Drone(UAV) based Intelligence Surveillance System for Disaster Management

A Project Report
Submitted in partial fulfillment of
the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

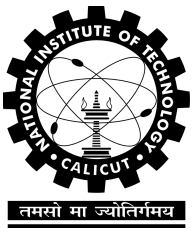
by

Guda Harsha Vardhan	B160342EC
Guntur Srivenkat	B160725EC
Arun Jarapala	B160530EC
Kunal Gupta	B160763EC
Harsha Vardhan Gotru	B160685EC

Under the guidance of

Dr.Praveen Sankaran

Dr.Sudheer A P



Department of Electronics and Communication Engineering
National Institute of Technology Calicut
Calicut, Kerala, India – 673 601

2019-2020

Dedicated to all.

Acknowledgments

We would like to extend our sincere gratitude to our project guides Dr Praveen Sankaran and Dr Sudheer A P for their support during the entire project. The completion of our project would not have been possible without the valuable suggestions from the evaluation panel. We would like to acknowledge the support of Robotics Lab for providing its resources for us to carry out our project work. We would also like to acknowledge the Electronics and Communication Department for extending its full support during the entirety of the project.

Declaration

We hereby declare that except where specific reference is made to the work of others, the contents of this project report are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This project report is our own work and does not contain any outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Guda Harsha Vardhan (B160342EC)

Guntur Srivenkat (B160725EC)

Arun Jarapala (B160530EC)

Kunal Gupta (B160763EC)

Harsha Vardhan Gotru (B160685EC)

NIT Calicut
27/June/2020

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the project report entitled **Drone(UAV) based Intelligence Surveillance System for Disaster Management** submitted by **Guda Harsha Vardhan (B160342EC), Guntur Srivenkat (B160725EC), Arun Jarapala (B160530EC), Kunal Gupta (B160763EC), Harsha Vardhan Gotru (B160685EC)** to National Institute of Technology Calicut for the award of the degree of Bachelor of Technology in Electronics and Communication Engineering, is a bonafide record of the project work carried out by them under my supervision and guidance. The content of the project report, in full or parts have not been submitted to any other institute or university for the award of any degree or diploma.

Praveen Sankaran, Assistant Professor
Department of ECE
NIT Calicut

Sudheer A P, Assistant Professor,
Depatment of Mechanical Engineering,
NIT Calicut
(*Project Guides*)

Dr Deepthi P P
Head of the Department of ECE
NIT Calicut

NIT Calicut
27/June/2020

Abstract

Nearly 85% of Indian terrain is prone to calamities such as earthquakes, floods, landslides. Thousands of people lose their lives due to inaccessibility to the information of their where-abouts after the disaster. To tackle the sudden disasters one should always be ready to face those situations. The recent advances in the design of smaller computers and commendable computation power can help tackle the problem with less intervention of humans. We aim to design an autonomous Unmanned Aerial Vehicle(UAV) system to detect people in distress and send the information of the people to an already established ground control station(GCS). It consists of a long range communication system to communicate with GCS and a system to perform computer vision, avoid obstacles and navigate autonomously and plan its path accordingly. Further it can be used to build a swarm of such systems to cover a wider area and effectively help more people in distress. It involves development of a fully operational drone with PID control of drone, computer vision, image processing, GPS tracking and navigation, data transfer and acquisition.

Contents

List of Figures

Chapter 1

Introduction

India, the second most populous country is one of the most disaster prone countries in the world.According to the statistics[31], 68% of India's land is prone to drought, 60% to earthquakes, 12% to floods and 8% to cyclones, It is surprising that yet India has not formulated the national plan for disaster management till date, according to the CAG Report of 2013. The Indian administration has been a little too late in recognizing the importance of a National Disaster Management Plan (NDMP) except the State of Gujarat.

India is sensitive to over 30 different disasters having a heavy toll on economic , social and human resource capacity and long-term effects on growth, development, productivity and macroeconomic performance. And it should be noted that there is a strong connection between citizens' vulnerability and capacity.

It is high time that Civil Defense 's position in the disaster management process must be strengthened and an effective National Disaster Management Plan must be developed. Even now, local-level communication systems have not been built much. There are no Standard Operating Procedures for National Disaster Response Force deployments. There have been several instances where there has been a relief and rescue mobilization but the harm will already have been done by the time the teams meet.

The present government has to take up the implementation of NDMP at the ground level seriously. The biggest challenge in this is also to motivate the real estate construction companies to also construct affordable range housing in conformity to disaster resistant standards. Today, it is mostly the high end buildings that are seismic resistant. Most people do not die of earthquakes per say but due to the collapse of buildings in such an eventuality. Also, we have to change the way we look at the relief-response paradigm from being a

brief and temporary process, as in most cases it is a prolonged process that dislocates people and renders them difficult from getting back to a normal life.

Having the above mentioned bottle-necks in rescue process, there is a need to further make rescue process swift and efficient. This can be achieved by the sophisticated technologies the world is enjoying in recent times. in The proposed system adds an aid to already existing disaster management tools and makes rescue more safe and easy in remotely inaccessible areas.

1.1 Literature Survey

Liang Zhao *et al.* [8] presented an algorithm that detects pedestrians in a cluttered scene. This is achieved by a combination of stereo-based segmentation and neural-network-based recognition. It involves the segmentation of the image into sub-image object candidate based on the discontinuity in disparities. The sub-image object candidates are splitted into smaller sub-images that satisfy a particular object's size and shape constraints. The object recognition is achieved by using different intensity gradients of candidate sub-images. These are given as the inputs to a trained neural network which generates the output as the recognised object.

The detection of faces and people in the real-world causes challenging problems. This is because there may be different varities of colour, texture and the surrounding background objects. Based on such use of an over-complete dictionary of basis functions and the combination with statistical learning techniques, a framework for object detection in the above said kind of clumsy areas of data has been presented in the paper[13].

L. Apvrille *et al.* [25] and his team used a autonomous drone for assisting rescue services within the context of natural disasters. The drone capture videos of the surroundings and transmit the videos to a remote computer which process the data and retrieves the necessary information.

In another work [6] an intelligent system of autonomous navigation can significantly increase the speed, efficiency and safety of performing reconnaissance, search and rescue operations. It can be achieved by different ways of classification and decision making which is further realized by the usage of various kinds of logics. These include neural networks of deep learning, systems of neuro-fuzzy inference and impulse based neural networks. The

extraction of information is the main aspect based on which the classification and recognition efficiency of the system is determined.

The research [7] creates an autonomous Micro Air Vehicle(MAV) device that can traverse forest trails under the canopy using deep neural networks (DNNs). The Vision modules allow the MAV to detect and avoid trailing people and pets.

D. Camara *et al.* [11] suggested a drone-based solution to aid in the search and rescue of disaster situations by offering a temporary communication framework, generating up-to - date maps of the affected area and looking for hot spots where emergency teams may be more likely to locate victims.

Elizabeth *et.al* [27] proposed a method that helps in augmenting conservation drones with automatic detection in near real time. The model uses pre-trained VGG16 model and trains a new model on a small dataset which was labelled by the authors. In [12] a system is developed to detect the cattle using drones and Convolutional Neural Networks.

Ajith Anil Meera *et al.* [17] proposed an algorithm to check Unmanned Aerial Vehicles for goal. The paper simultaneously trades between coverage, avoidance of obstacles, target re-observation, altitude-dependent sensor efficiency, flight time and FoV to generate the optimal, finite horizon 3D polynomial path in an obstruction-filled setting. The proposed layered optimization approach promotes a balanced exploration-exploitation strategy that makes it resilient against false detection. The algorithm was adapted for a broad variety of environmental conditions and obstacle densities.

The research in the paper [15] discusses about the advantages and disadvantages of LoRaWAN and implementation of it in a drone.The paper [16] investigates the potential security vulnerabilities in LoRa. In particular, analyzing the LoRa network stack and discussion of the possible susceptibility of LoRa devices to different types of attacks using commercial-off-the-shelf hardware is done.

LoRa and XBee are used in parallel to function as a transceiver for wireless communication through radio frequency channel. The XBee is operated on less power as well as lesser bit rate. The packet delivery perentage of LoRa is better when compared to XBee because LoRa never fails to deliver the packet to the destination location,i.e, 100 percent successful. However, in the case of XBee, when the data rate is increased gradually, the performance

efficiency of the XBee module decreases considerably[14].

Meera, *et al.*[1] demonstrated an autonomous target search system using drone which uses computer vision to detect humans an various path planning algorithms are studied in the paper, most of the work is carried out as a simulation in an unreal world where all the obstacles are known.

