acceleration using one or more accelerometers. The IMU detects changes in rotational attributes like pitch, roll and yaw using one or more gyroscopes. Some IMU on drones include a magnetometer, mostly to assist calibration against orientation drift.

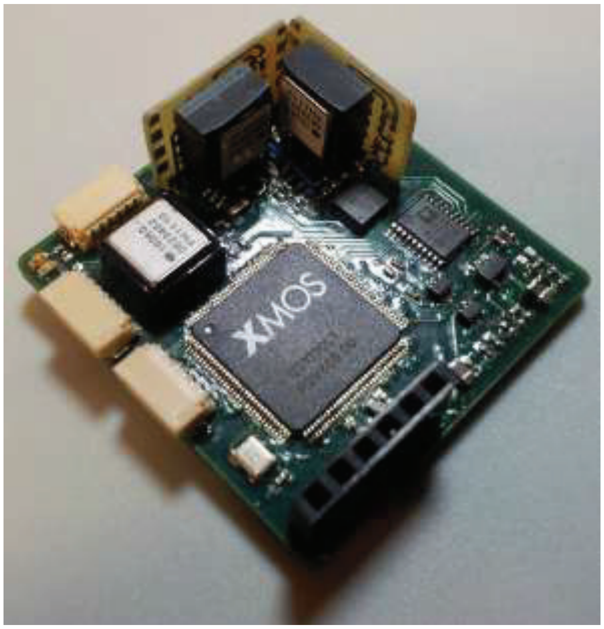


Fig 4.3 IMU Controller

- **RC Controller**

  The choice of the transmitter depends on the choice of the protocol you are going to use and the signal receiver that is on-board the drone. Radio control (often abbreviated to R/C or simply RC) is the use of control signals transmitted by radio to remotely control a device. Examples of simple radio control systems are garage door openers and keyless entry systems for vehicles, in which a small handheld radio transmitter unlocks or opens doors. Radio control is also used for control of model vehicles from a hand-held radio transmitter. Industrial, military, and scientific research organizations make use of radio-controlled vehicles as well. A rapidly growing application is control of unmanned aerial vehicles (UAVs or drones) for both civilian and military uses, although these have more sophisticated control systems than traditional[6].

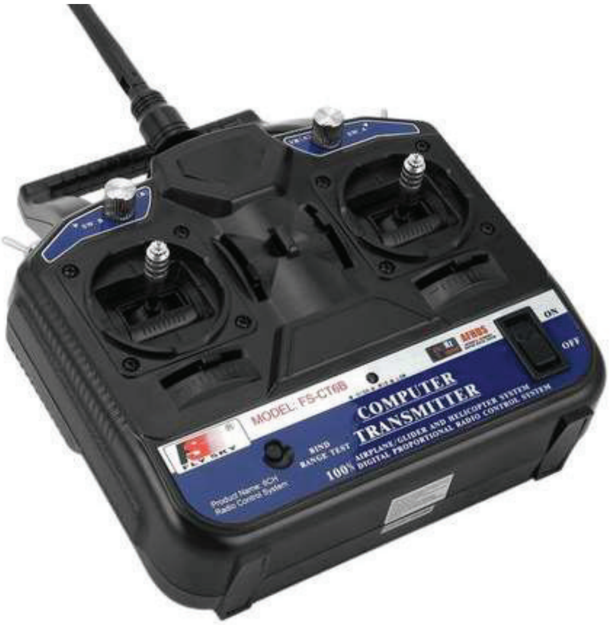

MODEL: FS-CT6B
COMPUTER TRANSMITTER
AIRPLANE/GLIDER AND HELICOPTER SYSTEM
100% DIGITAL PROPORTIONAL RADIO CONTROL SYSTEM
BIND RANGE TEST
ON
OFF
Product Name: 6CH Radio Control System

Fig 4.4 RC Controller

- **Flight Controller**

  It is the brain of the aircraft. It's basically a circuit board with sensors that detects orientation changes of your drone. The flight controller (a.k.a FC) is the brain of the aircraft. It’s a circuit board with a range of sensors that detect movement of the drone, as well as user commands. Using this data, it then controls the speed of the motors to make the craft move as instructed. Nearly all flight controllers have basic sensors such as Gyro (Gyroscopes) and Acc (Accelerometer). Some FC includes more advanced sensors such as Barometer (barometric pressure sensors) and magnetometer (compass). The FC is also a hub for other peripherals, such as GPS, LED, Sonar sensor etc. FPV drone flight controllers are rapidly evolving: becoming smaller, with more features integrated, and using better processors and hardware.

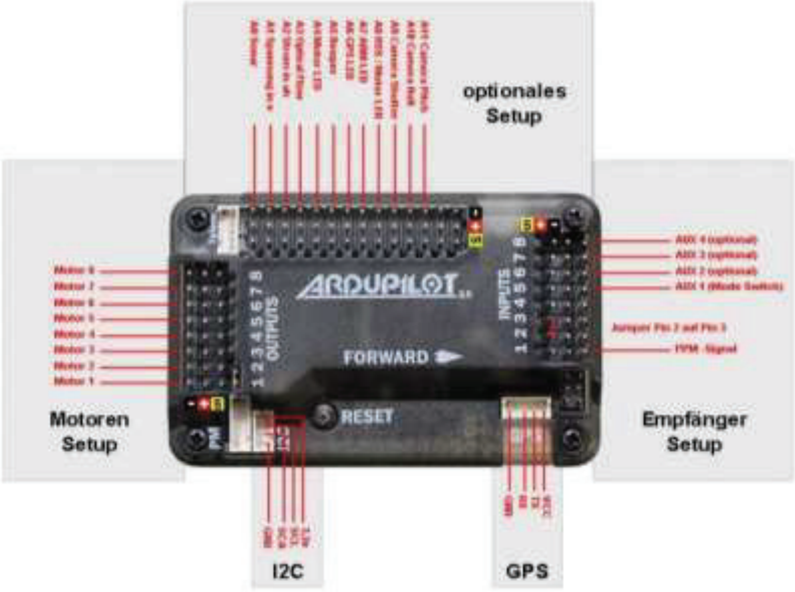


Fig 4.5 Flight Controller

- **Apm (2.8)**

  It is a complete open source autopilot system. The APM 2.8 is a complete open source autopilot system and the bestselling technology that won the prestigious Outback Challenge UAV competition. It allows the user to turn any fixed, rotary wing or multirotor vehicle (even cars and boats) into a fully autonomous vehicle; capable of performing programmed GPS missions with waypoints. This revision of the board has an optional onboard compass, which is designed for vehicles (especially multicopters and rovers) where the compass should be placed as far from power and motor sources as possible to avoid magnetic interference. (On fixed wing aircraft it's often easier to mount APM far enough away from the motors and ESCs to avoid magnetic interference, so this is not as critical, but APM 2.8 gives more flexibility in that positioning and is a good choice for them, too). This is designed to be used with the 3DR uBlox GPS with Compass, so that the GPS/Compass unit can be mounted further from noise sources than APM itself. APM 2.8 requires a GPS unit for full autonomy[7].

- **Power Module**

  It is used For Power Conversion Equipment Such As Industrial Motor Drives, Embedded Motor Drives , terruptible Power Supplies, AC-DC Power Supplies .

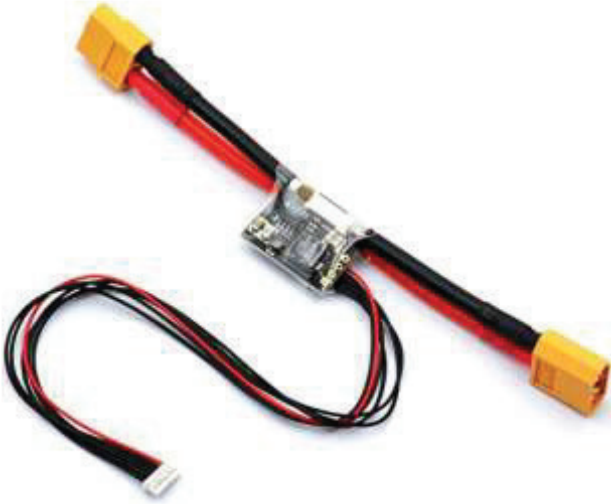


Fig 4.6 Power Module

- **GPS**

  It is used for Locating Position. The Global Positioning System (GPS), originally NAVSTAR GPS, is a satellite-based radio navigation system owned by the United States government and operated by the United States Space Force. It is one of the global navigation satellite systems (GNSS) that provides geolocation and time information to a GPS receiver anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. Obstacles such as mountains and buildings block the relatively weak GPS signals. The GPS does not require the user to transmit any data, and it operates independently of any telephonic or internet reception, though these technologies can enhance the usefulness of the GPS positioning information. The GPS provides critical positioning capabilities to military, civil, and commercial users around the world. The United States government created the system, maintains it, and makes it freely accessible to anyone with a GPS receiver[7].

- **Ultrasonic Sensor**

  It is used for Detecting Ground and Also use In Terrain Follow Mode. Ultrasonic transducers or ultrasonic sensors are a type of acoustic sensor divided into three broad categories: transmitters, receivers and transceivers. Transmitters convert electrical signals into ultrasound, receivers convert ultrasound into electrical signals, and transceivers can both transmit and receive ultrasound. In a similar way to radar and sonar, ultrasonic transducers are used in systems which evaluate targets by interpreting the reflected signals. For example, by measuring the time between sending a signal and receiving an echo the distance of an object can be calculated. Passive ultrasonic sensors are basically microphones that detect ultrasonic noise that is present under certain conditions.The design of transducer can vary greatly depending on its use: those used for diagnostic purposes, for example the range-finding applications listed above, are generally lower power than those used for the purpose of changing the properties of the liquid medium, or targets immersed in the liquid medium, through chemical, biological or physical (e.g. erosive) effects. The latter class include ultrasonic probes and ultrasonic baths, which apply ultrasonic energy to agitate particles, clean, erode or disrupt biological cells[8].

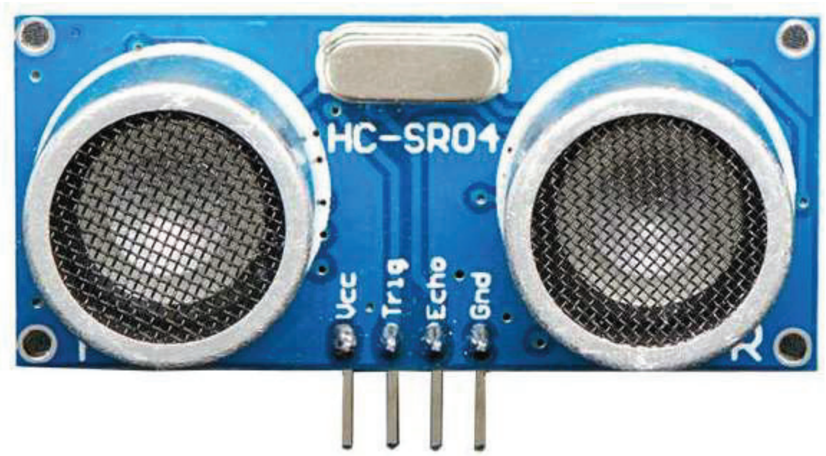


Fig 4.7: UltraSonic Sensors

- **Microphone**

  It is used for Recording Audios. A microphone, colloquially named mic or mike, is a device – a transducer – that converts sound into an electrical signal. Microphones are used in many applications such as telephones, hearing aids, public address systems for concert halls and public events, motion picture production, live and recorded audio engineering, sound recording, two-way radios, megaphones, radio and television broadcasting, and in computers for recording voice, speech recognition, VoIP, and for non-acoustic purposes such as ultrasonic sensors or knock sensors. Several types of microphone are in use, which employ different methods to convert the air pressure variations of a sound wave to an electrical signal. The most common are the dynamic microphone, which uses a coil of wire suspended in a magnetic field; the condenser microphone, which uses the vibrating diaphragm as a capacitor plate; and the piezoelectric microphone, which uses a crystal of piezoelectric material. Microphones typically need to be connected to a preamplifier before the signal can be recorded [9].

- **Pi Cam**

  It is used for Detecting object. PiCAM is an autonomous high definition video system with an embedded web server. It uses the Omnivision 5647 sensor which offers a detailed resolution up to 5MP (megapixel). The PiCAM provides multi-day recordings, depending on the memory capacity of the SD card used for on-system storage. When using the new Pi-Camera-Board V2 with the Sony IMX219 sensor even resolutions up to 8MP are possible. This allows even more brilliant shots with a larger zoom factor. The surveillance area of a PiCAM is very large. Due to the pan and tilt functions almost a 360° view is possible. Details in the visible field can be enlarged using the zoom function. A PiCAM can be used direct[10].

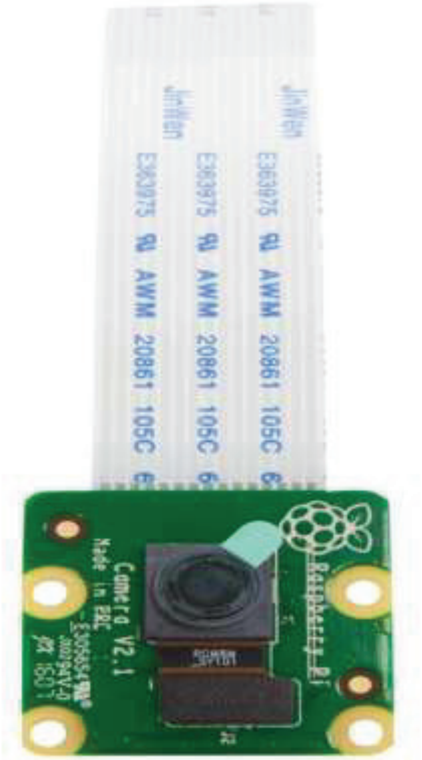


Fig 4.8: Pi-Cam

4.4 TECHNICAL SPECIFICATION

- 3 cell 1,000 mAH LiPo rechargeable battery.
- High pitch propeller for great manoeuvrability
- 4 brushless motors with micro ball bearing and rare earth magnets
- 14.5 watt & 28,500 rpm when hoverin
- Self-lubricating bronze bearings, tempered steel prop shafts.
- Low noise Nylatron gears for 8.625 propeller shafts.
- Emergency stop controlled by software.
- Fully reprogrammable motor controller
- Water resistant electronic motor controller
- Foam to isolate the inertial center from the engine's vibrations.
- EPP hull;
- Carbon fibre tubes
- 380g with outdoor hull
- 420g with indoor hull;
- High grade 30% fibre charged nylon plastic parts.

4.5 CONNECTIVITY

- Wi-Fi
- LAN
- Power Supply
- GPS

4.6 LANGUAGES USED

- Python

4.7 DATABASE DESIGN

Database design is required to manage the large bodies of information. In addition to the database system must provide for the safety of information handled, despite the system crashes due to attempts art unauthorized access. For developing an efficient database, we have used the real time database.

CHAPTER 5

METHODOLOGY

5.1 TECHNOLOGIES USED

5.1.1 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach[9].

5.2 FEATURES AND SPECIFICATIONS

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support Functional programming and aspect-oriented programming (including by meta programming and meta objects. Many other paradigms are supported via extensions, including design by contract and logic programming. Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution, which binds method and variable names during program execution. Python's design offers some support for functional programming in the Lisp tradition. It has filter, map and reduce functions; list comprehensions, dictionaries, sets, and generator expressions. The standard library has two modules that implement functional tools borrowed from Haskell and Standard ML[10].

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Readability counts.

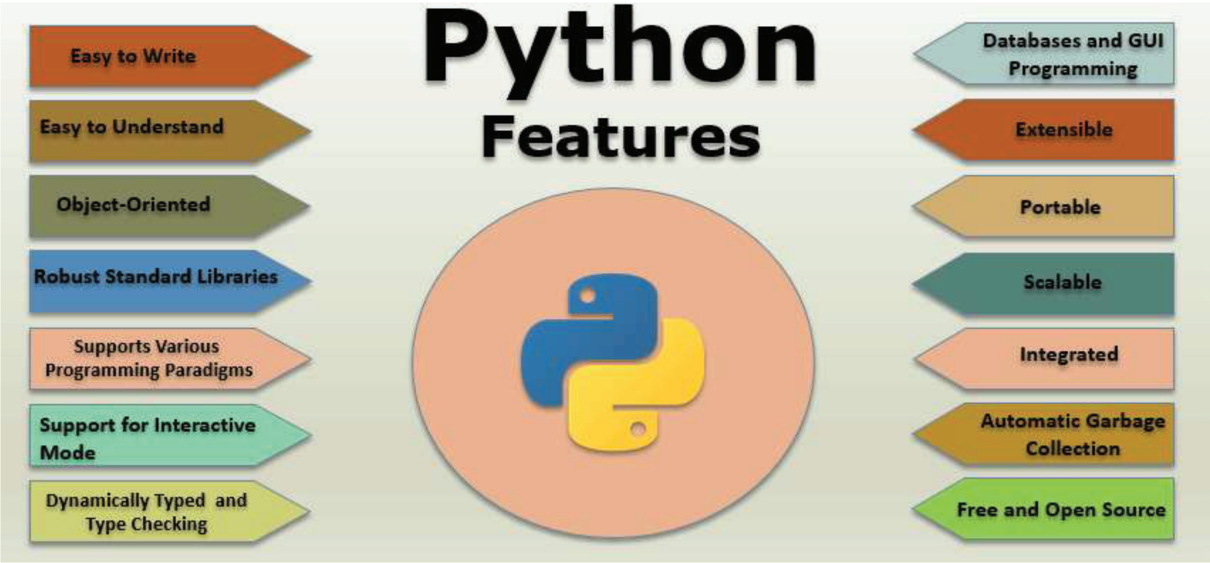

Python
Features
Easy to Write
Easy to Understand
Object-Oriented
Robust Standard Libraries
Supports Various Programming Paradigms
Support for Interactive Mode
Dynamically Typed and Type Checking
Databases and GUI Programming
Extensible
Portable
Scalable
Integrated
Automatic Garbage Collection
Free and Open Source

Figure 5.1: Features and Specifications

5.2.1 Architecture of PyPI

PyPI is a central index of Python projects where people can browse existing projects by category or register their own work. Source or binary distributions can be uploaded and added to an existing project, and then downloaded for installation or study. PyPI also offers web services that can be used by tools like installers[11].

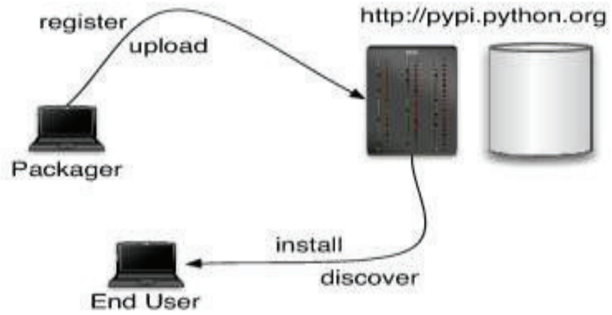


Fig 5.2 PyPI Work Flow

- **Registering Projects and Uploading Distributions**

  Registering a project to PyPI is done with the Distutils register command. It builds a POST request containing the metadata of the project, whatever its version is. The request requires an Authorization header, as PyPI uses Basic Authentication to make sure every registered project is associated with a user that has first registered with PyPI. Credentials are kept in the local Distutils configuration or typed in the prompt every time a register command is invoked.

  An example of its use is:

```
$ python setup.py register
running register
Registering MPTools to http://pypi.python.org/pypi
Server response (200): OK
```

Each registered project gets a web page with an HTML version of the metadata, and packagers can upload distributions to PyPI using **upload**:

```
$ python setup.py sdist upload
running sdist
…
running upload
Submitting dist/mopytools-0.1.tar.gz to http://pypi.python.org/pypi
Server response (200): OK
```

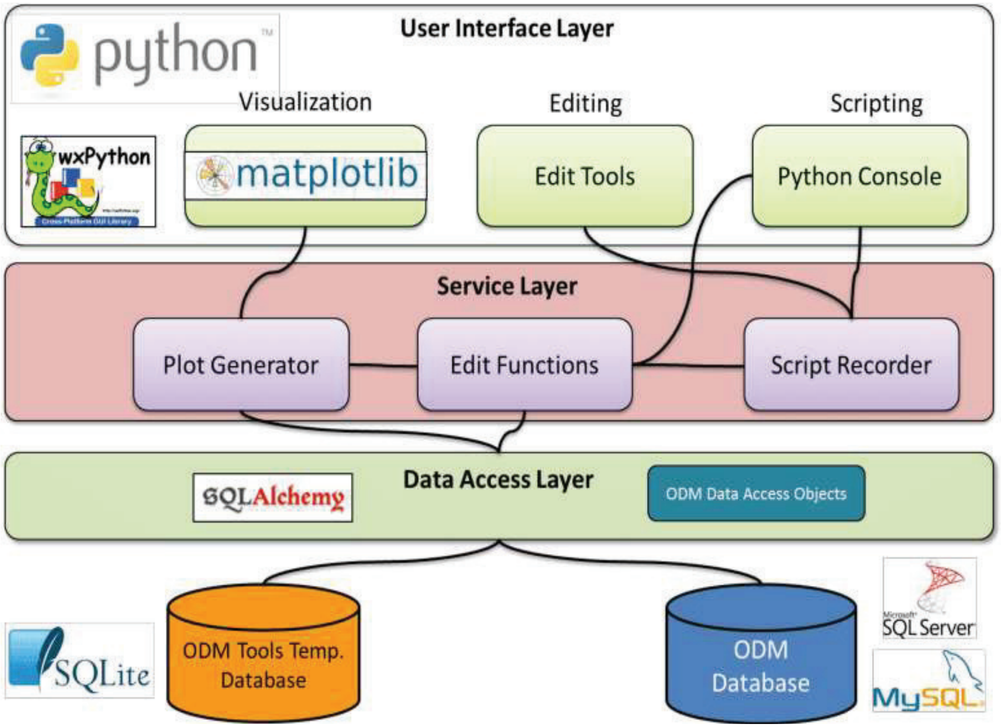


Fig 5.3 Simple Python Architecture

5.3 METHODOLOGY ADOPTED FOR ASSEMBLING OF A DRONE

5.3.1 Working Principle

1. First, we are making a frame of light weight material.
2. Quadcopter is a device with a intense mixture of Electronics, Mechanical and mainly on the principle of Aviation.
3. The Quadcopter has 4 motors whose speed of rotation and the direction of rotation changes according to the users desire to move the device in a particular direction (i.e

Takeoff motion, Landing motion, Forward motion, Backward motion, Left motion, Right Motion.)

4. The rotation of Motors changes as per the transmitted signal send from the 6-Channel transmitter.
5. The signal from microcontroller goes to ESC's which in turn control the speed of motor.

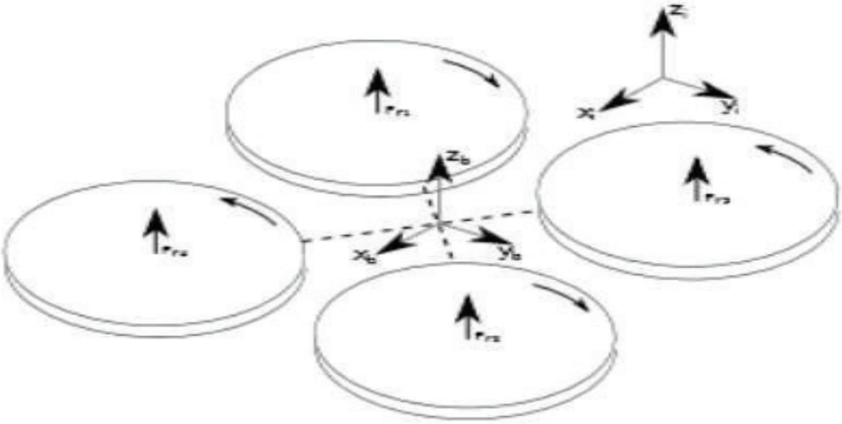


Fig 5.4: Axis of Drone

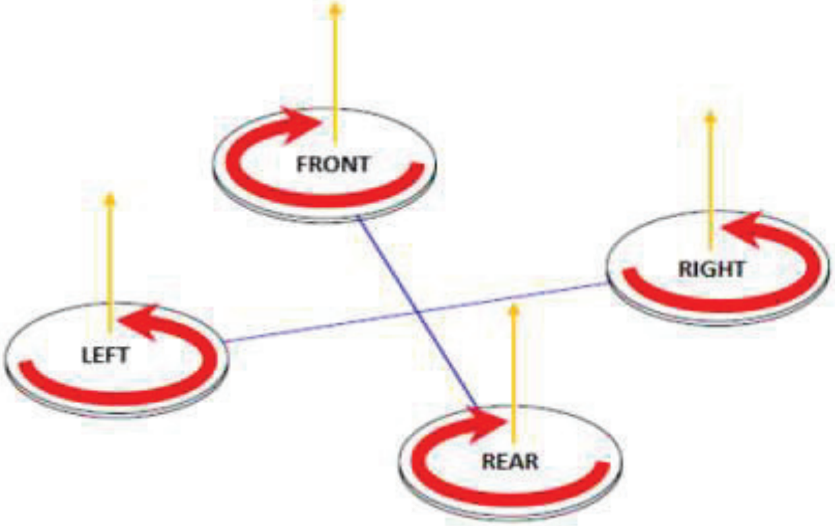


Fig 5.5: Take off Motion

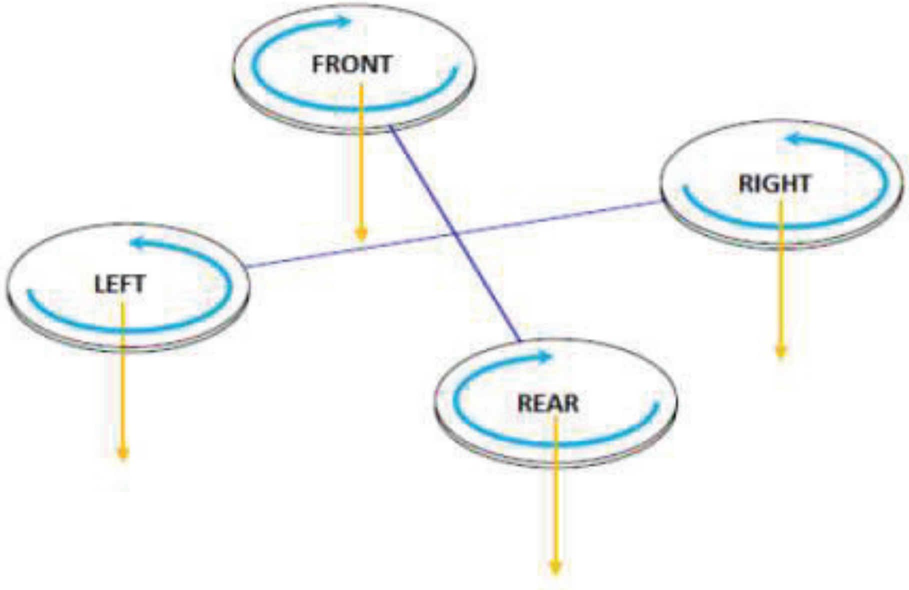


Fig 5.6: Landing Motion

5.3.2 Warning

We should use the Drone safely and responsibly at all times, so as to avoid any damage or harm being caused to any person, animal or property next to which you are flying the Drone. In this respect you should ensure that you always operate the Drone in compliance with this Quick Start Guide and our Safe Use of the Drone instructions. Parrot also reminds you that you should not use the Drone for any unauthorised or unlawful purposes, as you will otherwise be fully liable for any loss or damage caused as a result of such unauthorized use.

5.3.3 How Do Drone Controllers Work?

A drone controller works by sending a radio signal from the remote control to the drone, which tells the drone what to do. Radio signals are sent from the radio transmitter in the drone controller and received by the drone’s receiver. This is why the drone controller is sometimes simply called the drone radio transmitter or the drone radio controller.

5.3.4 Drone Remote Controller Terminology

There are four primary ways to move a drone using your controller:

- **Roll**
  Move your drone left or right in the air, “rolling” your drone.

- **Pitch**
  Tilt your drone forward or backward.

- **Yaw**
  Rotates your drone clockwise or counter clockwise, allowing you to make circles or patterns in the air.

- **Throttle**
  Controls the amount of power sent to your drone, which makes the drone go faster or slower.

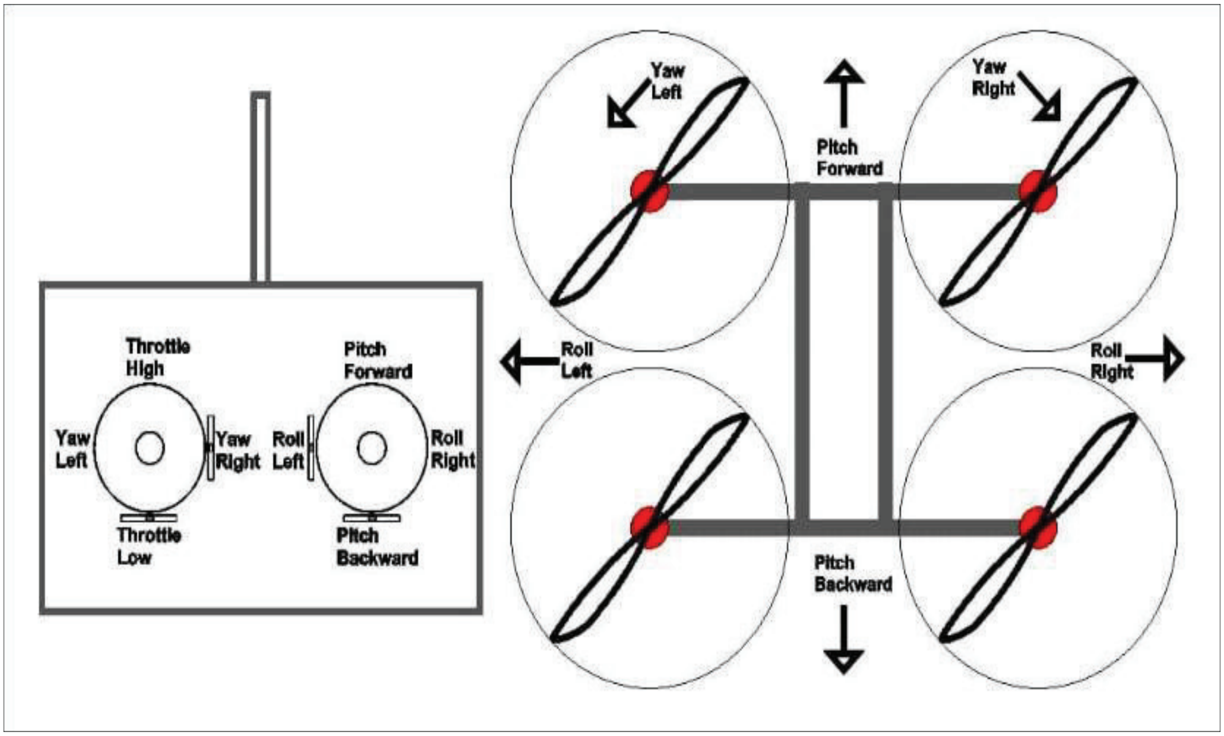


Fig 5.7 Drone Remote Controller Terminology

5.3.5 Basic Drone Controller Features

When flying, the harder you push the stick in a certain direction, the more power will be transmitted to the drone in that direction.

If we're just learning how to fly, push the sticks gently until you get the hang of things. Also, keep in mind that each controller is different. Even if we have some experience flying we'll want to get a feel for a new drone controller before going full throttle.

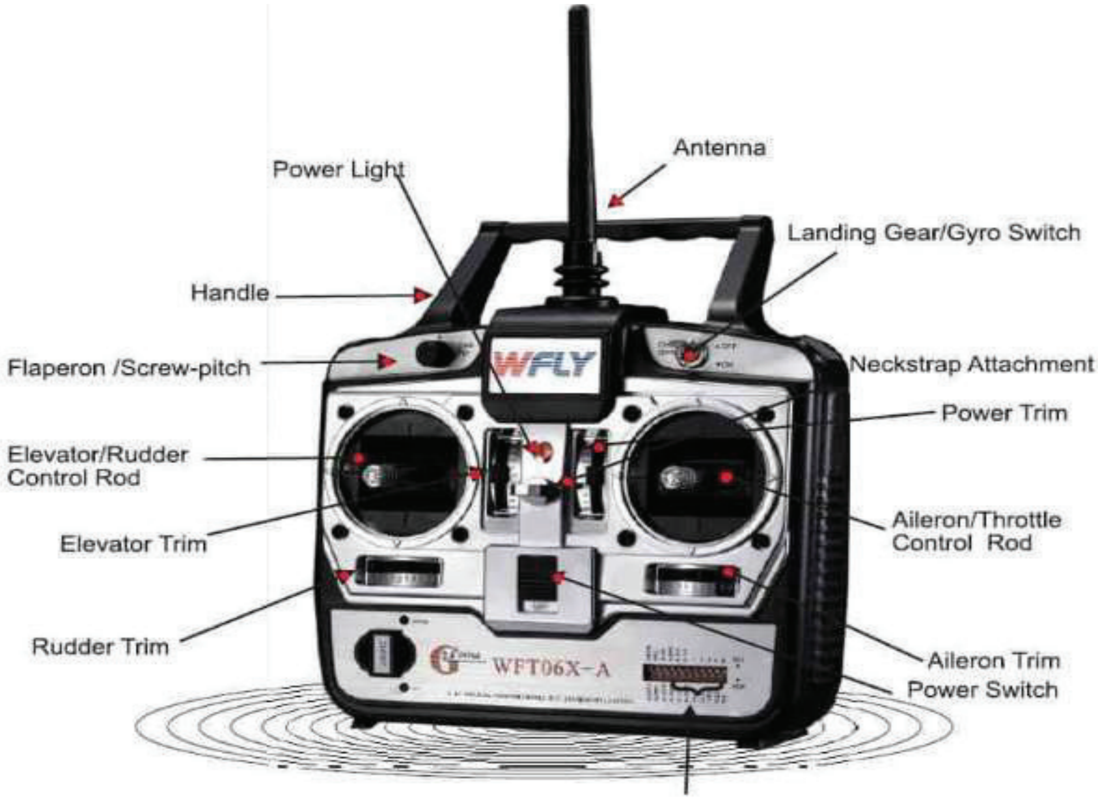


Fig 5.8 Transmitter Features

- **Right Stick**

  The right stick lets you control the roll and pitch of your drone, allowing you to move the drone right/left as well as forward and backward.