To add more to it such as LoRa communication and develop a system in real world is a task we have taken up on. The following sections walk you through the developments and challenges faced during the course of realising the idea stumbled up on.

Chapter 2

Background

2.1 Autonomous Drones

An Unmanned Aerial Vehicle (UAV) or a drone is a type of aircraft that operates without a human pilot on-board. UAVs are controlled either operated by a pilot at a ground station or operated autonomously by a pre-programmed flight plan. UAVs are replacing the man-made air-crafts or the satellites because of their peculiar feature of capturing high-resolution imagery below the cloud level. These drones are capable of collecting, mapping, processing and automation for computation using inexpensive open source tools.

Flight and navigation are the two basic functions of UAVs. Drones consists of a power source such as battery, rotors, propellers and a frame. These are used to achieve flight. The frame is typically made of lightweight, so that weight is reduced and maneuverability is increased during flight. A drone controller by an operator is used to launch, navigate and land drones. Controllers communicate with the drone using radio waves, including WiFi. Drones also contain other technical components like Electronic Speed Controllers(ESC), cameras, collision avoidance sensors, ultrasonic sensors, flight controller, GPS module, battery, receiver, accelerometer and altimeter.

In recent years, UAVs paved their way into different fields and are serving a variety of recreational, photography, commercial and military purposes. Initially they were used for military purposes such as intelligence gathering,anti-aircraft target practice. But now, they are serving wide range of humanitarian purposes ranging from search and rescue, surveillance,traffic monitoring, weather monitoring and firefighting.

The main features associated to a drone are camera type, video resolu-

tion, megapixels and media storage format, maximum flight time, maximum speeds, hover accuracy, obstacle sensory range, flight logs and altitude hold.

2.2 Deep learning

Deep learning refers to the training of neural networks. Neural networks refers to the networks or functions that imitates the workings of the human brain called neurons in processing data and creating patterns for use in decision making. It is a subset of machine learning. Deep learning networks are capable of learning from unsupervised data that is unstructured or unlabelled. There are various types of deep learning architecture. Fully connected neural network, recurrent neural networks, convolutional neural networks.

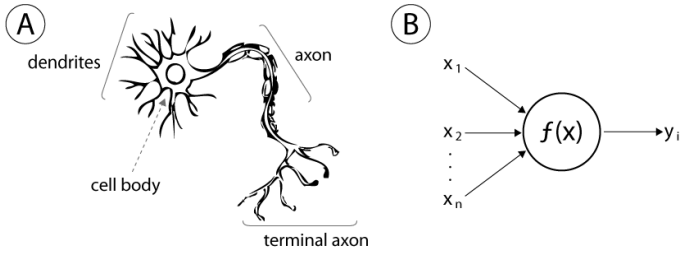


Figure 2.1: Neural Network architecture

Convolutional Neural Networks is the most celebrated DL architecture and it is widely used for image recognition.The pre-processing required for ConvNet is much lower than for other classification algorithms. In CNN, the model is trained with a data set of images and the model extracts the features from the data which is used for prediction.The hidden layers mainly consist of Convolutional layers, ReLU activation functions, pooling, fully connected and normalization layers.

The Convolution layers are the central method for extracting the features from the input image. It is a mathematical operation where the original image represented in pixels is convoluted by a filter inorder to extract the features from the image and also to reduce the size of the image. Filters also called as Kernels are randomly generated using the back-propagation algorithm. The back-propagation algorithm aims to change all of the network weights repeatedly to bring the actual output closer to the target output.

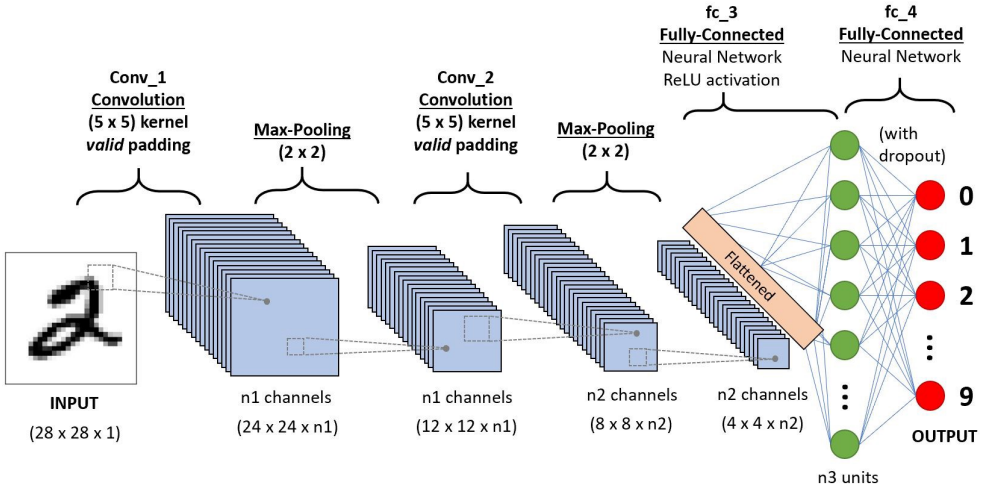


Figure 2.2: Basic CNN architecture

The process is done in such a way it reduces the error for each output neuron and thereby reducing the overall error of the network. Rectified Linear Unit(ReLU) is used as the nonlinear activation function. Activation function allows the model for the development of complex mappings between network inputs and outputs, which are important for the learning and modeling of complex data such as images, video, audio, and non-linear or high-dimensional data sets.

In CNN, convolution layers are followed by pooling layers. Pooling layers further reduces the output dimensions of the convolutional layers retaining only important pieces of information. This reduction in size is required to reduce the number of parameters to be learned to prevent overfitting. Max-pooling selects only the highest pixel value in the feature map, whereas the arithmetic mean of pixels is measured in mean-pooling(Fig.2.3). In addition, Max-pooling allows edge detection while average pooling extracts are so smooth. Generally average pooling is not a good method because it takes all pixels into account and results in an average value which is not relevant for tasks of object detection.

There is a ' Flatten ' layer between the convolutional layer and the fully connected layer. Flattening converts a 2-D matrix of features into a vector that can be fed into a neural network classifier with full connections as shown in Fig.2.4

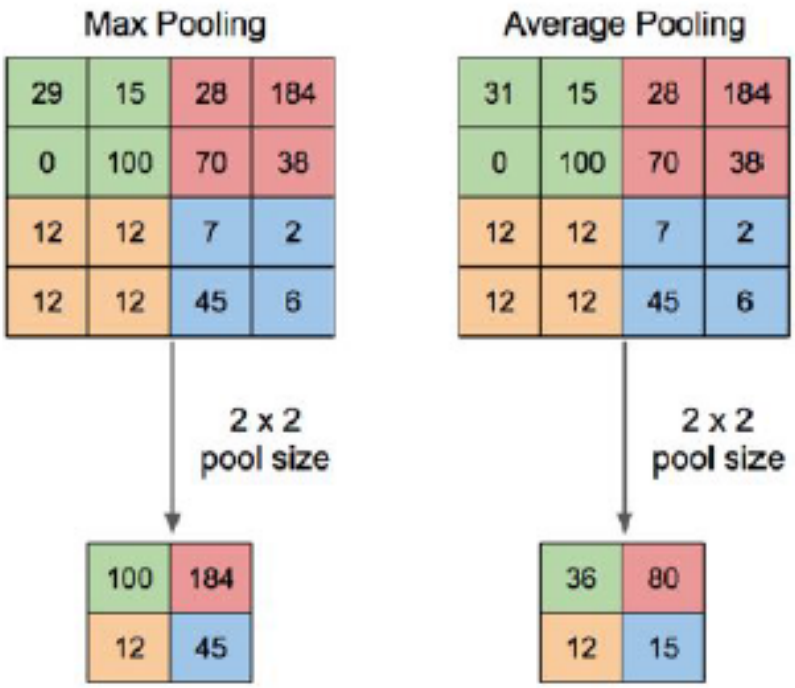


Figure 2.3: Max-pooling vs Mean-pooling.