- **Left Stick**

  The left stick lets you control the yaw and throttle of your drone, allowing you to control the height at which you fly and letting you rotate the drone clockwise or counter clockwise in flight.

- **Trim Buttons**

  Each drone remote controller has its own trim button.

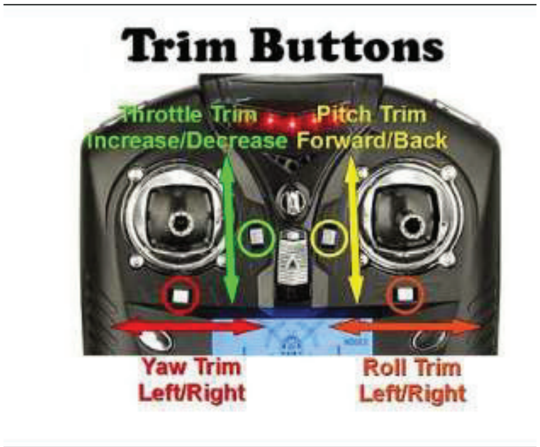


Fig 5.9 Trim Buttons

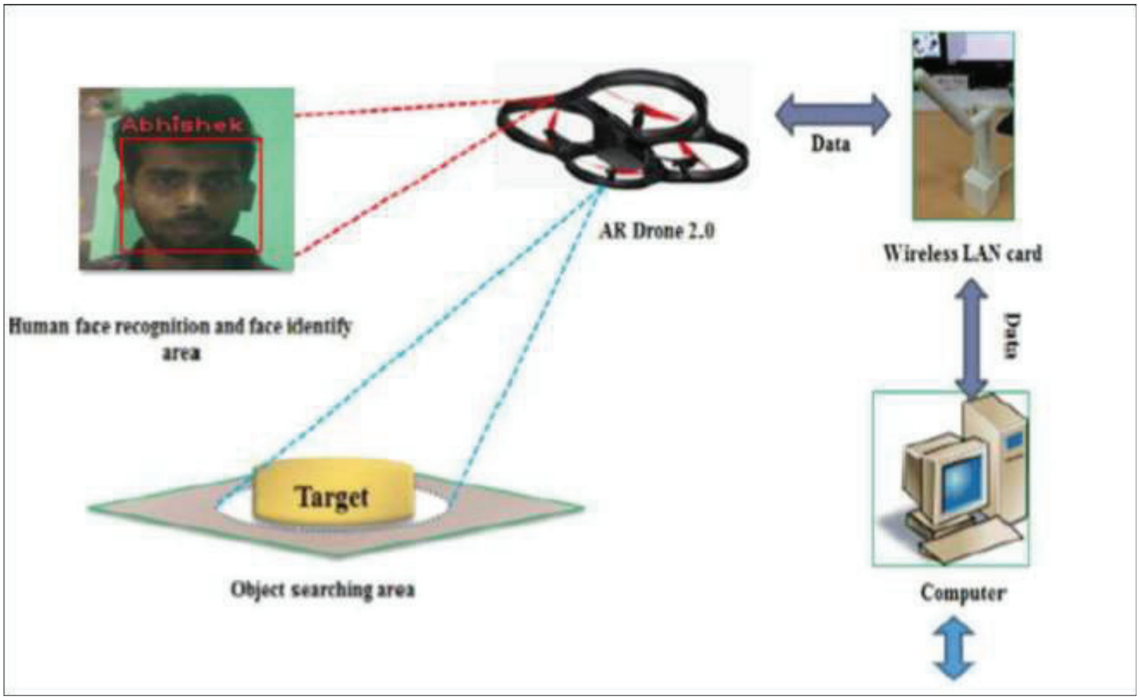


Fig 5.10 Working Diagram of Drone

CHAPTER 6

MACHINE LEARNING

6.1 INTRODUCTION

It is a concept which allows the machine to learn from examples and experience, and that too without being explicitly programmed. So instead of writing the code, we feed data to the generic algorithm, and the algorithm/ machine builds the logic based on the given data. The process of learning begins with observations of data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly. Machine Learning algorithm is trained using a training data set to create a model. When new input data is introduced to the ML algorithm, it makes a prediction on the basis of the model. The prediction is evaluated for accuracy and if the accuracy is acceptable, the Machine Learning algorithm is deployed. If the accuracy is not acceptable, the Machine Learning algorithm is trained again and again with an augmented training data set.

6.2 WHAT IS MACHINE LEARNING

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves.

6.3 ADVANTAGES OF MACHINE LEARNING LANGUAGE

1. Easily Identifies Trends and Patterns

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

2. No Human Intervention Needed (Automation)

With ML, you don’t need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

3. Continuous Improvement

As **ML algorithms** gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

4. Handling Multi-Dimensional And Multi-Variety Data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

5. Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

6.4 DISADVANTAGES OF MACHINE LEARNING

With all those advantages to its powerfulness and popularity, Machine Learning isn’t perfect. The following factors serve to limit it:

1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

4. High Error-Susceptibility

Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

6.5 FUTURE OF MACHINE LEARNING

- **Deeper personalization**

  In the future, users will receive more precise recommendations and ads will become both more effective and less annoying.

- **Neural networks running on our mobile devices**

  Mobile device may have the ability to conduct machine learning tasks locally, opening up a wide range of opportunities for object recognition, speech, face detection, and other innovations for mobile platforms.

- **Real-time speech translation**

  In late 2014 Skype launched Skype Translator. It's been improving the service since then, and currently provides real-time audio translation among seven languages. If this technology continues to develop, it could significantly improve the quality of international communication or even eradicate language barriers.

- **Health and fitness**

  Fitness tracking wearables and apps are pretty popular right now. People gladly use wearables and connected apps to track their sport activities and everyday life. Machine learning has the potential to take this a step further, however, by providing more detailed feedback and tips about a user's activity and condition, making fitness trackers more effective.

- **Prolonging a mobile device's battery life**

  This may sound a lot less epic than other possibilities of machine learning, but preserving battery life is one of the most frustrating concerns for mobile app users. Along with the automation of system resource allocation for apps, machine learning could also reduce the amount of unnecessary battery consumption by apps.

6.6 APPLICATIONS OF MACHINE LEARNING

These are the real world Machine Learning Applications-

6.6.1 Image Recognition:

It is one of the most common machine learning applications. There are many situations where you can classify the object as a digital image. For digital images, the measurements describe the outputs of each pixel in the image.

6.6.2 Speech Recognition

Speech recognition (SR) is the translation of spoken words into text. It is also known as “automatic speech recognition” (ASR), “computer speech recognition”, or “speech to text” (STT). In speech recognition, a software application recognizes spoken words. The measurements in this Machine Learning application might be a set of numbers that represent the speech signal. We can segment the signal into portions that contain distinct words or phonemes.

6.6.3 Medical Diagnosis

ML provides methods, techniques, and tools that can help in solving diagnostic and prognostic problems in a variety of medical domains. It is being used for the analysis of the importance of clinical parameters and of their combinations for prognosis, e.g. prediction of disease progression, for the extraction of medical knowledge for outcomes research, for therapy planning and support, and for overall patient management.

Learning Associations: Learning association is the process of developing insights into various associations between products. A good example is how seemingly unrelated products may reveal an association to one another. Often studying the association between the products people buy. If buyer buys ‘X’, then he or she force to buy ‘Y’.

6.6.4 Prediction:

The current prediction is one of the hottest machine learning algorithms. Let’s take an example of retail, earlier we were able to get insights like sales report last month / year / 5-years / Diwali Christmas. These type of reporting is called as historical reporting. But currently business is 21 more interested in finding out what will be my sales next month / year / Diwali, etc. So that business can take a required decision (related to procurement, stocks, etc.) on time.

6.7 TYPES OF MACHINE LEARNING

Machine learning is broadly classified into three categories

1. Supervised Machine Learning
2. Unsupervised Machine Learning
3. Reinforcement Machine Learning

6.7.1 SUPERVISED MACHINE LEARNING

Supervised learning is where we have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output. This is a process of an algorithm learning from training dataset. Supervised learning is where we generate a mapping function between input variable (X) and an output variable (Y) and we can use algorithm to generate a function between them.

Supervised learning as the name indicates the presence of a supervisor as a teacher. Basically Supervised learning is a learning in which we teach or train the machine using data which is well labeled that means some data is already tagged with the correct answer. After that, the machine is provided with a new set of examples(data) so that supervised learning algorithm analyses the training data(set of training examples) and produces a correct outcome from labeled data. For instance, suppose we are given an basket filled with different kinds of fruits. Now the first step is to trace in the machine with all different fruits one by one like this:

- If shape of object is rounded and depression at top having color red then it will be labelled as –**Apple.**
- If shape of object is long curving cylinder having color green-yellow then it will be labelled as –**Banana.**

Now suppose after training the data, we have given a new separate fruit say Banana from basket and asked to identify it. Since the machine has already learned the things from previous data and this time have to use it wisely. It will first classify the fruit with its shape and color and would confirm the fruit name as BANANA and put it in Banana category. Thus, the machine learns the things from training data (basket containing fruits) and then apply the knowledge to test data (new fruit).

Supervised learning classified into two categories of algorithms:

- **Classification:** A classification problem is when the output variable is a category, such as “Red” or “blue” or “disease” and “no disease”.

- **Regression**: A regression problem is when the output variable is a real value, such as “dollars” or “weight”.

Some popular examples of supervised machine learning algorithms are:

- Linear Regression for regression problems
- Random forest for classification and regression problems
- Support vector machines for classification problems

6.7.2 UNSUPERVISED MACHINE LEARNING

Categorize the above picture into two parts. First may contain all pics having dogs in it and second part may contain all pics having cats in it. Unsupervised learning is where we only have input data (X) and no corresponding output variables. The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data. These are called unsupervised learning because unlike supervised learning above there is no correct answers and there is no teacher [12]. Algorithms are left to their own devises to discover and present the interesting structure in the data.

Unsupervised learning is the training of machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance. Here the task of machine is to group unsorted information according to similarities, patterns and differences without any prior training of data. Unlike supervised learning, no teacher is provided that means no training will be given to the machine. Therefore, machine is restricted to find the hidden structure in unlabeled data by our-self. For instance, suppose it is given an image having both dogs and cats which have not seen ever.

Figure 6.1: Dogs and Cats

Thus, the machine has no idea about the features of dogs and cat so we can’t categorize it in dogs and cats. But it can categorize them according to their similarities, patterns, and differences.

Unsupervised learning problems can be further grouped into clustering and association problems.

- **Clustering:** A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.
- **Association:** An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y. Some popular examples of unsupervised learning algorithms are:

    - k-means for clustering problems.
    - Apriori algorithm for association rule learning problems.

6.7.3 REINFORCEMENT MACHINE LEARNING

Reinforcement Machine Learning is a learning by interacting with space or an environment. An Rl agent learns from the consequences of its actions, rather than being taught explicitly. It select its actions from its past experiences and also by new choices[13].

Reinforcement learning refers to goal-oriented algorithms, which learn how to attain a complex objective (goal) or maximize along a particular dimension over many steps; for example, maximize the points won in a game[14].

They can start from a blank slate, and under the right conditions they achieve superhuman performance. Like a child incentivized by spankings and candy, these algorithms are penalized when they make the wrong decisions and rewarded when they make the right ones – this is reinforcement.

Reinforcement learning algorithms can be expected to perform better and better in more ambiguous, real-life environments while choosing from an arbitrary number of possible actions, rather than from the limited options of a video game. That is, with time we expect them to be valuable to achieve goals in the real world.

6.8 MACHINE LEARNING LIBRARIES

6.8.1 Pandas

In computer programming, pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals[15].

LIBRARY FEATURES

1. Data Frame object for data manipulation with integrated indexing.
2. Tools for reading and writing data between in-memory data structures and different file formats.
3. Data alignment and integrated handling of missing data.
4. Reshaping and pivoting of data sets.
5. Label-based slicing, fancy indexing, and sub setting of large data sets

6.8.2 Numpy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numpy array into Numeric, with extensive modifications. NumPy is open-source software and has many contributor[15].

The core functionality of NumPy is its "ndarray", for n-dimensional array, data structure. These arrays are strided views on memory. In contrast to Python's built-in list data structure (which, despite the name, is a dynamic array), these arrays are homogeneously typed. All elements of a single array must be of the same type. Such arrays can also be views into memory buffers allocated by C/C++, Cython, and Fortran extensions to the CPython interpreter without the need to copy data around, giving a degree of compatibility with existing numerical libraries. This functionality is exploited by the SciPy package, which wraps a number of such libraries (notably BLAS and LAPACK). NumPy has built-in support for memory-mapped ndarrays.

6.8.3 Scipy

SciPy is a free and open-source Python library used for scientific computing and technical computing. SciPy contains modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers and other tasks common in science and engineering. SciPy builds on the NumPy array object and is part of the NumPy stack which includes tools like Matplotlib, pandas and SymPy, and an expanding set of scientific computing libraries. This NumPy stack has similar users to other applications such as MATLAB, GNU Octave, and Scilab. The NumPy stack is also sometimes referred to as the SciPy stack.

SciPy is also a family of conferences for users and developers of these tools: SciPy (in the United States), EuroSciPy (in Europe) and SciPy.in (in India). Enthought originated the SciPy conference in the United States and continues to sponsor many of the international conferences as well as host the SciPy website.

6.8.4 Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+ [15]. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged.SciPy makes use of Matplotlib.

Matplotlib was originally written by John D. Hunter, has an active development community and is distributed under a BSD-style license. Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012 and further joined by Thomas Caswell.

As of 23 June 2017, matplotlib 2.0.x supports Python versions 2.7 through 3.6. Python3 support started with Matplotlib 1.2. Matplotlib 1.4 is the last version to support Python 2.6.

6.8.5 Scikit-Learn

Scikit -learn (formerly scikits .learn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy[15].

6.8.6 Opencv

OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision.] Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel. The library is cross-platform and free for use under the open-source BSD license.

OpenCV supports the deep learning frameworks TensorFlow, Torch/PyTorch and Caffe.

1. Advance vision research by providing not only open but also optimized code for basic vision infrastructure. No more reinventing the wheel.
2. Disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable.
3. Advance vision-based commercial applications by making portable, performance optimized code available for free – with a license that did not require code to be open or free itself.

Facial recognition system

Figure 6.2: Facial recognition system

Swiss European surveillance: face recognition and vehicle make, model, color and license plate reader. A facial recognition system is a technology capable of identifying or verifying a person from a digital image or a video frame from a video source. There are multiple methods in which facial recognition systems work, but in general, they work by comparing selected facial features from given image with faces within a database.

Facial recognition system is also described as a Biometric Artificial Intelligence based application that can uniquely identify a person by analysing patterns based on the person's facial textures and shape[16].

Iris recognition

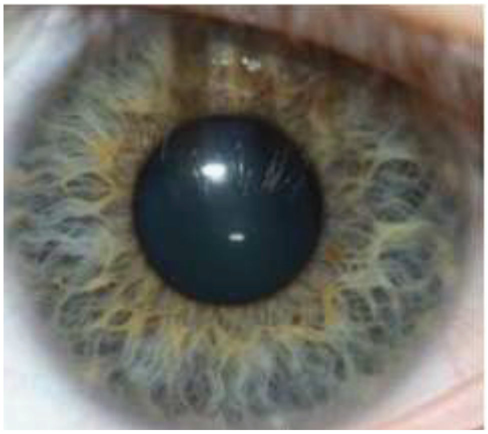


Figure 6.3: Iris Recognition

Iris recognition biometric systems apply mathematical pattern-recognition techniques to images of the irises of an individual's eyes. Iris recognition is an automated method of biometric identification that uses mathematical pattern-recognition techniques on video images of one or both of the irises of an individual's eyes, whose complex patterns are unique, stable, and can be seen from some distance. Retinal scanning is a different, ocular-based biometric technology that uses the unique patterns on a person's retina blood vessels and is often confused with iris recognition.