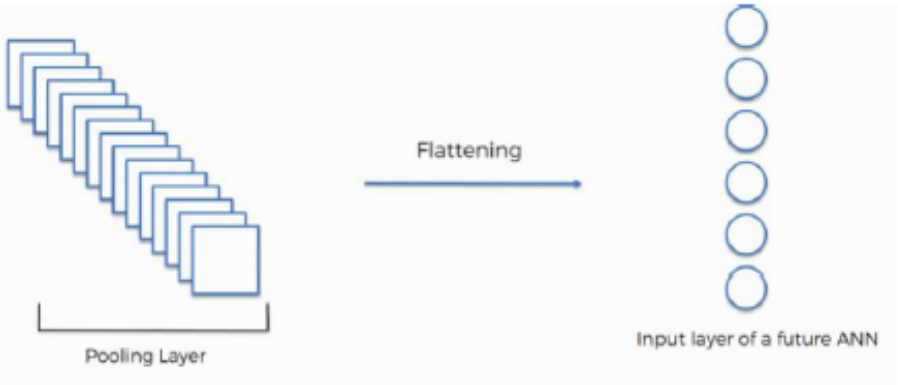


Figure 2.4: Flattening.

In CNN, the softmax function is applied to the very last layer instead of other activation functions like ReLU, sigmoid, tanh. The softmax function converts the output of the last layer in the neural network into a probability distribution.

The UAV flies over the pre-planned area of the disaster, and the camera mounted on the UAV takes pictures of the area under inspection bringing a vital part of the desired information. If the information is of less complexity, manual extraction is sufficient. If the information in the data is of greater complexity, the manual extraction is not sufficient. In these situations, deep learning has proved to be a sufficient substitute.

2.3 Computer vision

Computer vision is essentially characterized as a field of research aimed at developing new techniques that help computers closely track and recognize the content of digital images such as photos and videos. This is a multi-disciplinary area that could be loosely referred to as a sub-field of artificial intelligence and machine learning that could require the use of specialized methods and the use of generation. A given system of computer vision may require that image processing be applied to the raw input.

The most popular computer vision applications involve trying to recognize things in photos such as object classification, identification, verification, detection, segmentation, recognition and landmark detection. Other uses include information retrieval and image-finding.

A typical use of computer vision for digitizing handwritten material is handwriting recognition. In autonomous vehicles, much of the underlying technology relies on analyzing the multiple video feeds entering the car and using computer vision to analyze and pick a path of action. Medical sector is one of the major field where computer vision can aid. Most of the treatment is image analysis, such as x-ray readings, MRI scans and other medical forms.

2.4 LoRa Communication

LoRa (Long Range) is a spread-spectrum modulation technique derived from chirp spread spectrum (CSS) technology and is the first low-cost application for commercial use of chirp spread spectrum. It was established by Cycleo of Grenoble, France, and acquired by LoRa Alliance founding member Semtech in 2012.

LoRa, in essence, is a clever way of getting really good sensitivity of the receiver and low bit error rate (BER) from cheap chips. This means devices with low data rates can get a longer range using LoRa rather than using other comparably priced network technologies. The LoRa modulation and network interface has been developed and configured to provide precisely the type of connectivity required for remote IoT and M2M nodes.

The physical layer of LoRa or PHY is crucial to network activity. It governs the aspects of the RF signal transmitted between the nodes and endpoints, i.e. the sensors and the gateway of LoRa where signals are received. The major areas affected by the physical layer are the frequencies, modulation type, power rates, communication between the transmitting and receiving components.

LoRa system is also known as an open protocol stack. The various types of companies involved in the development, use and implementation of LoRa have been able to come together to create an easy-to-use, low-cost networking solution for all manner of connected IoT devices. This was only possible because open source stack was built.

Certain components of the network infrastructure, aside from the LoRa wireless system's RF components, include the overall system design, backhaul, server, and device computers. The architecture as a whole is also called LoRaWAN.

2.5 Tensorflow

TensorFlow [29] is an open source artificial intelligence library, primarily used as a software tool for implementing Deep learning applications. It is called tensorflow because it takes tensors(multidimensional array) as input.

Tensorflow allows developers to create large-scale neural networks with many layers. It contains preprocessing the data, building the model, training and estimating the model. It is mainly used for classification, perception, understanding, discovering, prediction and creation.

2.6 Darknet/YOLO

Darknet-53 is made up of 5 different sets of convolutional layer. Different sets of these layers are repeated for different number of times and are separated by few convolutional layers with stride (2,2). These convolutional layers with strides (2,2) will help in reducing the output size. The main purpose of using these convolutional layers instead of Maxpool layers is to reduce the loss of features.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 2.5: Darknet - 53.

Though maxpooling layers have been conventionally used in many image

based applications of neural networks, it contributes to small deviations in the spatial dependencies. In order to increase precision of object detection, maxpool layers are not used, instead a traditional convolutional layer is used and the stride is set to (2,2) which gives both reduction in size and there is no loss in spatial dependencies.

In the following chapter, a detailed discussion of drone and its specifications are done along with the results of manual and autonomous flight.

Chapter 3

Flight of Drone

The designed drone has four propellers powered with four Brushless DC (BLDC) motors with an rpm of around 13,000 In these four propellers, two propellers rotate in clockwise direction while the other two rotate in counter clock wise direction, this set up cancels out the rotational torque in the z-axis. The propellers used are "1045" meaning that the length of propeller is 10" and each rotation of the propeller gives a pull of 4.5" inch forward.

Drone is fabricated in the 'X' model as shown in the figure 3.1, rotors and propellers are kept in clockwise(CW) and counter clockwise(CCW) alternatively to form an 'X' with respect to the forward of the flight controller.

3.1 Drone specifications

Calculations made using a Lithium Polymer (Li-Po) battery of 3 Cell 1800mAh battery of 11.1V, total drone weight of 850 grams with a 1045 propeller gives the following results, calculations are carried out using ecalc**??**.

Table 3.4 gives the specifications of battery such as load, voltage, rated voltage, energy, total capacity, used capacity, minimum flight time, mixed flight time, hover flight time for the 1800mAH battery used on the drone. Table 3.2 gives details about the power consumption such as current, voltage, revolutions, electric power, mechanical power, efficiency.

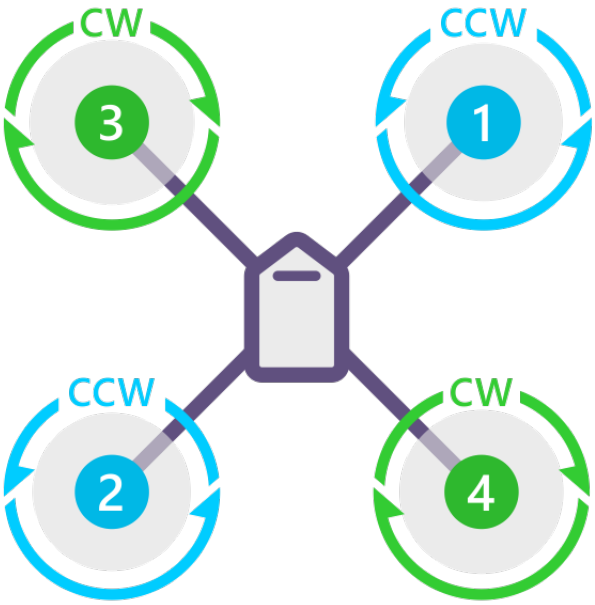


Figure 3.1: 'X' configuration of the drone.

3.2 Flight Controller

PixHawk(PX4) is used as the flight controller for the flight and control of the drone.

PixHawk is an open-source autonomous hardware developed on STM32, an ARM microprocessor. It is low cost and can be easily accessible to develop hobby drones. It can also be integrated with several other sensors such as gimbal, camera, GPS or additional gyroscopes to make the developed drones robust.

3.2.1 Dronekit

Dronekit is an open-source python library which can be used to carry out autonomous maneuvers with PX4 as hardware. It can also be used to do software in loop and hardware in loop simulations, which helps to get an outlook without even flying the drone and risking damages.

Load	46.69 C
Voltage	9.56V
Rated Voltage	11.10 V
Energy	19.98 Wh
Total Capacity	1800 mAh
Used Capacity	1530 mAh
min. Flight Time	1.1 min
Mixed Flight Time	5.0 min
Hover Flight Time	9.5 min
Weight	153 g

Table 3.1: Battery Specifications.

Current	5.77 A
Voltage	10.64 V
Revolutions	13637 rpm
Electric Power	61.4 W
Mechanical Power	51.4 W
Efficiency	83.7 %

Table 3.2: Motor's optimum efficiency.

3.3 PID control

Drones are highly unstable due to fact that it has 6 DoF and requires a flight controller to make it fly in a stable manner with a PID control loop taken care by the flight controller while the values of Proportional (P), Integral (I), Differential (D) have to be tuned in a way to withstand the wind currents in that particular area. A good PID control can achieve a better altitude, hover and can also be extended for the autonomous flight of the drone. It can be able to fly safe and stable in the act of wind currents and only a mild reactions can be observed.

The PID control loop can be seen in the figure 3.2 where the set point is the stable angle (taken from the flight controller), current angle is the angle sampled at that instant from the gyroscope in the flight controller and the output is the Pulse Width Modulation (PWM) values given to the rotors of the drone.

$$error = currentangle - setpoint \tag{3.1}$$

Current	21.01 A
Voltage	9.44 V
Revolutions*	8610 rpm
electric Power	198.3 W
mech. Power	127.0 W
Power-Weight	933.0 W/kg
Efficiency	64.1 %
est. Temperature	67 °C
Wattmeter readings	
Current	84.04 A
Voltage	9.56 V
Power	803.4 W

Table 3.3: Motor operating at maximum speed.

Current	2.43 A
Voltage	10.91 V
Revolutions*	4047 rpm
Throttle (log)	24 %
Throttle (linear)	36 %
electric Power	26.5 W
mech. Power	18.5 W
Power-Weight	126.7 W/kg
Efficiency	70.0 %
est. Temperature	30 °CC
specific Thrust	8.03 g/W

Table 3.4: Motor Hovering at an altitude.

This is calculated in a loop until drone is in air. To summarize PID control loop takes in the values of gyroscope/angle (i.e., sensor data from the sensors placed on the drone or the flight controller) to give out PWM values for the each rotor through an Electronic Speed Controller (ESC) to keep it stable in the air during its flight.

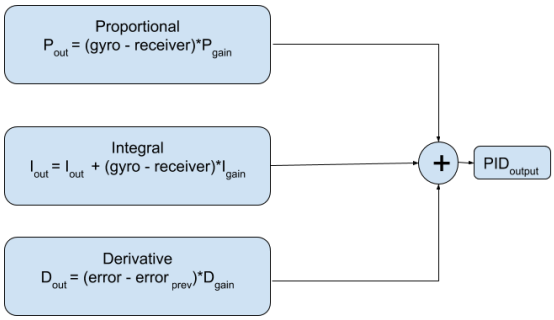


Figure 3.2: A block diagram of PID loop.

3.3.1 Tuning the PID controller

There are three axis for the drone namely,
Pitch-axis: The drone tilts forward or backward.
Roll-axis: The drone tilts to the left or right side.
Yaw-axis: The drone revolves viewed from above counter or counterclockwise.

Initially the values of P, I, D are set to be zero and the tuning process can be started from here.

When only the P-controller is used the drone keeps oscillating in the centre position as there is no friction to control it's oscillations due to the over compensation of the error. The I-controller alone also oscillates but at a slower rates due to the over compensation of the error when compared to the P-controller. The D-controller is also important for a drone which acts like friction to both the P and I controllers. D controller is triggered only during a change of error occurs. It creates a smooth transition unlike the oscillations in the P and I controllers as it is dealt with the error difference.

Figure 3.2 shows the block diagram of the PID loop, where gyro is the angular velocity measured by the gyroscope and receiver is the receiver output.
Code snippet to realise a PID loop is given below,

```
pid_error_temp = gyro roll input - pid roll setpoint; //error =
    gyro - receiver
pid_i_nem_roll += pid_i_gain_roll * pid_error_temp:

if (pid_i_mem_roll > pid_max_roll)pid_i_mem_roll = pid_nax_roll;
```

```
5  else if (pid_i_mem_roll < pid_max_roll *-1)pid_i_mem_roll =
       pid_max_roll* -1;
6
7  pid_output_roll - pid_p_gain_roll* pid_error_temp +
       pid_1_men_roll + pid_d_gain_roll8 (pid_error_temp-
       pid_last_d_error);
8
9  if (pid_output_roll > pid_max roll)pid_output_roll =
       pid_max_roll;
10 else if (pid output_roll < pid_max_roll*(-1))pid_output_roll =
       pid_max_roll*(-1);
11
12 pid_last_d_error = pid error_temp:
```

Centre of Gravity (CG) plays an important role in stability and PID tuning of a drone. Utmost care has to be taken to keep the CG of the drone at the symmetrical centre of the drone. This makes it easier for the controlling of the drone. As it has been dealt with lot of external components such as Jetson Nano, Gimbal camera on the drone, these components are placed judiciously to coincide the CG with symmetrical centre of the drone.

3.4 Autonomous Flight

A drone is said to be autonomous when it can safely do a maneuver in the air without the intervention of a human or commands from some source either through Radio Control(RC) or a ground control station(GCS). During the maneuver it has the knowledge of the GPS location of the target and the drone plans the path to be followed considering the geofences in the region and is successfully able to navigate to the target location with the given specified altitudes at different regions.

There has been a substantial amount of research carried out in the domain of autonomous navigation of drones, such as amazon uses autonomous delivery drones to delivery packages, UPS has partnered a start-up to delivery vaccines[30] in underprivileged areas in Rwanda, African Continent.

3.5 Results

3.5.1 Manual Flight

A manual flight test was carried out in NITC ground using the initial setup shown in the Fig. 3.3. To compensate the wavey shakes created by the vibrations of propellers, a gimbal setup was attached along with the camera. During this flight test video feed is given to Jeston Nano and the trained YOLOv3-tiny on darknet is used to predict the live video feed. This is used to count number of people in each of the frame. A snapshot is taken during the flight and processing is shown in the figure 9.2

3.5.2 Autonomous Flight

Dronekit API for PixHawk flight controller is used to control the flight of drone in Python. It is tested to raise to a particular height and hover for a while. However there were some errors in barometer readings and GPS altitude readings, i.e. both did not match with each other resulting in raise of drone to an ten times higher level than specified and sudden crash leading to wreck of fabricated stand support which was again fabricated.

Figure 3.3: Prototype of the drone system.

In the following chapter, it is discussed about the person recognition and pose estimation using YOLOv3, details of the single object recognition

through modification of final layers. A basic version to detect pose based on the ratio of height and width of bounding boxes of the person.

Chapter 4

Object Recognition for Person

4.1 You Only Look Once(YOLO)

The principle objective of this project is to detect humans using visual data during disasters. But at a rudimentary level, we are performing an object detection and recognition task, where the object being detected or recognised is a human. So an object detection model had to be developed which can perform the required task effectively in a disaster scenario. In order to fulfil the needs of the project, the YOLO object detection model was chosen because of its high accuracy and speed. The basic YOLO model was then altered so that it can detect only humans in the scene.

YOLO stands for You Only Look Once, and it is a deep learning architecture developed by Joseph Redmon based by improvement of Faster R-CNN(Region Based Convolution Neural Network). YOLOv3 is an improved version of previously mentioned YOLO, it is used for object recognition at higher frame rates when compared to traditional Faster R-CNNs like Single Shot Detection. YOLO is chosen so that we have a higher frame rate with a slight compromise with accuracy in recognition of objects.

Literature had been studied extensively to get a note of current research in autonomous UAVs, disaster rescue systems, and Long range communication systems. You Only Look Once(YOLO)[3] object recognition model is used to recognise object. YOLO object recognition model is trained using COCO dataset[4] and tested to recognise objects which mainly include persons. Figure 4.1 shows the recognised objects in a bounding boxes around them. The above model is implemented at 5 FPS(frames per second) on Nvidia Jetson TX2 (8GB integrated GPU) in real-time.

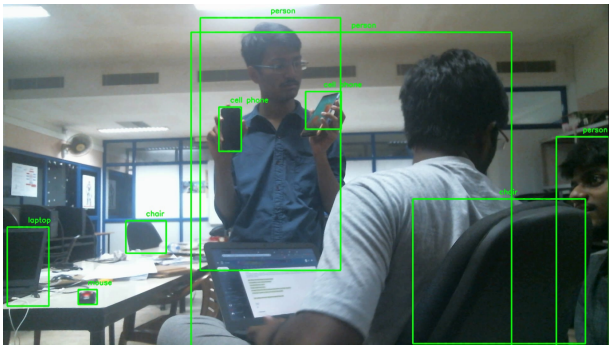


Figure 4.1: Recognised objects using YOLO.

Recognition of person from different of Point of View(POV) has to be tested using the above developed model and frame latency has to be reduced. To achieve it a lighter version of YOLO i.e., *YOLO tiny* can be used but it reduces the accuracy of predictions made on objects.