Iris recognition uses video camera technology with subtle near infrared illumination to acquire images of the detail-rich, intricate structures of the iris which are visible externally. Digital templates encoded from these patterns by mathematical and statistical algorithms allow the identification of an individual or someone pretending to be that individual[16]. Databases of enrolled templates are searched by matcher engines at speeds measured in the millions of templates per second per (single-core) CPU, and with remarkably low false match rates.

6.8.7 Seaborn

Statistical data visualization Graph Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. For a brief introduction to the ideas behind the library, programmer can read the introductory notes by visiting the installation page to see how you can download the package.

CHAPTER 7

TOOLS

7.1 PYTHON IDLE

IDLE (short for Integrated DeveLopment Environment or Integrated Development and Learning Environment) is an integrated development environment for Python, which has been bundled with the default implementation of the language since 1.5.2b1. It is packaged as an optional part of the Python packaging with many Linux distributions. It is completely written in Python and the Tkinter GUI toolkit (wrapper functions for Tcl/Tk).

IDLE is intended to be a simple IDE and suitable for beginners, especially in an educational environment. To that end, it is cross-platform, and avoids feature clutter.

According to the included README, its main features are:

- Multi-window text editor with syntax highlighting, autocompletion, smart indent and other.
- Python shell with syntax highlighting.
- Integrated debugger with stepping, persistent breakpoints, and call stack visibility.

IDLE has been criticized for various usability issues, including losing focus, lack of copying to clipboard feature, lack of line numbering options, and general user interface design; it has been called a "disposable" IDE, because users frequently move on to a more advanced IDE as they gain experience.

7.2 SPYDER

An integrated development environment (IDE) facilitates computer programmers by integrating fundamental tools (e.g., code editor, compiler, and debugger) into a single software package. Users do not need to install the language’s compiler/interpreter on their machines; an IDE provides the environment itself.

Spyder is a dedicated IDE for Python. It incorporates some useful features that make it a popular IDE.

Figure 7.1 Spyder

7.2.1 Features

The following are some of the more salient features of Spyder:

- It analyzes code to provide automatic code completion, horizontal/vertical splitting, and a go-to-definition.
- It is specifically designed for data scientists; hence, it integrates well with data science libraries like NumPy.
- It allows you to run the IPython console.
- It includes a powerful debugger.
- It contains an integrated documentation browser.

Some pros and cons of using Spyder are listed below:

Pros

- It is a light-weight IDE.
- Its features, like online help, are great for beginners.

Cons

- It is not very customizable.
- It’s more basic than its counterparts (e.g., PyCharm).

7.3 JUPYTER

In this case, "notebook" or "notebook documents" denote documents that contain both code and rich text elements, such as figures, links, equations, ... Because of the mix of code and text elements, these documents are the ideal place to bring together an analysis description, and its results, as well as, they can be executed perform the data analysis in real time.

The Jupyter Notebook App produces these documents. We'll talk about this in a bit. For now, you should know that "Jupyter" is a loose acronym meaning Julia, Python, and R. These programming languages were the first target languages of the Jupyter application, but nowadays, the notebook technology also supports many other languages. The main components of the whole environment are, on the one hand, the notebooks themselves and the application. On the other hand, you also have a notebook kernel and a notebook dashboard.
Let's look at these components in more detail.

7.3.1 What is the Jupyter Notebook App?

As a server-client application, the Jupyter Notebook App allows you to edit and run your notebooks via a web browser. The application can be executed on a PC without Internet access, or it can be installed on a remote server, where you can access it through the Internet. Its two main components are the kernels and a dashboard.

A kernel is a program that runs and introspects the user's code. The Jupyter Notebook App has a kernel for Python code, but there are also kernels available for other programming languages.

The dashboard of the application not only shows you the notebook documents that you have made and can reopen but can also be used to manage the kernels: you can which ones are running and shut them down if necessary.

7.4 WHAT IS PYCHARM?

PyCharm is a hybrid-platform developed by JetBrains as an IDE for Python. It is commonly used for Python application development. Some of the unicorn organizations such as Twitter, Facebook, Amazon, and Pinterest use PyCharm as their Python IDE!
It supports two versions: v2.x and v3.x.

We can run PyCharm on Windows, Linux, or Mac OS. Additionally, it contains modules and packages that help programmers develop software using Python in less time and with minimal effort. Further, it can also be customized according to the requirements of developers.

7.4.1 Features of Pycharm

Why should we use it for our next Python project?

Below, we have compiled some of the essential features provided by PyCharm.

1. Intelligent Code Editor:

- It helps us write high-quality codes!
- It consists of color schemes for keywords, classes, and functions. This helps increase the readability and understanding of the code.
- It helps identify errors easily.
- It provides the autocomplete feature and instructions for the completion of the code.

2. Code Navigation:

- It helps developers in editing and enhancing the code with less effort and time.
- With code navigation, a developer can easily navigate to a function, class, or file.
- A programmer can locate an element, a symbol, or a variable in the source code within no time.
- Using the lens mode, further, a developer can thoroughly inspect and debug the entire source code.

3. Refactoring

- It has the advantage of making efficient and quick changes to both local and global variables.
- Refactoring in PyCharm enables developers to improve the internal structure without changing the external performance of the code.
- It also helps split up more extended classes and functions with the help of the extract method.

4. Assistance for Many Other Web Technologies:

- It helps developers create web applications in Python.
- It supports popular web technologies such as HTML, CSS, and JavaScript.
- Developers have the choice of live editing with this IDE. At the same time, they can preview the created/updated web page.
- The developers can follow the changes directly on a web browser.
- PyCharm also supports AnglularJS and NodeJS for developing web applications.

5. Support for Popular Python Web Frameworks

- PyCharm supports web frameworks such as Django.
- It provides the autocomplete feature and suggestions for the parameters of Django.
- It helps in debugging the codes of Django.
- It also assist web2py and Pyramid, the other popular web frameworks.

6. Assistance for Python Scientific Libraries

- PyCharm supports Python's scientific libraries such as Matplotlib, NumPy, and Anaconda.
- These scientific libraries help in building projects of Data Science and Machine Learning.
- It consists of interactive graphs that help developers understand data.
- It is capable of integrating with various tools such as IPython, Django, and Pytest. This integration helps innovate unique solutions[15].

7.5 ANACONDA

1. Anaconda is a python distribution, with installation and package management tools.
2. It provides large selection of packages and commercial support. It is an environment manager, which provides the facility to create different python environments, each with their own settings.
3. It also provides much greater advantages in the data science platform.

7.5.1 CONDA

1. "Conda", the Anaconda's own **package manager**, is used for updating anaconda and its packages.
2. Conda is a **cross platform package and environment manager**. It provides installing, executing and updating different packages along with their dependencies.

CHAPTER 8

DEEP LEARNING

8.1 INTRODUCTION

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It’s achieving results that were not possible before.

In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers.

8.2 HOW DOES DEEP LEARNING ATTAIN SUCH IMPRESSIVE RESULTS?

In a word, accuracy Deep Learning achieves recognition accuracy at higher levels than ever before. This helps consumer electronics meet user expectations, and it is crucial for safety-critical applications like driverless cars. Recent advances in deep learning have improved to the point where deep learning outperforms humans in some tasks like classifying objects in images.

While deep learning was first theorized in the 1980s, there are two main reasons it has only recently become useful:

1. Deep learning requires large amounts of labeled data. For example, driverless car development requires millions of images and thousands of hours of video.
2. Deep learning requires substantial computing power. High-performance GPUs have a parallel architecture that is efficient for deep learning. When combined with clusters or cloud computing, this enables development teams to reduce training time for a deep learning network from weeks to hours or less.

8.3 EXAMPLES OF DEEP LEARNING AT WORK

Deep learning applications are used in industries from automated driving to medical devices.

Automated Driving

Automotive researchers are using deep learning to automatically detect objects such as stop signs and traffic lights. In addition, deep learning is used to detect pedestrians, which helps decrease accidents.

Aerospace and Defense

Deep learning is used to identify objects from satellites that locate areas of interest, and identify safe or unsafe zones for troops.

Medical Research

Cancer researchers are using deep learning to automatically detect cancer cells. Teams at UCLA built an advanced microscope that yields a high-dimensional data set used to train a deep learning application to accurately identify cancer cells.

Industrial Automation

Deep learning is helping to improve worker safety around heavy machinery by automatically detecting when people or objects are within an unsafe distance of machines.

Electronics

Deep learning is being used in automated hearing and speech translation. For example, home assistance devices that respond to your voice and know your preferences are powered by deep learning applications.

8.4 HOW DEEP LEARNING WORKS?

Most deep learning methods use neural network architectures, which is why deep learning models are often referred to as deep neural networks. The term “deep” usually refers to the number of hidden layers in the neural network. Traditional neural networks only contain 2-3 hidden layers, while deep networks can have as many as 150. Deep learning models are trained by using large sets of labeled data and neural network architectures that learn features directly from the data without the need for manual feature extraction.

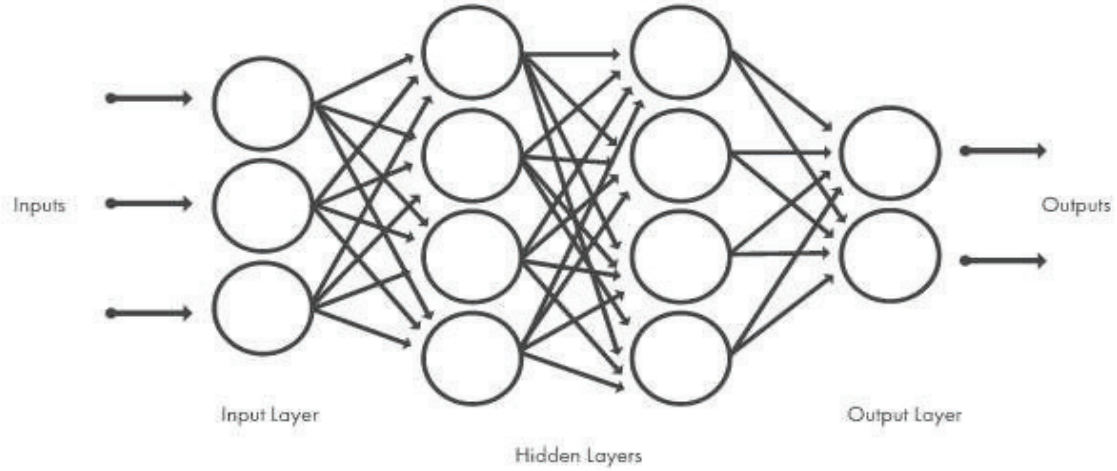


Figure 8.1: Neural networks

8.5 HOW TO CREATE AND TRAIN DEEP LEARNING MODELS?

The three most common ways people use deep learning to perform object classification are :

- **Training from Scratch**
  To train a deep network from scratch, you gather a very large labeled data set and design a network architecture that will learn the features and model. This is good for new applications, or applications that will have a large number of output categories.[17].

- **Transfer Learning**
  Most deep learning applications use the transfer learning approach, a process that involves fine-tuning a pretrained model. You start with an existing network, such as

AlexNet or GoogLeNet, and feed in new data containing previously unknown classes. After making some tweaks to the network, you can now perform a new task, such as categorizing only dogs or cats instead of 1000 different objects. This also has the advantage of needing much less data (processing thousands of images, rather than millions), so computation time drops to minutes or hours. Transfer learning requires an interface to the internals of the pre-existing network, so it can be surgically modified and enhanced for the new task[18]. MATLAB has tools and functions designed to help you do transfer learning.

- **Feature Extraction**

  A slightly less common, more specialized approach to deep learning is to use the network as a **feature extractor**. Since all the layers are tasked with learning certain features from images, we can pull these features out of the network at any time during the training process.

8.6 WHAT'S THE DIFFERENCE BETWEEN MACHINE LEARNING AND DEEP LEARNING?

Deep learning is a specialized form of machine learning. A machine learning workflow starts with relevant features being manually extracted from images. The features are then used to create a model that categorizes the objects in the image. With a deep learning workflow, relevant features are automatically extracted from images. In addition, deep learning performs “end-to-end learning” – where a network is given raw data and a task to perform, such as classification, and it learns how to do this automatically.

Another key difference is deep learning algorithms scale with data, whereas shallow learning converges. Shallow learning refers to machine learning methods that plateau at a certain level of performance when you add more examples and training data to the network. A key advantage of deep learning networks is that they often continue to improve as the size of your data increases.

8.7 WHAT IS TENSORFLOW?

Currently, the most famous deep learning library in the world is Google's TensorFlow. Google product uses machine learning in all of its products to improve the search engine, translation, image captioning or recommendations.

To give a concrete example, Google users can experience a faster and more refined the search with AI. If the user types a keyword the search bar, Google provides a recommendation about what could be the next word.

Google wants to use machine learning to take advantage of their massive datasets to give users the best experience. Three different groups use machine learning:

- Researchers
- Data scientists
- Programmers.

They can all use the same toolset to collaborate with each other and improve their efficiency. Google does not just have any data; they have the world's most massive computer, so Tensor Flow was built to scale. Tensor Flow is a library developed by the Google Brain Team to accelerate machine learning and deep neural network research. It was built to run on multiple CPUs or GPUs and even mobile operating systems, and it has several wrappers in several languages like Python, C++ or Java.

8.7.1 Tensor Flow Architecture

Tensorflow architecture works in three parts:

- Preprocessing the data
- Build the model
- Train and estimate the model

It is called Tensorflow because it takes input as a multi-dimensional array, also known as tensors. You can construct a sort of flowchart of operations (called a Graph) that you want to perform on that input. The input goes in at one end, and then it flows through this system of multiple operations and comes out the other end as output. This is why it is called TensorFlow because the tensor goes in it flows through a list of operations, and then it comes out the other side.

8.7.2 Where Can TensorFlow Run?

TensorFlow hardware, and software requirements can be classified into

Development Phase:

This is when you train the mode. Training is usually done on your Desktop or laptop.

Run Phase or Inference Phase:

Once training is done TensorFlow can be run on many different platforms. We can run it on

- Desktop running Windows, macOS or Linux
- Cloud as a web service
- Mobile devices like iOS and Android

You can train it on multiple machines then you can run it on a different machine, once you have the trained model. The model can be trained and used on GPUs as well as CPUs. GPUs were initially designed for video games. In late 2010, Stanford researchers found that GPU was also very good at matrix operations and algebra so that it makes them very fast for doing these kinds of calculations. Deep learning relies on a lot of matrix multiplication. TensorFlow is very fast at computing the matrix multiplication because it is written in C++. Although it is implemented in C++, TensorFlow can be accessed and controlled by other languages mainly, Python. Finally, a significant feature of TensorFlow is the TensorBoard. The TensorBoard enables to monitor graphically and visually what TensorFlow is doing.

8.7.3 Introduction to Components of Tensorflow

1. **Tensor**

   Tensorflow's name is directly derived from its core framework: **Tensor**. In Tensorflow, all the computations involve tensors. A tensor is a vector or matrix of n-dimensions that represents all types of data. All values in a tensor hold identical data type with a known (or partially known) shape**.** The shape of the data is the dimensionality of the matrix or array. A tensor can be originated from the input data or the result of a computation. In TensorFlow, all the operations are conducted inside a graph. The graph is a set of computation that takes place successively. Each operation is called an op node and are connected to each other. The graph outlines the ops and connections between the nodes. However, it does not display the values. The edge of the nodes is the tensor, i.e., a way to populate the operation with data.

2. **Graphs**

TensorFlow makes use of a graph framework. The graph gathers and describes all the series computations done during the training. The graph has lots of advantages:

- It was done to run on multiple CPUs or GPUs and even mobile operating system
- The portability of the graph allows to preserve the computations for immediate or later use. The graph can be saved to be executed in the future.
- All the computations in the graph are done by connecting tensors together
- A tensor has a node and an edge. The node carries the mathematical operation and produces an endpoints outputs. The edges the edges explain the input/output relationships between nodes.