As discussed in methodology section an annotated VOC dataset is taken, these images are split into two categories namely, train, test and annotations of person are extracted to train YOLOv3 using a single class object(person) on Google Colab using TensorFlow framework. The saved weights are saved on the drive. Fig. 4.2 shows the plot of evaluation and test on the dataset, where evaluation loss represents the forward train loss where evaluation is done through train images, test loss is the loss when the network is given with unknown images from the test data. It can be observed from Fig. 4.2 that the network has a minimum loss for both evaluation and test at 22^{nd} epoch, so the weights extracted at that particular epoch are taken to recognise persons. All the computation are carried out using Tensorflow API with 32-bit floating point operations to preserve accuracy. The count of detected people and their pose is estimated, the basic estimation of their pose is done using the developed model discussed in the previous section.

The single object trained model's accuracy can be increased using data augmentation or by using a bigger dataset, it can also be trained using the data collected from the camera mount on drone during the flight or from the footage of CCTVs which are mounted at higher altitudes. The speed of recognition can be improved by using TensorRT of NVIDIA to convert all operations into 8-bit floating point operations and optimising the networks using the mentioned library with a trade off in accuracy.

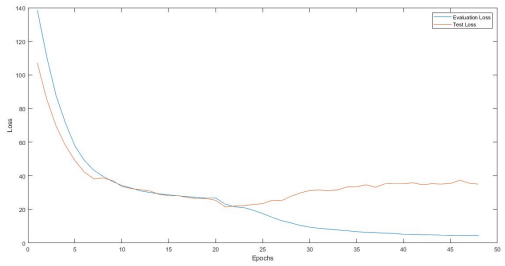


Figure 4.2:

We have worked on object recognition model YOLO in tensorflow and trained on VOC 2007 dataset. However, when it was tried to be ported on to Jetson Nano, there were some memory issues as it has both GPU and CPU embedded on a single chip. So, it is trained again on darknet using the same dataset. It was then ported on to Jetson Nano, it was working at an average of 8 FPS. It has an average loss of 0.48545 and accuracy of 98.01% on training data.

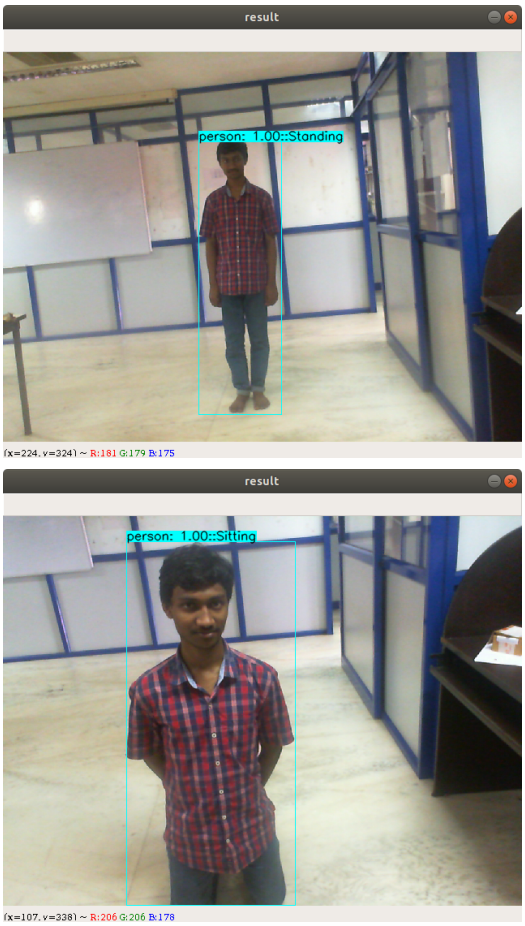


Figure 4.3: a. Pose estimation of person standing b. Pose estimation of person sitting

Chapter 5

Estimation of Pose

5.1 Introduction

As the project progressed, a very important requirement was needed. To determine the condition of the people being detected so that they can be placed in an order of priority to be given based on the severity of their condition and the surrounding situations.

There are different ways a person in a disaster affected areas can be, for example during floods, people can be seen walking in areas filled with water upto their waist level. So, a model was developed for detecting the position of the persons detected and the detected position is them used to determine the level of severity for that particular person. A model named posenet is used to detect the pose of a person. Once the current severity of a person is determined, the details are sent to the ground station so that the disaster relief team can plan accordingly to rescue the people with higher severity level first followed by lower ones.

Posenet detects the pose of a person by first identifying the locations of seventeen differen points in a human as listed in the below table 5.1. A visual representation of all the seventeen pose points are shown in Fig.5.1. These seventeen points are combined and their relative position from each other and the relative angles are used to determine the current pose of the person under reference.

Table 5.1: Pose Points.

Point No.	Pose Point	Point No.	Pose Point
1	Right ankle	10	Left ankle
2	Right knee	11	Left knee
3	Right hip	12	Left hip
4	Right wrist	13	Left wrist
5	Right elbow	14	Left elbow
6	Right shoulder	15	Left shoulder
7	Right ear	16	Left ear
8	Right eye	17	Left eye
9	Nose		

To detect these seventeen different points, posenet uses a combination of convolutional neural network layers. In the application done in this project, a lighter version of convolutional neural networks, the MobileNets are used. This is done to reduce the number of on-board computations required.

Figure 5.1: Pose Points.

MobileNets are light weigth Convolutional Neural Networks (CNNs) which can also run on mobile phones and low powered CPUs. The main aim of mo-

bilenets is to enable large models work on small systems like mobile devices and small embedded applications. MobileNets are built of depthwise separable filters, which reduce the number of computations required when compared to a conventional convolutional neural network.

5.2 MobileNet

As mentioned earlier, the complete computer vision application will be taking place on the drone, so the main challenge is to reduce the number of on-board computations required as much as possible, while not compromising on the performance of the model being developed. So, a clear option was to use the MobileNet, which reduces the number of computations required drastically. It was developed for applications based on mostly small architectures which cannot handle very large number of computations.

The idea in MobileNet is to factorize a single convolution neural network layer, that is to divide the complete process into two sub-process with reduced complexity. The first sub-process takes care of the dimensions of the output and the second sub-process takes care of the number of channels in the output Both combined together gives the required output of required dimensions in less number of computations.

The two sub-process together reduce the number of computations required by a huge factor, but in some applications, the complexity have to be further reduced, in such cases, two separate hyper parameters are used which scale the input image and/or every layer of the modified convolutional neural network to a smaller dimension.

As mentioned above, each convolution layer is factorized into two sub-processes to reduce the number of computations. The two sub-processes introduced earlier are named depthwise convolution and pointwise convolution. The complete details of each sub-process and the hyper-parameters will be discussed in this section.

Fig.5.2 shows a standard convolutional filter along with the depthwise convolution and pointwise convolution. In depthwise convolution, a unique filter is applied to each input channel of the layer. Then the output of the depthwise convolution is given to the pointwise convolution, where the 1×1 convolution is applied across all the channels of the depthwise convolution

output, that is, a linear combination of the output from the depthwise convolution is computed. These two process combined gives the output of same dimension as that of standard convolution filter.

If the input to n^{th} layer is given as $H_n \times W_n \times C_n$, where H_n is the height of the input at n^{th} layer, W_n is the width of the input at n^{th} layer and C_n is the number of channels of the input at n^{th} layer. Now if we consider the filter dimensions being used are $F_n \times F_n \times C_{n+1}$, where F_n is the dimension of the filter and C_{n+1} is the number of filters, that is, number of output channels of the convolution layer. Then the number of computations in case of a standard convolutional layer will be equal to

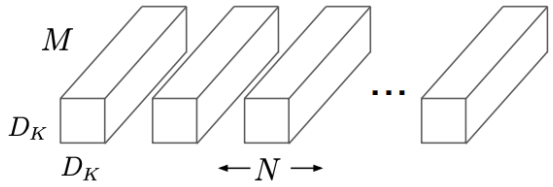
$$H_n \times W_n \times C_n \times F_n \times F_n \times C_{n+1} \tag{5.1}$$

Here, the number of channels after the convolution is given by C_{n+1}.