8.7.4 Why is Tensor Flow Popular?

TensorFlow is the best library of all because it is built to be accessible for everyone. Tensorflow library incorporates different API to built at scale deep learning architecture like CNN or RNN. TensorFlow is based on graph computation; it allows the developer to visualize the construction of the neural network with Tensorboad. This tool is helpful to debug the program. Finally, Tensorflow is built to be deployed at scale. It runs on CPU and GPU. Tensorflow attracts the largest popularity on GitHub compare to the other deep learning framework.

8.7.5 List of Prominent Algorithms supported by TensorFlow

Currently, TensorFlow 1.10 has a built-in API for:

- Linear regression: tf.estimator.LinearRegressor
- Classification:tf.estimator.LinearClassifier
- Deep learning classification: tf.estimator.DNNClassifier
- Deep learning wipe and deep: tf.estimator.DNNLinearCombinedClassifier
- Booster tree regression: tf.estimator.BoostedTreesRegressor
- Boosted tree classification: tf.estimator.BoostedTreesClassifier

Simple TensorFlow Example

```
import numpy as np
import tensorflow as tf
```

CHAPTER 9

MODULES

9.1 FACE RECOGNITION

Identifying a person with an image has been popularised through the mass media. However, it is less robust to fingerprint or retina scanning. This report describes the face detection and recognition mini-project undertaken for the visual perception and autonomy module at Plymouth university. It reports the technologies available in the Open-ComputerVision (OpenCV) library and methodology to implement them using Python.

For face detection, Haar-Cascades were used and for face recognition Eigenfaces, Fisherfaces and Local binary pattern histograms were used. The methodology is described including flow charts for each stage of the system. Next, the results are shown including plots and screen-shots followed by a discussion of encountered challenges.

9.2 FACE RECOGNITION ALGORITHM

Step 1: Import numpy, matplotlib, open-cv and glob.

Step 2: Using a glob loop through each of the .jpg files present in your current working directory and store them in a list ‘txtfiles’.

Step 3: Read each of the .jpg files using cv2.imread().

Step 4: Convert the image to gray image as OpenCV face detector expects gray images.

Step 5: Now, we have to load our Haar classifiers(downloaded XML files) for face detection and eye detection, which takes as input the training file of the Haar classifier.

Step 6: Now, how do we detect a face from an image using the CascadeClassifier ? ¬ EX - detectMultiScale(image, scaleFactor, minNeighbors)

Step 7: Now print the number of faces from each image (this is basically count the population of people in disaster situation).

Step 8: Now basically we’re finding faces, breaking the faces, their sizes, and drawing rectangles and stored into database.

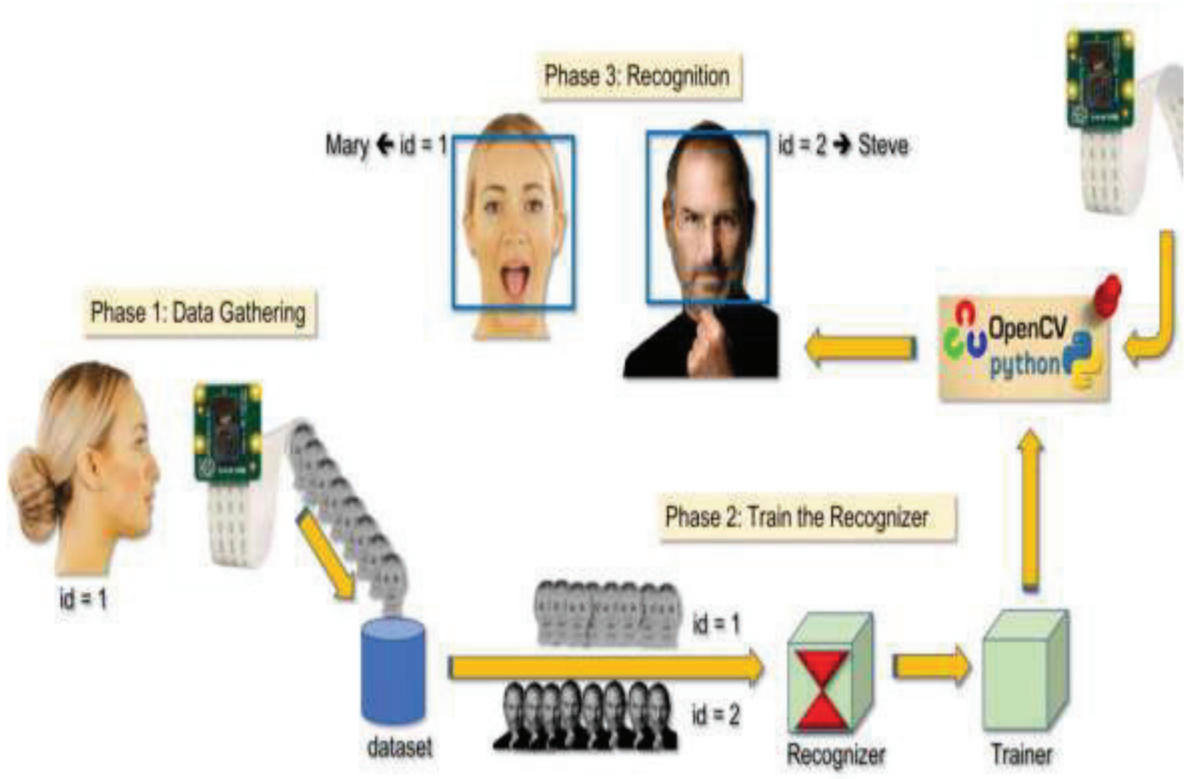


Figure 9.1: Face Recognition

9.3 EVOLUTION OF OBJECT DETECTION AND LOCALIZATION ALGORITHMS

Understanding recent evolution of object detection and localization with intuitive explanation of underlying concepts. Object detection is one of the areas of computer vision that is maturing very rapidly. Thanks to deep learning! Every year, new algorithms/ models keep on outperforming the previous ones. In-fact, one of the latest state of the art software system for object detection was just released last week by Facebook AI team. The software is called Detectron that incorporates numerous research projects for object detection and is powered by the Caffe2 deep learning framework.

9.3.1 Brief Introduction About CNN

Before I explain the working of object detection algorithms, I want to spend a few lines on Convolutional Neural Networks, also called CNN or ConvNets. CNNs are the basic building blocks for most of the computer vision tasks in deep learning era.

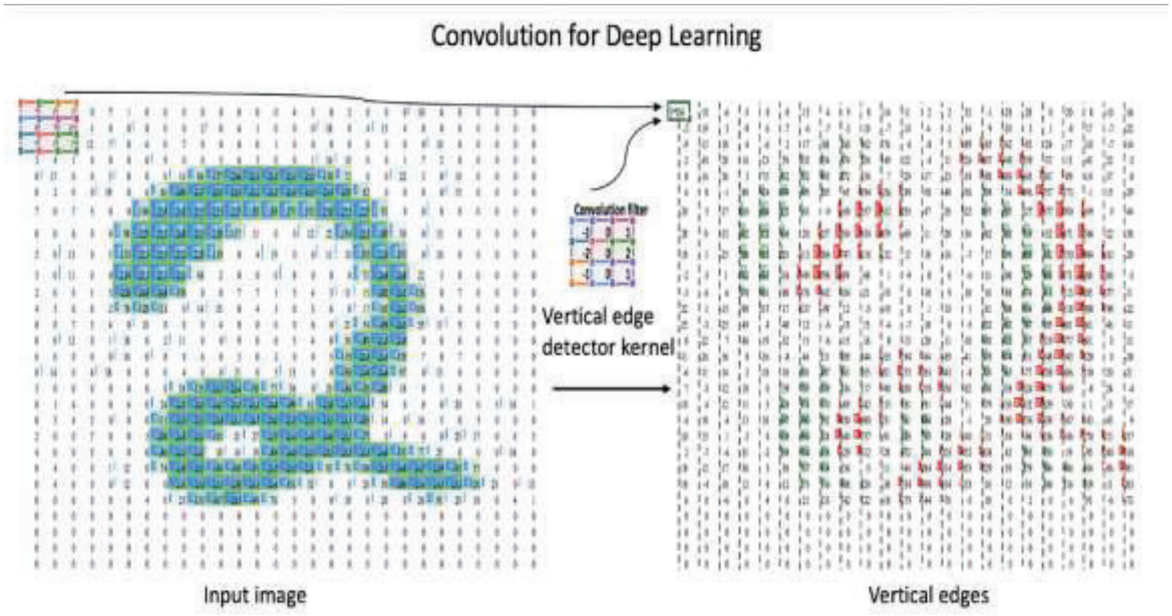


Figure 9.2: Convolution Demo in Excel

What we want?

We want some algorithm that looks at an image, sees the pattern in the image and tells what type of object is there in the image. For e.g., is that image of Cat or a Dog.

What is image for a computer?

Just matrix of numbers. For e.g. see the figure 1 above. The image on left is just a 28*28 pixels image of handwritten digit 2 (taken from MNIST data), which is represented as matrix of numbers in Excel spreadsheet.

How can we teach computers learn to recognize the object in image?

By making computers learn the patterns like vertical edges, horizontal edges, round shapes and maybe plenty of other patterns unknown to humans.

How computers learn patterns?

Convolutions!

Convolution is a mathematical operation between two matrices to give a third matrix. The smaller matrix, which we call filter or kernel (3x3 in figure 1) is operated on the matrix of image pixels. Depending on the numbers in the filter matrix, the output matrix can recognize the specific patterns present in the input image. In example above, the filter is vertical edge detector which learns vertical edges in the input image. In context of deep learning, the input images and their subsequent outputs are passed from a number of such filters. The numbers in filters are learnt by neural net and patterns are derived on its own[19].

Why convolutions work?

Because in most of the images, the objects have consistency in relative pixel densities (magnitude of numbers) that can be leveraged by convolutions.

I know that only a few lines on CNN is not enough for a reader who doesn't know about CNN. But CNN is not the main topic of this blog and I have provided the basic intro, so that the reader may not have to open 10 more links to first understand CNN before continuing further.

9.3.2 Categorization of Computer Vision Tasks

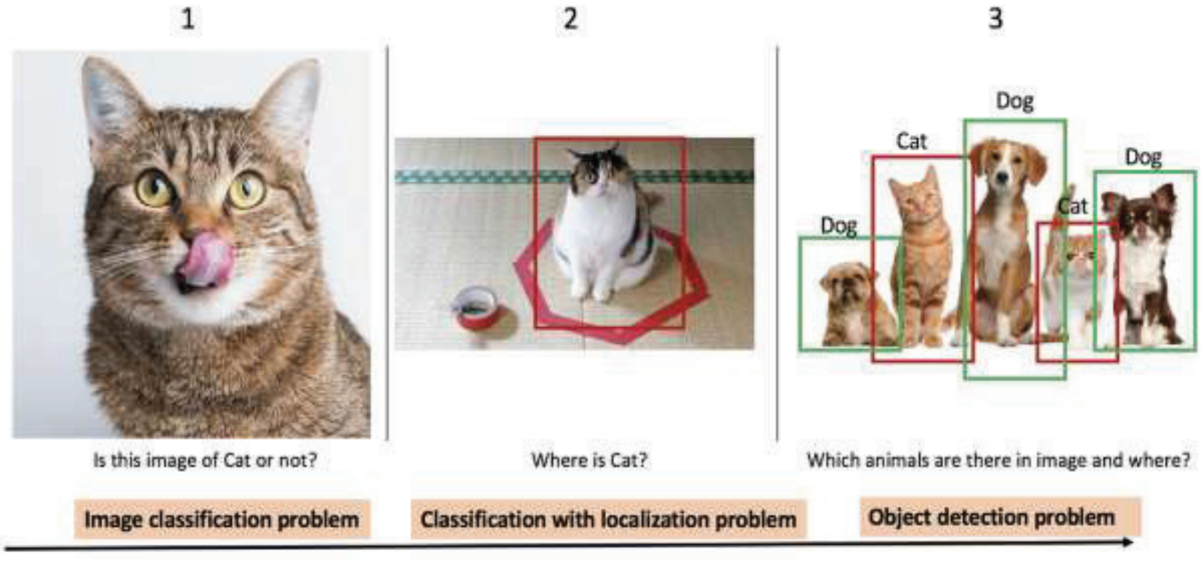

1
2
3
Dog
Cat
Dog
Cat
Dog
Dog
Is this image of Cat or not?
Where is Cat?
Which animals are there in image and where?
Image classification problem
Classification with localization problem
Object detection problem

Figure 9.3: Common Computer Vision Tasks

Taking an example of cat and dog images in Figure 2, following are the most common tasks done by computer vision modeling algorithms:

1. **Image Classification:**
   This is the most common computer vision problem where an algorithm looks at an image and classifies the object in it. Image classification has a wide variety of

applications, ranging from face detection on social networks to cancer detection in medicine. Such problems are typically modeled using Convolutional Neural Nets (CNNs).

2. **Object classification and localization:**

   Let's say we not only want to know whether there is cat in the image, but where exactly is the cat. Object localization algorithms not only label the class of an object, but also draw a bounding box around position of object in the image.

3. **Multiple objects detection and localization:**

   What if there are multiple objects in the image (3 dogs and 2 cats as in above figure) and we want to detect them all? That would be an object detection and localization problem. A well known application of this is in self-driving cars where the algorithm not only needs to detect the cars, but also pedestrians, motorcycles, trees and other objects in the frame. These kind of problems need to leverage the ideas or concepts learnt from image classification as well as from object localization.

Now coming back to computer vision tasks. In context of deep learning, the basic algorithmic difference among the above 3 types of tasks is just choosing relevant input and outputs. Let me explain this line in detail with an infographic.

9.3.3 Image Classification

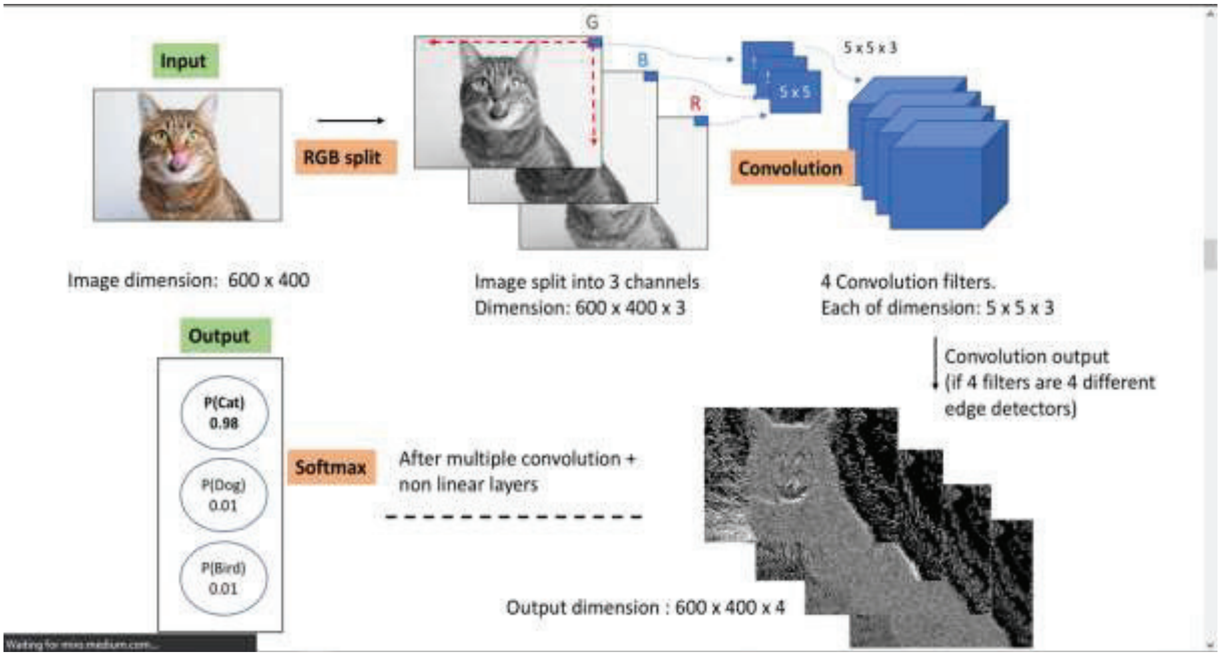

Input
RGB split
G
B
R
5 x 5 x 3
5 x 5
Convolution
Image dimension: 600 x 400
Image split into 3 channels
Dimension: 600 x 400 x 3
4 Convolution filters.
Each of dimension: 5 x 5 x 3
Output
Convolution output
(if 4 filters are 4 different
edge detectors)
P(Cat)
0.98
P(Dog)
0.01
P(Bird)
0.01
Softmax
After multiple convolution +
non linear layers
Output dimension : 600 x 400 x 4
Waiting for miro.medium.com...

Figure 9.4: Steps for Image Classification Using CNN

The infographic in Figure 3 shows how a typical CNN for image classification looks like.

1. Convolve an input image of some height, width and channel depth

2. The output of convolution is treated with non-linear transformations, typically Max Pool and RELU.

3. The above 3 operations of Convolution, Max Pool and RELU are performed multiple times.

4. The output of final layer is sent to Soft max layer which converts the numbers between 0 and 1, giving probability of image being of particular class. We minimize our loss so as to make the predictions from this last layer as close to actual values.

9.3.4 Object Classification and Localization

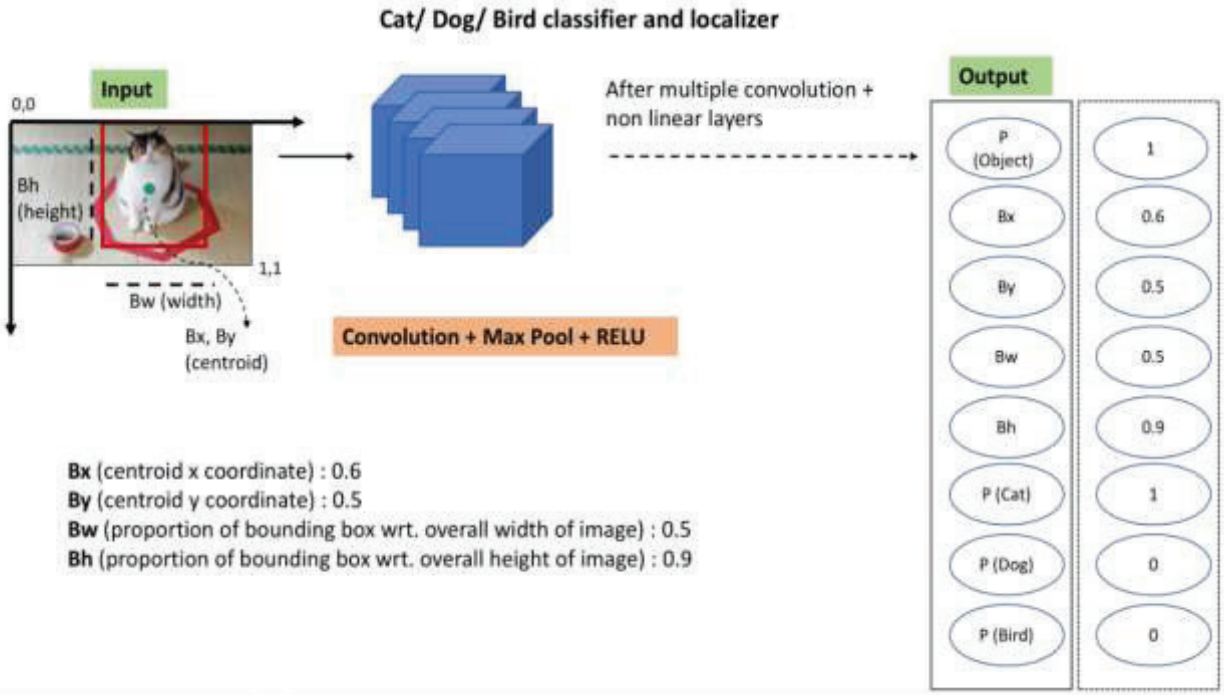

Cat/ Dog/ Bird classifier and localizer
Input
0,0
Bh (height)
1,1
Bw (width)
Bx, By (centroid)
Convolution + Max Pool + RELU
After multiple convolution + non linear layers
Output
P (Object)
Bx
By
Bw
Bh
P (Cat)
P (Dog)
P (Bird)
1
0.6
0.5
0.5
0.9
1
0
0
Bx (centroid x coordinate) : 0.6
By (centroid y coordinate) : 0.5
Bw (proportion of bounding box wrt. overall width of image) : 0.5
Bh (proportion of bounding box wrt. overall height of image) : 0.9

Figure 9.5: Input and output for object localization problem

Now, to make our model draw the bounding boxes of an object, we just change the output labels from the previous algorithm, so as to make our model learn the class of object and also the position of the object in the image. We add 4 more numbers in the output layer which include centroid position of the object and proportion of width and height of bounding box in the image.

Simple, right? Just add a bunch of output units to spit out the x, y coordinates of different positions you want to recognize. These different positions or landmark would be consistent for a particular object in all the images we have. For e.g. for a car, height would be smaller than width and centroid would have some specific pixel density as compared to other points in the image.

Implying the same logic, what do you think would change if we there are multiple objects in the image and we want to classify and localize all of them? I would suggest you to pause and ponder at this moment and you might get the answer yourself.

9.3.5: Multiple Objects Detection and Localization

To detect all kinds of objects in an image, we can directly use what we learnt so far from object localization. The difference is that we want our algorithm to be able to classify and localize all the objects in an image, not just one. So the idea is, just crop the image into multiple images and run CNN for all the cropped images to detect an object.
The way algorithm works is the following:

1. Make a window of size much smaller than actual image size. Crop it and pass it to ConvNet (CNN) and have ConvNet make the predictions.

2. Keep on sliding the window and pass the cropped images into ConvNet.

3. After cropping all the portions of image with this window size, repeat all the steps again for a bit bigger window size. Again pass cropped images into ConvNet and let it make predictions.

4. At the end, you will have a set of cropped regions which will have some object, together with class and bounding box of the object.

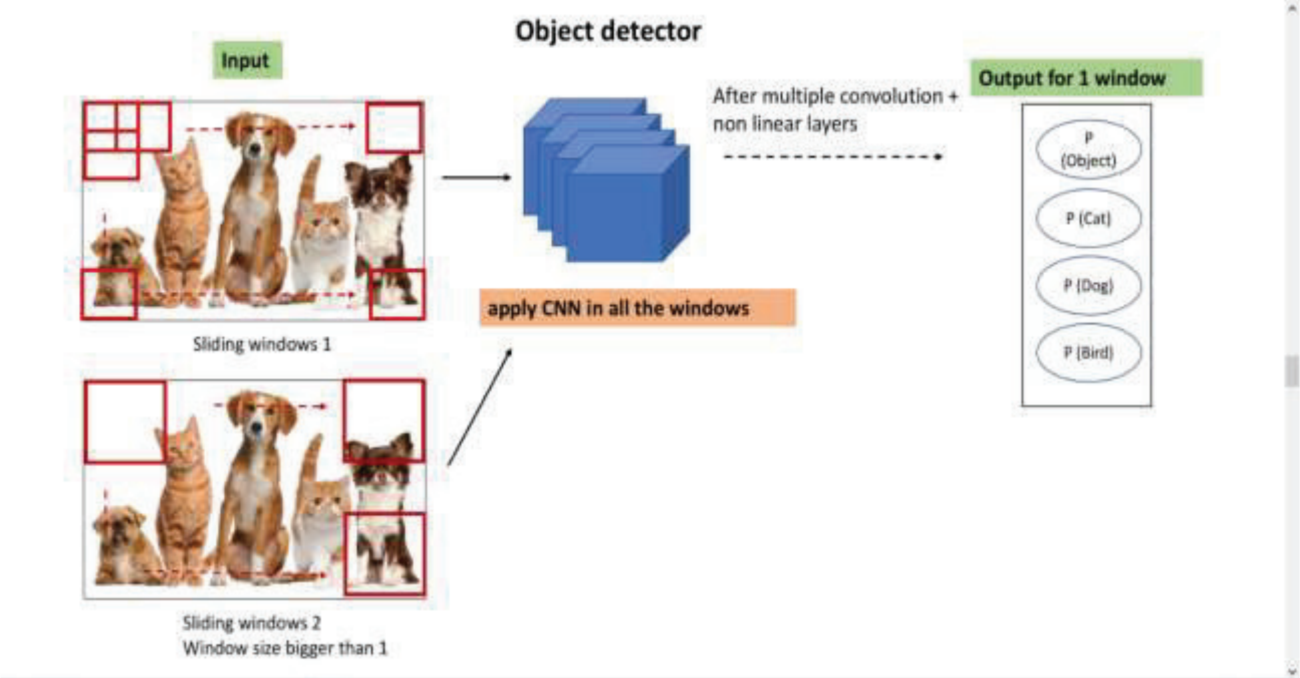


Figure 9.6: Input and output for object detection and localization problems

This solution is known as object detection with sliding windows. It is very basic solution which has many caveats as the following:

A. Computationally expensive: Cropping multiple images and passing it through ConvNet is going to be computationally very expensive.

Solution: There is a simple hack to improve the computation power of sliding window method. It is to replace the fully connected layer in ConvNet with 1x1 convolution layers and for a given window size, pass the input image only once. So, in actual implementation we do not pass the cropped images one at a time, but we pass the complete image at once.

B. Inaccurate bounding boxes: We are sliding windows of square shape all over the image, maybe the object is rectangular or maybe none of the squares match perfectly with the actual size of the object. Although this algorithm has ability to find and localize multiple objects in an image, but the accuracy of bounding box is still bad.

9.3.6: Better Solution? YOLO

It turns out that we have YOLO (You Only Look Once) which is much more accurate and faster than the sliding window algorithm. It is based on only a minor tweak on the top of algorithms that we already know. The idea is to divide the image into multiple grids. Then we change the label of our data such that we implement both localization and classification algorithm for each grid cell.

YOLO IN EASY STEPS:

1. Divide the image into multiple grids. For illustration, I have drawn 4x4 grids in above figure, but actual implementation of YOLO has different number of grids. (7x7 for training YOLO on PASCAL VOC dataset)

2. If C is number of unique objects in our data, S*S is number of grids into which we split our image, then our output vector will be of length S*S*(C+5). For e.g. in above case, our target vector is 4*4*(3+5) as we divided our images into 4*4 grids and are training for 3 unique objects: Car, Light and Pedestrian.

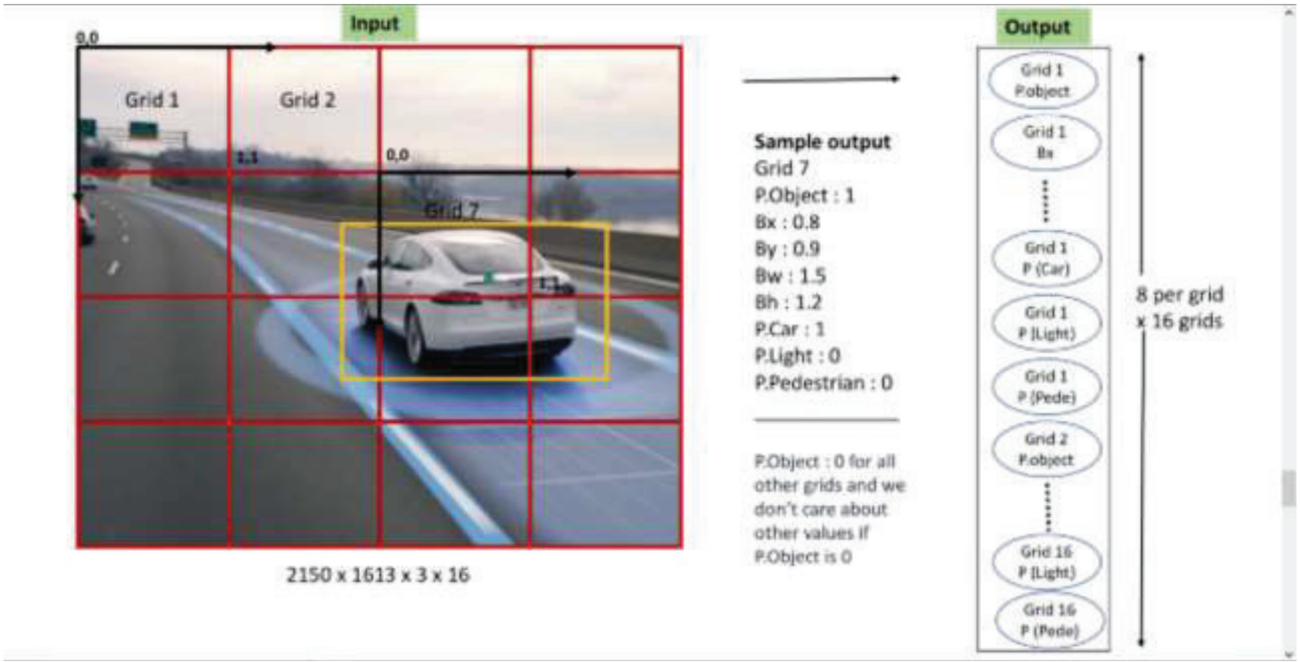


Figure 9.7: Bounding boxes, input and output for YOLO

3. Make one deep convolutional neural net with loss function as error between output activations and label vector. Basically, the model predicts the output of all the grids in just one forward pass of input image through ConvNet.

4. Keep in mind that the label for object being present in a grid cell (P.Object) is determined by the presence of object's centroid in that grid. This is important to not allow one object to be counted multiple times in different grids[19].

Caveats of YOLO and their solutions:

A. **Can't detect multiple objects in same grid.**

This issue can be solved by choosing smaller grid size. But even by choosing smaller grid size, the algorithm can still fail in cases where objects are very close to each other, like image of flock of birds.

Solution: Anchor boxes.

In addition to having 5+C labels for each grid cell (where C is number of distinct objects), the idea of anchor boxes is to have (5+C)*A labels for each grid cell, where A is required anchor boxes. If one object is assigned to one anchor box in one grid, other object can be assigned to the other anchor box of same grid.

B. **Possibility to detect one object multiple times.**

Solution: Non-max suppression. Non max suppression removes the low probability bounding boxes which are very close to a high probability bounding boxes.

9.3.7 Result

As of today, there are multiple versions of pre-trained YOLO models available in different deep learning frameworks, including Tensorflow. The latest YOLO paper is: “YOLO9000: Better, Faster, Stronger” . The model is trained on 9000 classes. There are also a number of Regional CNN (R-CNN) algorithms based on selective regional proposal. Detectron, software system developed by Facebook AI also implements a variant of R-CNN, Masked R-CNN.

9.4 PREDICT AGE AND GENDER USING CONVOLUTIONAL NEURAL NETWORK AND OPENCV

9.4.1 Introduction

Age and gender, two of the key facial attributes, play a very foundational role in social interactions, making age and gender estimation from a single face image an important task in intelligent applications, such as access control, human-computer interaction, law enforcement, marketing intelligence and visual surveillance, etc.

9.4.2 Requirements:

1. pip install OpenCV-python.
2. numpy.
3. pip install pafy.

Pafy

Pafy library is used to retrieve YouTube content and metadata (such as Title, rating, view count, duration, rating, author, thumbnail, keywords etc).

STEPS TO FOLLOW:

1. Face detection with Haar cascades
2. Gender Recognition with CNN
3. Age Recognition with CNN

1. Face detection with Haar cascades:

This is a part most of us at least have heard of. OpenCV/JavaCV provide direct methods to import Haar-cascades and use them to detect faces. I will not be explaining this part in deep.

2. Gender Recognition with CNN:

Gender recognition using OpenCV's fisherfaces implementation is quite popular and some of you may have tried or read about it also. But, in this example, I will be using a different approach to recognize gender. This method was introduced by two Israel researchers, Gil Levi and Tal Hassner in 2015. I have used the CNN models trained by them in this example. We are going to use the OpenCV's dnn package which stands for "Deep Neural Networks". In the dnn package, OpenCV has provided a class called Net which can be used to populate a neural network. Furthermore, these packages support importing neural network models from well known deep learning frameworks like caffe, tensorflow and torch. The researchers I had mentioned above have published their CNN models as caffe models. Therefore, we will be using the CaffeImporter import that model into our application[19].

3. Age Recognition With Cnn

This is almost similar to the gender detection part except that the corresponding prototxt file and the caffe model file are “deploy_agenet.prototxt” and “age_net.caffemodel”. Furthermore, the CNN’s output layer (probability layer) in this CNN consists of 8 values for 8 age classes (“0–2”, “4–6”, “8–13”, “15–20”, “25–32”, “38–43”, “48–53” and “60-”) A caffe model has 2 associated files [20].

- **prototxt** — The definition of CNN goes in here. This file defines the layers in the neural network, each layer’s inputs, outputs and functionality.

- **.caffemodel** — This contains the information of the trained neural network (trained model).

9.4.3 Result

As we have seen in this article that in just a few lines of code we have built an age and gender detection model, from here on you can also incorporate emotion detection and object detection in the same model and create a fully functional application.

CHAPTER 10

EXPERIMENTAL RESULTS AND DISCUSSION

In this section of the document we will be discussing the verification and testing of each hardware and software component. All problems will be described in detail and the solutions we made to solve these problems. In this section we will also discuss our overall results of the project and what we could have done to improve upon our project. Future work for this project will also be mentioned in this section of the document.