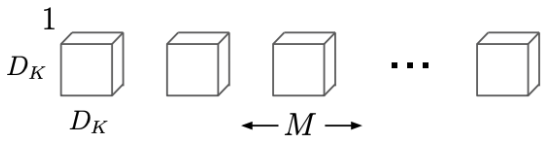
In case of the MobileNet, first for depthwise convolution, the filters being used will be of dimensions $F_n \times F_n \times C_n$, where F_n is the dimension of the filter and C_n is the number of filter. Note that here only the height and width of the output will be achieved, and the number of channels is same as that of the input. For number of channels, pointwise convolution will be used and the number of channels in the output will then be equal to number of channels in output of standard convolutional layer. The number of computations in case of a depthwise convolutional layer will be equal to

$$H_n \times W_n \times C_n \times F_n \times F_n \tag{5.2}$$

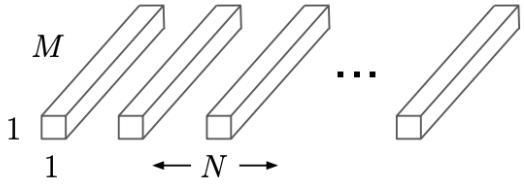
Now, for pointwise convolution, we will be applying a 1×1 convolution accross all the channels of the depthwise convolution output (C_n), that is, a linear combination of the output from the depthwise convolution is computed. So the filter dimensions for pointwise convolution will be $1 \times 1 \times C_n \times C_{n+1}$, where C_n is the number of channels in input to the layer and C_{n+1} is the number of channels in the output of the layer. We see that the number of channels is now same as the number of channels in the output of the standard convolutional layer. The number of computations in case of a pointwise



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters



(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

Figure 5.2: Standard convolution filter, depthwise convolutional filter and pointwise convolutional filter

convolutional layer will be equal to

$$H_n \times W_n \times C_n \times C_{n+1} \tag{5.3}$$

Now, upon combining the number of computation of both depthwise convolution and pointiwse convolution, we get

$$(H_n \times W_n \times C_n \times F_n \times F_n \times) + (H_n \times W_n \times C_n \times C_{n+1}) \tag{5.4}$$

So the decrease in the number of computations will be

$$\frac{(H_n \times W_n \times C_n \times F_n \times F_n \times) + (H_n \times W_n \times C_n \times C_{n+1})}{H_n \times W_n \times C_n \times F_n \times F_n \times C_{n+1}} \tag{5.5}$$

which upon simplification becomes

$$\frac{1}{C_{n+1}} + \frac{1}{F_n^2} \tag{5.6}$$

Eq.5.6 gives the total reduction in the number of computation when the combination of depthwise convolution and pointwise convolution is used instead of the standard convolution.

In MobileNet, the batch normalization and the activation layer are placed after both, the depthwise convolution and the pointwise convolution. Whereas in a standard convolutional layer, the batch normalization and the activation layer are used only once. A comparison between the standard convolutional layer and the MobileNet convolutional layer is shown in Table. 5.2.

Table 5.2: Conventional convolutional layer and MobileNet layer comparision.

Standard	MobileNet
Conventional filter (maybe 3×3)	Depthwise convolution (maybe 3×3)
Batch Normalization	Batch Normalization
Activation function (ReLU)	Activation function (ReLU)
	Depthwise convolution (1×1)
	Batch Normalization
	Activation function (ReLU)

In some cases, the number of computation have to be reduced even fur-

ther to match the applications requirements. So, with some compromise on the accuracies, a scaling factor is used for the input image and/or each convolutional layer. The scaling factors are called the hyper-parameters and these can be varied between the range of 0 to 1. These hyper-parameters are called the width multiplier and the resolution multiplier.

In width multiplier, a scaling factor α, the parameter, is used. This factor is used to reduce the width of the complete network in an uniform manner, which will result in a reduced number of computations. The parameter α lies in the range $(0, 1]$, so the network width can be reduced by a factor which is not equal to zero. The number of computations, when the width multiplier is used, reduced by approximately α^2. Since it reduce the width of the network evenly, the number of computations required are

$$(H_n \times W_n \times \alpha C_n \times F_n \times F_n \times) + (H_n \times W_n \times \alpha C_n \times \alpha C_{n+1}) \tag{5.7}$$

The next hyper-parameter is the resolution multiplier, which is denoted by ρ. This hyper-parameter is applied at the beginning to the input image. So the input image is scaled down by a factor of ρ, hence all the layers are comparatively smaller and the number of computations reduce. Even in this hyper-parameter, the range is $(0, 1]$. The number of computations is given as

$$(\rho H_n \times \rho W_n \times \alpha C_n \times F_n \times F_n \times) + (\rho H_n \times \rho W_n \times \alpha C_n \times \alpha C_{n+1}) \tag{5.8}$$

This way, the hyper-parameters reduce the total number of computations further.

5.3 Googlenet

A combination of convolutional pose machines(CPMs) and Googlenet can be a possible solution for the process of human pose estimation. This can be carried out with the help of image sensor data. A few layers of Googlenet are

introduced directly in the first stage of the process involved in CPMs. This generates a response map of each human skeleton's points from the given set of images. For improved performance of the model, deeper network layers and more complex network structures can be implemented to enhance the ability of low level feature extraction. A fine-tuning strategy can also be applied to improve estimation accuracy.

The convergence time is an important factor for any pose estimation algorithm. With the help of the inception structure, the number of parameters of the model can be reduced significantly which thereby reduces the convergence time. Human pose estimation on single 2D human pose estimation datasets is also a challenging task. These datasets include MPII, Frames Labeled in Cinema (FLIC) and LSP. This can be achieved using a convolutional neural network such as the CPM. The model uses CNN for human pose estimation. Its main contribution lies in the use of sequential convolution architecture to express spatial information and texture information.

Multiple stages in the network as a whole form the ultimate sequential convolution architecture. However, in this process, there is a chance that the gradient descent vanishes in the deep network. This can be prevented because of the presence of supervised learning in every stage of the network. The input for the first stage is nothing but the actual image which, in later stages changes that the input taken for each subsequent stage is the feature map available to us. Spatial information, texture information, and central constraints as a single unit is more helpful for our process. This is the main reason behind combining the three. To ensure accuracy and to consider the distance relationship between the key points of each human skeleton, we have used multiple scales to process the input feature map and response map for the same convolution architecture.

Expensive hardware and pre-training on large datasets are required to maximise the accuracy on benchmarks. This, however, leads to highly complex and expensive deep network architectures for proper implementation. This makes it difficult to compare various kinds of methods and check their corresponding results for similarity in the outputs. A solution for this issue is to use a more efficient deep network architecture that can be trained on mid-range GPUs without the requirement of any pre-training. This offers us a method with a very low level of computation requirement of our network.

However, performance-wise, this has the same level of performance compared to the more complex models. The use of a Fully Convolutional GoogLeNet

(FCGN) is the idea available for human pose estimation. The complete system is not used for or method. Instead, the average pooling, drop-out, linear and soft-max layers are excluded. These are generally an important part of the system but are not necessary for our model. We only utilize the first 17 layers for the functioning of our model.

A multi-resolution network can be applied to achieve post estimation. Two seperate FCGNs which consist of shared weights are used. Each of these two FCGNs changes the resolution of the image before taking it as the input for its system. This helps in producing feature maps which thereby improve the accuracy of the pose estimation.There is a drawback to this method that the deconvolution filter is too large. Due to this, exploiting the context of neighbouring pixles in the feature maps is the only way to predict belief maps for the joints in the network. By utilizing a sigmoid function, the normalisation process for the beleif maps is carried out. Another step involved in the procedure is to use spatial drop out prior to upsampling. This helps us to regularise the network perfectly.

5.4 PoseNet

PoseNet is product of high level features. Point based SIFT registration fails when there difficult lighting, motion blur and different camera intrinsics. The main advantage of Posenet is that it can handle all the above three cases with great ability. We can also observe that by applying this method, the pose feature that is produced generalizes to other scenes. This helps in regressing pose in cases in which the model might have only a few dozen training examples. When applied on the indoor 7 Scenes dataset and outdoor Cambridge Landmarks dataset, effective localization can be achieved only with the help of PoseNet.

To validate that the convnet is regressing pose beyond that of the training examples the performance for finding the nearest neighbour representation is shown in the training data from the feature vector produced by the localization convnet. As performance exceeds the convnet is successfully able to regress pose beyond training examples. Proposed algorithm is also compared to the RGB-D SCoRe Forest algorithm.

Saliency maps produced by PoseNet are shown. The saliency map is the magnitude of the gradient of the loss function with respect to the pixel inten-

sities. This uses the sensitivity of the pose with respect to the pixels as an indicator of how important the convnet considers different parts of the image. These results show that the strongest response is observed from higher-level features such as windows and spires.