10.1 SYSTEM VERIFICATION AND TESTING

In this section of the document we will be discussing the methods we used to test each component of our quadcopter, the problems we faced, and how we solved them.

10.1.1 Testing Procedure

1.Verifying sensors board and Arduino Mega connection

By using the blinking template from the Arduino IDE and looking at the corresponding LED on the Arduino board and shield, we verified that both the boards were connected properly by changing the delay of the blinking LED.

2.Testing APM

To modify the user configuration header file had to be changed. Certain variables needed to be defined according what components our quadcopter had and what functions we wanted our quadcopter to perform.

3.Testing of sensors separately

Most of the testing programs for the sensors that came with the inbuilt software did not work at all. There were some errors in each program that needed to be solved before any testing could be done.

- The 3-axis gyroscope test program was not working properly. The program was missing a variable and the address to the gyroscope was addressed incorrectly. This was solved after adding the missing variable and changing the address to the correct location of where the gyroscope sensor was located.

- The 3-axis magnetometer test program was not working properly. The program was missing header files. This was solved after adding the necessary header files.

- The 3-axis accelerometer test program was also not working properly. No matter how the sensor was moved, the readings did not change. The problem with this program was that the scaling factors being used were set to zero. After setting the scaling factors to the correct numbers, the program worked properly and accelerometer measurements could be made.

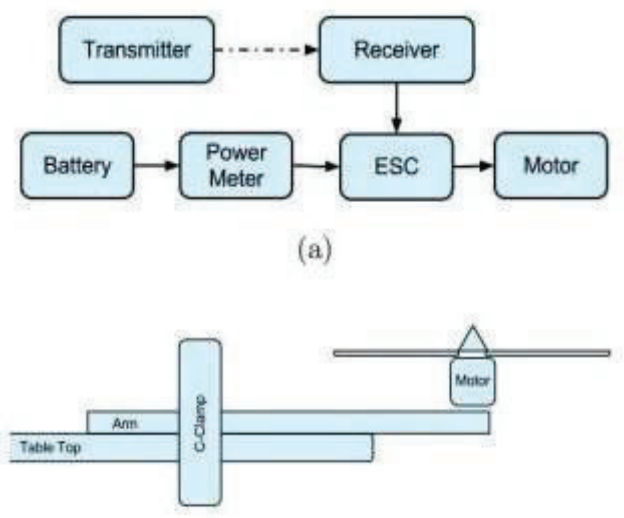
- The barometeric pressure sensor program worked without any problems.

4. Testing quadcopter orientation with sensors board

After testing and verifying that all the sensors worked. We used the user configurator to graph the outputs of our sensor measurements. The configurator uses a kinematics algorithm to calculate pitch, roll, and yaw of a quadcopter by using the measurements from the gyroscope, accelerometer, and magnetometer. The kinematics algorithm will be explained in further detail in the next section of the report. It is worth mentioning again that the result of pitch, roll, and yaw is a combined result of the gyroscope, accelerometer, and magnetometer readings calculated by the kinematics algorithm.

10.1.2 Motors and Electronic Speed Controller

Testing of the motors and ESC was done to verify that each component was working properly. The ESC needed some assembly, barrel connectors were soldered onto the ends of the ESC. A motor was mounted to an arm of the quadcopter and securely attached the arm to a table edge using a C-clamp. The motor was then wired up according to the block diagram of Figure 10.1



(a)

(b)

Figure 10.1: Motor & ESC test bench: (a) Block diagram. (b) Test bench.

Before starting the test, each ESC was programmed to the same settings and throttle range. For verification, the voltage, current, and power were measured using a power meter. The measurements were made at throttle speeds at 10% increments. This was done three to five times for each motor and ESC.

Unfortunately, one of the ESC could not be programmed properly. This seemed to be solved after setting the proper throttle range for the ESC. But during testing, the measurements for the ESC were much lower than the readings when compared to the other ESC. All the other ESC/motor measurements were very similar, but the measurements for this ESC started dropping off at about 50% throttle and the power to the motor from the ESC was cut at about 95% - 100% throttle.

As a result, more ESCs were purchased to replace the malfunctioning ESC. After receiving the new ESC, we soldered the necessary connectors and re-tested the ESC using the same setup shown in the block diagram above. However, during one of these test, the connectors of a ESC were not heat shrunk properly. The connectors were touching and caused a short circuit when the battery was connected and warped the wire casing. Fortunately, this did not affect the ESC and the ESC is still working properly.

10.1.3 Problems

The damaged arms needed to be replaced. The bent arms would affect the stabilization of the quadcopter because the motors would not be aligned properly. Luckily, we found some replacement arms at the local hardware store. A hollow U-bar was purchased and cut into the necessary lengths at the mechanic shop. The metal bar was about 2 mm too thick. One quick solution was to buy longer bolts to hold the arms and the frame together. We are currently still in the process of fixing our quadcopter.

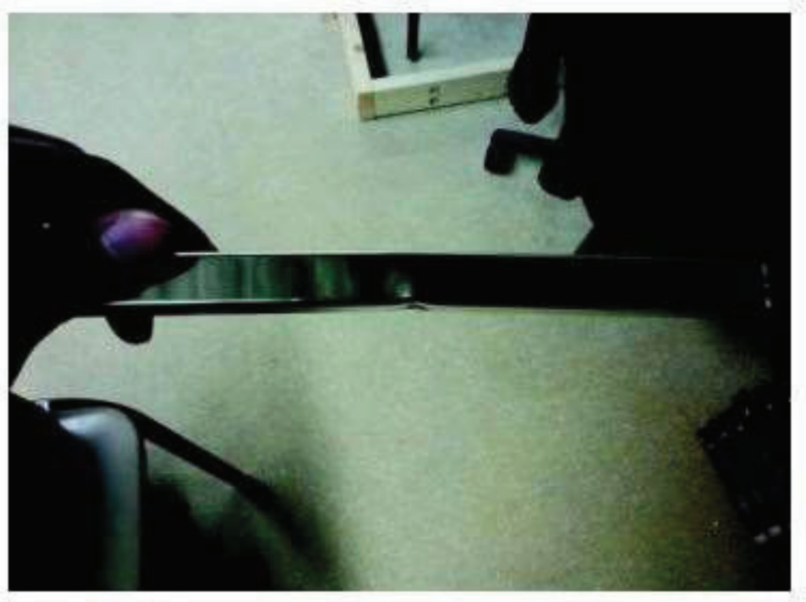


Figure 10.2: Damaged Arms.

10.2 DISCUSSION

10.2.1 Difficulties

1. We encountered delays in ordering parts. We ordered parts in late September and early October and parts did not arrive at the expected times. Our initial parts started arriving early November but some parts did not arrive until the last week of November. This delayed our schedule and caused us to fall behind.

2. There was a problem with the battery connectors being different. We did not notice the difference in connectors at the time of purchase. The JST-HX connectors on the battery did not match the EC3 connectors on the quadcopter. Fortunately, we were able to find the necessary connectors at a local shop and we were able to re-solder the new connectors to the battery.

3. Setting up the channels of the transmitter and receiver was also very important. The initial transmitter channel configurations did not match the quadcopter channel configurations. To solve this problem each channel of the transmitter needed to be tested separately to identify what it was controlling on the quadcopter and set to the correct corresponding channels.

4. We need to use li-po battery which is very costly so to maintain the cost at our level we have used 2200 MAH battery which gives only 15-20 min fly time which is very less so this is the main difficulty we face in our project.

10.2.2 Future Work

Our overall plans were too ambitious. Our team goals were to design, test, and build a quadcopter kit, interface a data logging system, and design and implement AI commands for our quadcopter. Unfortunately, we were not able to complete all these goals. In the future, if time permits, we would like to implement these commands. There were many problems that we did not account for throughout the project. These problems delayed our progress and we were not able to finish what we set out to do in the beginning. We have learned that we must allow time for solving problems in the future.

Other plans in the future include adding a sonic sensor for more accurate altitude determination. Currently the only methods to determine altitude is by using the barometric pressure sensor and the GPS receiver. There is no actual way to safely determine the quadcopters altitude relative to its landing surface. A sonic sensor could solve this problem, and be used to help aid the auto-landing command.

Another future plan would be adding more methods of collecting data. Many ports still remain unused on the control board. Adding a camera could allow for digital photos or video to be taken. Adding some way to stream data from the quadcopter to the controller could be another great feature to add to our quadcopter, this would allow for even easier access to the data collected by the quadcopter. Smart phone capabilities could be another feature our group may want to add in the future.

10.3 SUMMARY

Each component was tested was verified to be working as intended. Test flights have been conducted and the results confirm that the quadcopter can fly in a stable manner.

Conclusion

Under the supervision of Mr. Zubair Iqbal at Moradabad Institute of Technology, a quadcopter was built and tested. The parameters were adjusted and stabilization of the quadcopter was achieved. The quadcopter can maintain stable flight without any observable oscillations. Currently, the quadcopter weighs roughly 1685 grams, maintains an average flight duration of 12 minutes, and can be controlled within a range of 300 meters. The calamity is a sudden accident or a natural catastrophe that causes great damage or loss of life.

Thus, VISION provides applicable information and assistance, therefore we need to collect calamitous information and build accordingly. They can provide an instant telecommunication infrastructure, assist in telemedicine-enabled clinical services, perform equipment / drug / patient delivery, enhance search and rescue efforts, assess damage and map calamity zones.

The small scale VISION is cost effective and has many applications. This is a brief look at the emerging technologies merely scratching the surface of employing VISION in a strike role. This project requires members not only to interface and program the components of the quadcopter, but also to expose them to mechanical components and the reality of the project is the management to accomplish the project objectives.

Bibliography

[1] "Map of winnipeg, manitoba," February 2013. [Online]. Available: https://maps.google.ca/

[2] R. Mahony, P. Pounds, and P. Corke. (2006, December) Modelling and control of a quad-rotor robot. In the Proceedings of the Australasian Conference on Robotics and Automation. Auckland, New Zealand. Accessed: September 2012. [Online]. Available: http://www.araa.asn.au/acra/acra2006/papers/paper

[3] I. Sa and P. Corke. Vertical infrastructure inspection using a quadcopter and shared autnomy control. Brisbane, Austrailia. Accessed: September 2012. [Online]. Available: https://wiki.qut.edu.au/download/attachments/107284164/FSR2012Inkyu5thsub.pdf?vers%20ion=1&modificationDate=1336532095000

[4] V. Ross. (2011, August) In development: Networks of unmanned quadcopters to ferry medicine to isolated areas. Discover Magazine. Accessed: September 2012. [Online]. Available: http://blogs.discovermagazine.com/80beats/2011/08/30/in-development-networks-of-% 20unmanned-quadcopters-to-ferry-medicine-to-isolated-areas/#.UTqfsxxgCXg

[5] T. Carancho. aeroquad software. Accessed: September 2012. [Online]. Available: https://code.google.com/p/aeroquad/

[6] G. Elliot. (2005, November) Development of an autonomous quadrotor flying platform. Accessed: September 2012. [Online]. Available: http://blogs.discovermagazine.com/80beats/2011/08/30/in-development-networks-of-% 20unmanned-quadcopters-to-ferry-medicine-to-isolated-areas/#.UTqfsxxgCXg

[7] G. M. Hoffman, H. Huang, S. L. Waslander, and C. J. Tomlin. Quadrotor helicopter flight dynamics and control: Theory and experiment. Accessed: October 2012. [Online]. Available: http://hoffmann.stanford.edu/papers/Quadrotor Dynamics GNC07.pdf

[8] S. W. Colton. (2010, June) Design and prototyping methods for brushless motors and motor control. Massachusetts, United States. Accessed: October 2012. [Online]. Available: http://web.mit.edu/scolton/www/SCThG.pdf

[9] R. F. Graf, "Dictionary of electronics," 1974., accessed: January 2013. [Online]. Available: http://www.rc-airplane-world.com/spread-spectrum.html

[10] "Canadian municipalities and the regulation of radio antennae and their support structures," accessed: January 2013. [Online]. Available: http://www.ic.gc.ca/eic/site/smt-gst.nsf/eng/ sf09386.html

[11] "72 MHz vs. 2.4 GHz radio systems compared," February 2013, accessed: January 2013. [Online. Available: http://sjrcf.com/docs/72mhz%20vs%202.4%20ghz.pdf

[12] P. Energy, "Battery types: Which batteries to use?" accessed: October 2012. [Online]. Available: http://www.pureenergybattery.com/pdf/batterytypes.pdf

[13] “Peukert’s law, a nerd’s attempt to explain battery capacity,” accessed: September 2012. [Online]. Available: http://www.batterystuff.com/kb/tools/ peukert-s-law-a-nerds-attempt-to-explain-battery-capacity.html

[14] https://www.dronezon.com/learn-aboutdronesquadcopters/what-is-drone-technology-or-howdoes-dronetechnology-work/

[15] https://towardsdatascience.com/introduction-to-machine-learning-for-beginners-eed6024fdb08

[16] https://towardsdatascience.com/beginners-guide-to-data-science-libraries-in-python-4b1a9326b712

[17] https://en.wikipedia.org/wiki/Quadcopter

[18] https://www.investopedia.com/terms/d/deep-learning.asp

[19] https://machinelearningmastery.com/what-is-deep-learning/

[20] Starling, “Dom tutorial - an introduction to orientation kinematics,” accessed: January 2013. [Online]. Available: http://www.starlino.com/dcm tutorial.html

IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

VISION – A Drone For Emergency Operations

Abhishek Kumar[1], Akshita Sharma[2], Ankit Agarwal[3], Ashish[4], Zubair Iqbal[5]

[1,2,3,4]U.G. Scholar, [5]Assistant Professor ,Computer Science and Engineering, M.I.T, Moradabad, U.P.

***Abstract*- In the current Era, there are such a large number of advancements in the field of automatons. Additionally, the military and business uses of automatons, there is no uncertainty in their productivity in the event of supporting crisis calamity activities. This paper assesses a few encounters and portrays a few activities utilizing automatons to help crisis debacle tasks. It concentrates for the most part in crisis tasks, similar to action following the event of a debacle like atomic mishaps, risky material discharges, floods, seismic tremors. For unique salvage groups, the automaton can help much in a fast area determination, identifying constant circumstance, tallying number of individuals or guys and females. Floods are commonplace for a moderate beginning debacle. Interestingly, overseeing floods is a mind boggling and troublesome errand. It requires consistent observing of dykes, overflowed and compromised zones. Automaton can help chiefs to a great extent holding a region under perception. Automaton can be utilized for fire location, intercession observing and furthermore for post-fire checking. In the event of atomic mishap or unsafe material spillage drone is likewise a compelling or can be the just one device for supporting crisis tasks.**

***Keywords*— Drone, GPS, Raspberry Pi.**

I. INTRODUCTION

A drone is an aircraft without a human pilot on board. Its flight is controlled either autonomously by computers in the vehicle or under the remote control of a pilot on the ground or in another vehicle.

We are going to create a Drone which is a flying robot with an embedded system and work in the conjunction with on board sensors and GPS. It enables Object Detection & Face Recognition. We named it as VISION-A Drone for Emergency Operations. It is used to detect and analysis the disaster and count number of people, their gender and prepare the dataset, also detect the real time situation of the area.

1.2 million Deaths, 2.9 billion People are affected. $1.7 trillion in damages. According to data from the United Nations Office for Disaster Risk Reduction, these staggering figures are the total economic and human impact of global disasters from 2002 to 2012[1]. With a steady growth in annual disasters, especially climate-related ones, emergency management strategies are being put under the microscope. Disaster management technologies, on the other hand, we have seen some remarkable breakthroughs in the past decade.

A small scale VISION(Drone) can be designed using Flight controller, APM(2.8), Quadcopter frame, Brushless motors, Raspberry Pi , GPS, Ultrasonic sensors, Pi Cam, Electronic Speed Controller (ESC),IR Sensors, Transmitter & Receiver.

II. COMPONENTS

Brushless Motor:

It is a simultaneous engine that is controlled by a DC electric source by a coordinated exchanging power gracefully, which creates an AC electric sign to drive the engines. Electronic Speed Controller with Battery Eliminator Circuit: It is an electronic circuit with the goal to fluctuate an electric Motor's speed, its heading and perhaps at the same time to go about as a unique brake [2].

Lithium battery:

Lithium-particle polymer battery is a battery-powered battery. It for the most part comprises of a few indistinguishable optional cells in corresponding to expand the release current ability and is every now and again accessible in arrangement design to build the absolute accessible voltage [3].

Transmitter/ Receiver:

It plays out the activity of transmitting and getting correspondence signals used to control any framework. It has a scope of 1 kilometer sweep and accompanies 4 channels which are utilized to run various parts associated with the collector [4].