A more surprising finding, however, is that PoseNet is also highly sensitive to large textureless patches like lane, grass and sky. Such textureless patches can be more informative than the highest answer points since the effect of a pixel group on the pose variable is the sum of the values of the saliency map over that pixel group. This evidence points to the ability of the net to locate information from these textureless surfaces, something that features based on interest points such as SIFT or SURF do not do. The last point is that PoseNet has an attenuated response, effectively masking people and other disruptive objects. Such artifacts are complex, and have been defined as not suitable for localization by the convnet.

Perspective-n-Point (PnP), the most common form of image localization, relies on precise knowledge of the scene distances, which is not always available. Recent developments in deep learning , particularly convolutionary neural networks (CNNs), have resulted in promising pose prediction methods based on image data. We investigate the potential of applying a specific CNN architecture, called PoseNet, to estimate the 6-DoF pose between an underwater vehicle and a fixed point, without artificial markers being needed.

A novel approach for human activities recognition (HAR) based on body articulations (joints) that represent the connection between bones in the human body which join the skeletal system such as the knee, shoulder and hand, and which are made to allow different degrees and types of movement. To implement the proposed system, PoseNet is used to extract articulation points, which will be classified employing transfer learning approach to recognize the activity. The created system will be named in the rest of the paper (PTLHAR). The experimental results show that the proposed approach provides a significant improvement over state-of-the-art methods.

Human pose information can be a very useful tool in a monocular camera based hand gesture recognition system. The system is made up of two separate modules. For extracting the human pose information, PoseNet is used initially. Kalman filtering method is applied for extracting the wrist joint information. Even when there are variations in the background and illumination,we can effectively detect and track the position of hands in the region of interest (ROI). The convolutional neural network (CNN) is used to

classify hand gestures. The hand gesture information can be used for various applications. The human pose information plays a crucial helping hand in this process. In order to demonstrate the usefulness of the system, we build a simple AR camera system in which several virtual objects are augmented on the human face according to the hand gestures.The system gets a 640x480 pixels input color image captured with a webcam. Then, Using the PoseNet, it estimates the posture and obtains keypoint positions and a keypoint reliability scores for user's 17 joints. The keypoint position provides(x, y) coordinates which exist on a two-dimensional plane of the joints. The keypoint reliability score determines the confidence to make sure that the estimated keypoint position is accurate and ranges between 0.0 and 1.0. In cases when illumination or the background is not kept constant, the joint information can be used by the system to detect the hands. To recognize hand gestures, additional information can be incorporated by the system to efficiently interact between the user and the computer. However, it has a high dependency on the PoseNet.

5.5 Results

The above described posenet was used to detect the pose points and then classify the current person's situation as sitting or standing or in debris. The pose points are used to obtain the angle between the points ankle, knee, hip for both right and left sides. If the angle is less than 30 degrees it is classified as sitting else it is classified as standing. If the pose points of legs are not at all detected it is classified as the person is in debris. The severity priority will be set accordingly based on number of people in debris. Model is tested using the images taken from various newspapers during the Kerala Floods 2018.

In Fig.5.3, we can see the person under consideration is being classified as "In debris". The person is half sunk under water and the classifier works properly. In Fig.5.4 as well, the person under consideration is determined to be under debris. In Fig.5.5, we can see that the classifier classifies the current situation of the person as standing, while it should have classified it as "In debris"

In the following chapter, it is discussed how the information extracted in the model discussed so far and the GPS location is displayed on a web-based Graphical User Interface(GUI).

Figure 5.3: Posenet output.

Figure 5.4: Posenet output.

Figure 5.5: Posenet output.

Chapter 6

GUI

6.1 Introduction

The complete information received at the base station is displayed using a Graphical User Interface (GUI). So the GUI plays a very important role and hence it is has to be designed such that the correct information is displayed and the data is in an organized manned. In the complete duration of the project, as various targets were reached, the GUI was constantly updated and optimized such that the complete GUI is user friendly and anyone can easily understand the information being displayed.

6.2 GUI using Python

Initially, a simple basic GUI was developed using Tkinter library in python. Fig.6.1 shows the initially developed GUI. This GUI consists of a few buttons and a few display windows. During the development of this GUI, two XBee S2C module were being used for communication between the drone and the ground station. So, information was being sent and received both ways. So the GUI was designed to take advantage of the two way communication. As told earlier, the GUI was update throughout the project, so different GUIs developed at different stages of the project represent different technical aspects of the project.

The initial GUI had five buttons, each button had an assigned function. When the button was clicked, the task specified on the button is performed. So, upon pressing the takeoff button, the GUI would send the required information to the drone and the drone would takeoff and go to a specified

altitude. Similarly, the GPS location of the quadcopter is displayed in the GUI upon pressing a button.

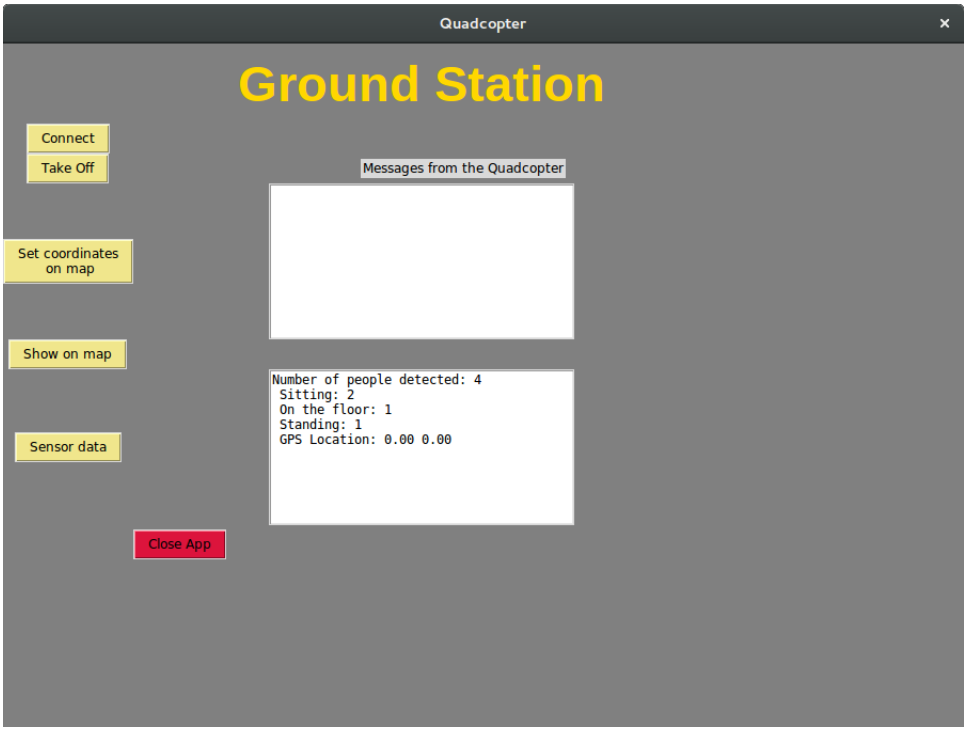


Figure 6.1: Initial GUI.

GUI app developed so far can be improved by integrating it with Google Maps to display the current location of drone accurately from the data received from the drone. LoRa communication's range can be increased by using a better module such as GAMMA 815.

A GUI app is developed using python an Tkinter toolbox, currently this GUI can initiate connection with PixHawk and Jetson Nano, get messages, GPS coordinates from drone and display it on screen, take-off the drone.

6.3 GUI using Web

As mentioned earlier, the graphical user interface was constantly updates as the project progressed, so different new developments were done and the GUI was updated. The major transform in GUI took place when the complete GUI was shifted from a Tkinter based backend, using python library, to a web based application, using HTML backend. The main reason for switching the complete GUI from a python framework to a web based application was to integrate maps on the GUI and for ease of use. The new web based GUI can be run on any system with a web browser installed in it.

Figure 6.2: Latest GUI.

The new GUI is shown in Fig.6.2. It has an integrated map which shows the current location of the drone. As the new posenet was developed by the time the GUI transformation took place, those details were also displayed in the GUI. The information being displayed was changed to the Coordinates of the drone, the current situation of the people at the current coordinates, number of people in debris, number of people sitting, and number of people standing. This will give an overall idea of the situation at the coordinates specified. So the disaster relief team can plan accordingly.

The next chapter discusses about the dataset preparation to collect a similar data to that of the data seen by the drone.