APM (2.8):

It is a finished open source autopilot framework. The APM 2.8 is a finished open source autopilot framework and the top of the line innovation that won the esteemed Outback Challenge UAV competition[5]. It permits the client to turn any fixed, rotating wing or multirotor vehicle (even vehicles and pontoons) into a completely independent vehicle; equipped for performing customized GPS missions with waypoints.

Raspberry Pi:

It resembles a working framework. The Raspberry Pi is a minimal effort, charge card estimated PC that connects to a PC screen or TV, and utilizations a standard console and mouse [6].

Quadcopter Frame:

It resembles a working framework. The Raspberry Pi is a minimal effort, charge card estimated PC that connects to a PC screen or TV, and utilizations a standard console and mouse.

Ultrasonic Sensor:

For identifying ground and furthermore utilized in Terrain Follow Mode. Ultrasonic transducers or ultrasonic sensors are a sort of acoustic sensor partitioned into three general classifications: transmitters, beneficiaries and transceivers. Transmitters convert electrical signs into ultrasound, recipients convert ultrasound into electrical signs, and handsets can both transmit and get ultrasound [8].

Power Module:

Utilized for power transformation hardware, for example, mechanical engine drives, implanted engine drives, air conditioning dc power supplies.

IR Sensor:

An Infrared Sensor is an electronic instrument that is utilized to detect certain attributes of its environmental factors. It is occurred by either producing or distinguishing Infrared Radiation.

III. PROPOSED SYSTEM

We are going to develop a drone with some AI features like object detection, face recognition and perform some special tasks like counting people and counting number of males and females with the help of ML and Al techniques which will help in some special or emergency operations so first we need to make a drone. To make a drone we need a frame, four brushless motors, ESC (electronic speed controller), propellers, flight controller, transmitter and receiver, power distribution board, GPS, and battery for power supply and assemble all these parts which will forms a drone which perform all basic drone tasks like take off, move forward, move backward, move left, move right and landing so this is a basic drone now we need to make it smart. With the help of Raspberry Pi and Pi cam we give it an Intelligence from python programming in raspberry pi. To give intelligence to the drone we use open CV for object detection and face recognition which take input from the Pi Cam and GPS and after analyses the data it send data to the system administrator which is a easy task rather than use a manual method. To understand the working of the Vision we took an example. For example there is a flood in the city and we need to rescue the stuck people in the city and we did not know that how many number of people in the city, so we send our vision drone to the location which have intelligence to do this work efficiently.

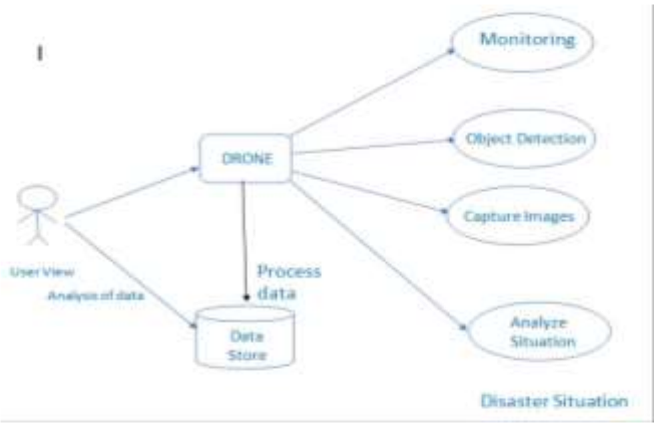


Fig.1 Use Case Diagram of Proposed System

IV. EXPERIMENT PROCEDURE

The essential goal of this paper is to assess how Drones or unmanned elevated vehicles – UAV's in the present or not so distant future can help survivors on account of a tidal wave, quake, flooding, and any cataclysmic event. At first, we expect that in any catastrophic event consistently exist the high chance of harm to the foundation, transportation frameworks, media communications frameworks access, and essential administrations right away. This exploration proposes three regions the employments of unmanned elevated vehicles – UAV's-(or Drones) on account of cataclysmic events reaction and helpful alleviation help.

Fig.2 Procedure of Vision-A Drone for Emergency Operations

Module 1. Face Recognition

A facial acknowledgment framework is an innovation fit for recognizing or confirming an individual from a computerized picture or a video outline from a video source. There are numerous techniques where facial acknowledgment frameworks work, yet all in all, they work by looking at chosen facial highlights from given picture with faces inside a database. It is additionally depicted as a biometric Artificial Intelligence based application that can exceptionally recognize an individual by breaking down examples dependent on the individual's facial surfaces and shape.

Algorithm Used For Face Recognition

Principal Component Analysis (PCA) for face acknowledgment depends on the data hypothesis approach. It removed the pertinent data in a face picture and encoded as effectively as could be expected under the circumstances. It recognizes the subspace of the picture space spread over by the preparation face picture information and stylistic layout relate the pixel esteems. The old style portrayal of a face picture is gotten by anticipating it to the arrange framework characterized by the foremost parts. The projection of face pictures into the vital part subspace accomplishes data pressure, decorrelation and dimensionality decrease to encourage dynamic. In numerical terms, the essential segments of the dispersion of appearances or the eigenvectors of the covariance framework of the arrangement of face pictures, is looked for by regarding a picture as a vector in a high dimensional face space. We apply PCA on the database and get the remarkable element vectors utilizing the accompanying technique .Suppose there are P designs and each patter has preparing pictures of m x n setup [9].

- **Step 1**: Import numpy, matplotlib, open-cv and glob
- **Step 2**: Using a glob loop through each of the .jpg files present in your current working directory and store them in a list 'txtfiles'.
- **Step 3**: Read each of the .jpg files using cv2.imread().
- **Step 4**: Convert the image to gray image as OpenCV face detector expects gray images.
- **Step 5**: Now, we have to load our Haar classifiers downloaded XML files) for face detection and eye detection, which takes as input the training file of the Haar classifier.
- **Step 6**: Now, how do we detect a face from an image using the Cascade Classifier ?
- EX-detectMultiScale(image,scaleFactor,minNeighbors)
- **Step 7**: Now print the number of faces from each image (this is basically count the population of people in disaster situation).
- **Step 8**: Now basically we're finding faces, breaking the faces, their sizes, and drawing rectangles and stored into database.

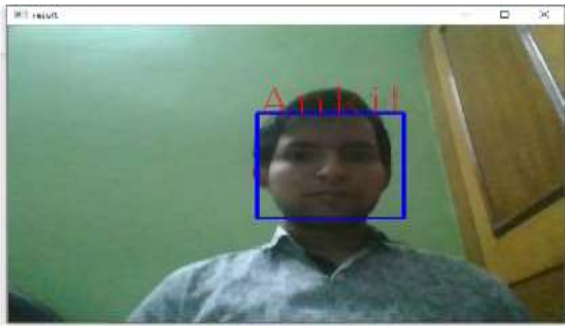


Fig.3 Example of Face Recognition

Module 2: Gender Recognition Algorithm

It is utilized to detect age and age of clients who passes by dependent on online face examinations. Sex character is the individual feeling of one's own sexual orientation. Sex character can relate with an individual's doled out sex during childbirth or can contrast from it. Sex articulation ordinarily mirrors an individual's sex character, however this isn't generally the situation. While an individual may communicate practices, mentalities, and appearances reliable with a specific sexual orientation job, such articulation may not really mirror their sex personality. The term sex personality was initially instituted by Robert J. Stoller in 1964[10].Gender acknowledgment utilizing OpenCV's fisherfaces usage is very mainstream and some of you may have attempted or found out about it moreover. Be that as it may, in this model, I will utilize an alternate way to deal with perceive sex. This technique was presented by two Israel scientists, Gil Levi and Tal Hassner in 2015. I have utilized the CNN models prepared by them in this model. We are going to utilize the OpenCV's dnn bundle which means "Profound Neural Networks".

Gender Recognition Algorithm

- **Step 1** : Import all the required libraries.
- **Step 2** : Often, we have to capture live stream with a camera. OpenCV provides a very simple interface to this. We can capture the video from the camera, convert it into grayscale video and display it.
- **Step 3** : Using set() I'll set the height and width of our video frame.
- **Step 4** : Create 3 separate lists for storing Model_Mean_Values, Age and Gender.
- **Step 5** : I have defined a function to load caffemodel and prototxt of both age and gender detector, these are basically pre-trained CNN models which will do the detection.
- **Step 6** : Now we will perform face detection, Age detection, and Gender detection and for that create a function video_detector(age_net,gender_net) inside your main function and pass age_net and gender_net as its parameters.
- **Step 7** : Read the cap object which is created from VideoCapture() in step 3.
- **Step 8** : Convert the image to gray image as OpenCV face detector expects gray images.
- **Step 9** : Load the pre-built model for facial detection.
- **Step 10** : Now, how do we detect a face from an image using the CascadeClassifier ?
- **Step 11** : Predict the gender.
- **Step 12** : Predict the Age.
- **Step 13** : Now we have to save into the database.

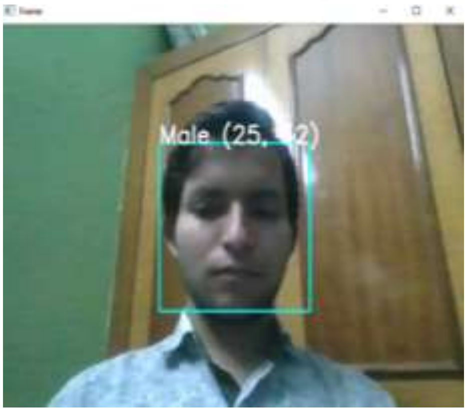


Fig.4 Example of Gender Recognition

Module 3: Working Principle of Drone

A Flying robot or a vehicle without a human pilot ready and a sort of unmanned vehicle. UAVs are a segment of an unmanned airplane framework; which incorporate a UAV, a ground-based controller, and an arrangement of correspondences between the two. An automaton controller works by imparting a radio sign from the remote control to the automaton, which instructs the automaton. Radio signs are sent from the radio transmitter in the automaton controller and got by the automaton's receiver[11]. This is the reason the automaton controller is now and again essentially called the automaton radio transmitter or the automaton radio controller

Working Principle of Drone

1. Firstly, we have to make a frame of light weight material such as, Aluminium frame.
2. Quadcopter is a device with a intense mixture of Electronics, Mechanical and mainly on the principle of Aviation.
3. The Quadcopter has 4 motors whose speed of rotation and the direction of rotation changes according to the users desire to move the device in a particular direction (i.e. Takeoff motion, Landing motion, Forward motion, Backward motion, Left motion, Right Motion.)
4. The rotation of Motors changes as per the transmitted signal send from the 6-Channel transmitter.
5. The signal from microcontroller goes to ESC's which in turn control the speed of motor.

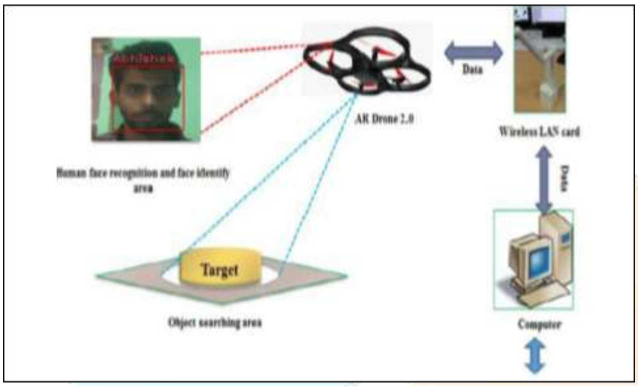


Fig.5 Working of Drone

V.USES OF VISION

- Real time surveillance
- Search and rescue.
- Save human life.
- Provide food, medical kit etc. in disaster situation.
- Used to detect Harmful Gases.
- Used many different areas like monitoring, security, personal drone delivery etc.

VI. ADVANTAGES

- Live Streaming and monitoring.
- Save lives.
- Drones minimize the obvious dangers and health risks.
- Identify and detect the human face.
- Used in dangerous environments.

VII. WARNING

We should utilize the Drone securely and mindfully consistently, to stay away from any harm or damage being caused to any individual, creature or property close to which you are flying the Drone. In this regard you ought to guarantee that you generally work the Drone in consistence with this Quick Start Guide and our Safe Use of the Drone directions. Parrot additionally advises you that you ought not utilize the Drone for any unapproved or unlawful purposes, as you will in any case be completely subject for any misfortune or harm caused because of such unapproved use.

Risks associated with the use of drones The use of drones on a large scale entails a high risk. The main danger is the fall of a drone from a great height, which may be due to:

1.Discharge of the battery,
2.Damage caused by weather conditions (low air temperature, precipitation),
3.Hitting in an obstacle (tree, building, high-voltage line).

These dangers can be anticipated The battery status and other telemetry information, including temperature can be controlled remotely by the framework. In the event of surpassing the one of the parameters the alert ought to be propelled. This will permit make the move, for example, crisis review the automaton to a branch. Nonetheless, the sensors and programming that dependent on the flight way and on the recognized impediments consistently update the course are liable for the evasion of obstructions. A genuine danger to the automaton, because of its worth is the individuals. It tends to be taken. In this circumstance, it might be useful the limitation work and perceiving the circumstance. Change of the machine course can demonstrate about the robbery. At this case, the automaton can start to take pictures utilizing cameras (sensors) and give a blare hindrance the hoodlum and centering consideration of witnesses [4]. These dangers can be anticipated; subsequently the move ought to be made to forestall their uprising. The battery status and other telemetry information, including temperature can be controlled remotely by the framework. In the event of surpassing the one of the parameters the caution ought to be propelled.

A significant dangers related with the broad utilization of regular citizen drones is connected with security. These gadgets have the capacity of following for the followed object and to watch it from a wide range of perspectives[12]. They can be outfitted with cameras, night vision gadgets and different sensors, encouraging sneaking around. While their wide utilizing by metropolitan administrations (counting the police) to control the regular citizen populace, could represent a genuine threat to human rights. Potential dangers related with the across the board utilization of automatons require the utilization of complex arrangements and the to human rights. presentation of intentional guideline focusing on compelling assurance of residents' security.

VIII. CONCLUSION

This paper concludes that the disaster is a sudden accident or a natural catastrophe that causes great damage or loss of life. So using the VISION, provides applicable information and assistance, therefore we need to build a procedure of disaster information collection and build. They can provide an instant telecommunications infrastructure, assist in telemedicine-enabled clinical services, ,perform equipment / drug / patient delivery, enhance search and rescue efforts, assess damage and map disaster zones. Base on this we can get effective information at appropriate time and provide the reference for the phases of disaster-preparedness, and its response. The small scale VISION is cost effective and has many applications. This brief look at the emerging technologies merely scratched the surface of employing VISION in a strike role.

IX. FUTURE WORK

Our general plans were excessively eager. Our group objectives were to configuration, test, and manufacture a quadcopter unit, interface an information logging framework, and plan and execute AI orders for our quadcopter. Shockingly, we couldn't finish every one of these objectives. Later on, if time licenses, we might want to actualize these orders. There were numerous issues that we didn't represent all through the venture. These issues deferred our advancement and we couldn't complete what we set out to do first and foremost. We have discovered that we should permit time for taking care of issues later on.

Different plans later on incorporate including a sonic sensor for progressively precise elevation assurance. Right now the main techniques to decide elevation is by utilizing the barometric weight sensor and the GPS collector. There is no real method to securely decide the quadcopters elevation comparative with its arrival surface. A sonic sensor could take care of this issue, and be utilized to help the auto-arrival order.

Another tentative arrangement would include more strategies for gathering information. Numerous ports despite everything stay unused on the control board. Including a camera could consider computerized photographs or video to be taken. Adding some approach to stream information from the quadcopter to the controller could be another incredible element to add to our quadcopter, this would consider significantly simpler access to the information gathered by the quadcopter. Advanced mobile phone capacities could be another element our gathering might need to include what's to come.

REFERENCES

[1] https://en.wikipedia.org/wiki/Unmanned_aerial_vehicle

[2] https://dronebotworkshop.com/how-does-a-quadcopterwork/

[3] https://sites.google.com/site/uavuni/references-1

[4] https://internetofthingsagenda.techtarget.com/definition/drone

[5] https://www.wired.com/2017/05/the-physics-of-drones/

[6] https://www.dronezon.com/learn-about-dronesquadcopters/what-is-drone-technology-or-how-does-dronetechnology-work/

[7] https://en.wikipedia.org/wiki/Quadcopter

[8] https://uavcoach.com/how-to-fly-a-quadcopter-guide/

[9] Schweier, C. and Markus, M. (2006) Classification of Collapsed Buildings for Fast Damage and Loss Assessment.

[10] https://www.mydronelab.com/blog/how-to-build-a-drone.html

[11] http://dronenodes.com/how-to-build-a-drone/#tab-con-28

[12] http://beginnerflyer.com/build-a-drone/