Chapter 7

Dataset to mimic Drone Flight

7.1 Introduction

The data plays a major role in computer vision domain, initially VOC 2007 is used to train on YOLO v3 to recognise persons in a given image. However, the data which is dealt in real time or the images collected by the flying drone differ from the ones collected in the VOC 2007 dataset. The above gap has set in pursuit to create a dataset which is similar to the one which can be seen through the point of view (PoV) of drone as to mimic it's flight. To obtain this kind of data, a camera is installed in the central workshop and data is collected at two different instances of a day i.e., afternoon 2pm and evening 5pm. The data is collected in form of a video, further this video is sampled

7.2 Dataset Preparation

7.2.1 Collection of Video from Central Workshop.

A camera is set up near Central Workshop to record a video which mimics the point of view(PoV) of a flying drone and a video of twenty minutes is recorded to test the developed models till now. It is further planned to annotate the data collected at different places just like Central Workshop to collect more data of people from PoV of drone. This collected data can be used to train a much more efficient network as the data is as real as possible. Fig. 7.1 shows a snapshot from the data collected from the Central Workshop. It can be observed in Fig. 7.2 that the model was only able to predict it at 50% of the times accurately. It can be attributed to the lack of focus beyond a particular distance and the camera being a HD generic camera does not suit

imaging from a farther distances as depicted in this case.

a. Evening at 5pm b. Afternoon at 2pm.

Figure 7.1: Snapshots of video captured at different instances.

a. Evening at 5pm b. Afternoon at 2pm.

Figure 7.2: Predictions on snapshots of video captured at different instances.

The recorded video is sampled at a rate of 5 frames per second (FPS) to create a dataset of images which can be trained on any algorithm developed. The sampled data is labeled with bounding boxes around the person using an online toolbox called labelbox [cite]. To account for the disturbances in images, synthetic noise is added to the images to the sampled data, a Gaussian Blur is added to the images to account for the infinite focus of the HD camera used, this gives an opportunity to work with low cost cameras which saves both cost and power.

7.3 Results

Labels were created for the de-pixelated images to obtain better results. Figure 7.3 shows an instance of labeled data from the sampled data collected at Central Workshop.

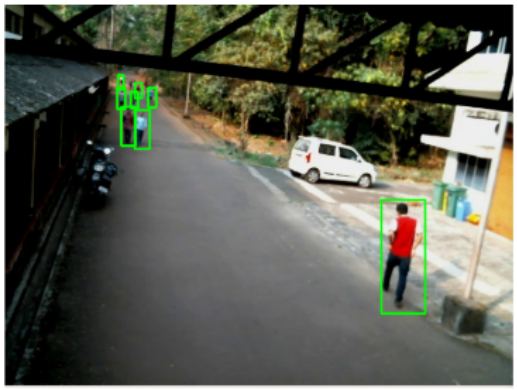


Figure 7.3: An image from labeled dataset of the images collected from Central Workshop.

A Gaussian filter of size 7×7 is used to perform Gaussian blurring of images to synthesize images with infinite focus. A sample of such image is shown in figure 7.4. Gaussian noise is added to the sampled data to obtain images similar to figure 7.5.

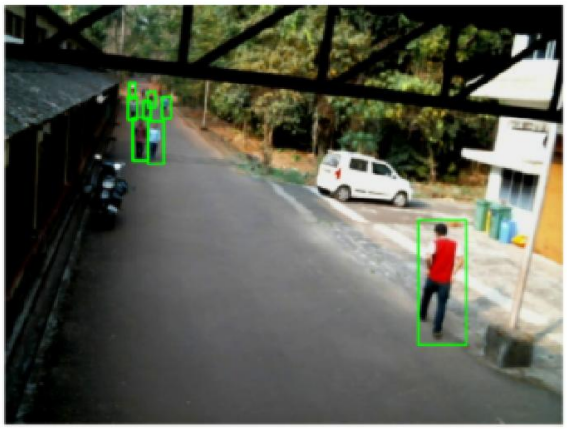


Figure 7.4: An image from labeled dataset of the images collected from Central Workshop.

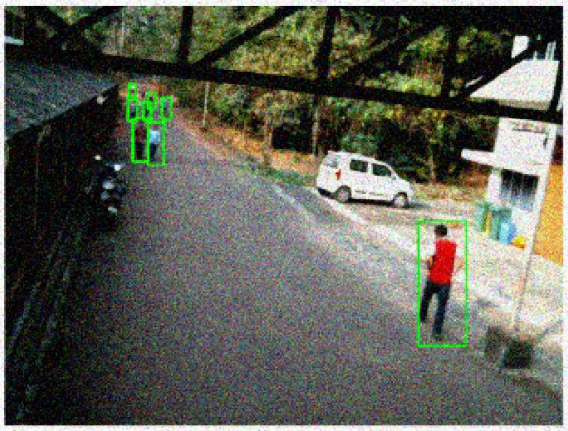


Figure 7.5: An image from labeled dataset of the images collected from Central Workshop.

Chapter 8

LoRa Communication

8.1 Introduction

A low-cost application of chirp spread spectrum (CSS) technology used for commercial purposes is called LoRa. It is a form of spread spectrum modulation technique. The development of LoRa was carried out by Cycleo of Grenoble, France. A founding member of the LoRa alliance named Semtech acquired LoRa in the year 2012.

Low bit error rate (BER) and good receiver sensitivity can be achieved using inexpensive chips through the use of LoRa technology. The range of low-data rate applications can be increased by a considerable margin using LoRa rather than using other comparably priced radio technologies. The design and optimization of the LoRa modulation and radio interface is done in such a way that they carry out similar kind of communications needed for remote IoT and M2M nodes.

The physical layer of LoRa is called PHY. It is one of the most important aspects for the proper operation of the system. The RF signal that is transmitted between the nodes and end-points, i.e. the sensors and the LoRa gateway where signals are received is controlled by this physical layer of LoRa. The parameters that can be affected by a change in the physical layer are frequencies, modulation format, power levels, signalling between the transmitting and receiving elements.

The overall system architecture, back-haul, server and the application computers are some of the other elements which are part of the network architecture excluding the RF elements of the Lora wireless system.. The

LoRaWAN is another name used for the system architecture.

8.2 LoRa Communication using XBee

Xbee is a radio frequency transceiver used for communication between two points. This is useful for short range communications. In this application, Xbee is used for communication between ground station and drone. XBee uses IEEE 802.15.4 protocol to send and receive bits. There is an open source software tool to configure Xbee modules and this tool is helpful to monitor the information being transmitted. In order to make two XBee S2C modules talk to each other, one need to setup one as a Coordinator and one as an End Device (or Router). One module (End Device) is connected to the GCS and another module(Coordinator) is connected to Jetson Nano. Information of the stranded people is sent through XBee at a baud rate of 115200 bps from Nano to the GCS.

8.3 LoRa communication using Gamma-868

GAMMA-868 is the LoRa module that is being used for communication between the GCS and the drone. It has a range upto 16km and it has various modes of operations like Telemetry mode and Serial Data mode. The module has different pins that can be configured to initiate the different functionalities of the module. The information from jetson nano is sent to a LoRa module which acts as a transmitter and this information is sent to another LoRa module which is in receiver mode with the help of an antenna and Arduino UNO.

This information is then sent to a LoRa module from Jetson Nano which is in the transmitter mode. Then the LoRa module communicates with another LoRa module acting as a receiver which is in the ground control station wirelessly. Finally, this information is displayed on GUI.

8.4 Results

The functioning of LoRa module for short range communication is verified and it produced satisfactory results using XBee.

As shown in Fig. 9.1 a camera mounted on gimbal is used to capture the images from the drone so that the gimbal holds the camera steadily and helps to capture clear images. This is a must because camera capturing shaky images is one of the drawback of our model in the previous semester. The image is then fed to a Jetson Nano that runs YOLO-v3 tiny algorithm to extract the annotations of person and then other information like the count and the position in which the people are identified by the drone.

The estimated pose of the persons under observation are performed under laboratory environment(Fig.8.1) and the information is sent from coordinator to end receiver. The required information is processed by Jetson Nano and it is sent to coordinator or transmitting Xbee module.

Figure 8.1: Information processed by jetson nano and fed to Coordinator

The end device is as shown in Fig.8.2 and it is continuously monitoring the pose of the persons under observation. It is also seen that the information from the coordinator to receiver is sent through packets. A console log helps to indicate the pose of the person.

Next step is to achieve long range communication between the drone and the GCS with the help of an antenna and Gamma-868. The LoRa module is configured in Transreceiver mode i.e., one LoRa module is paired to another LoRa module so as to send the information to and fro between these

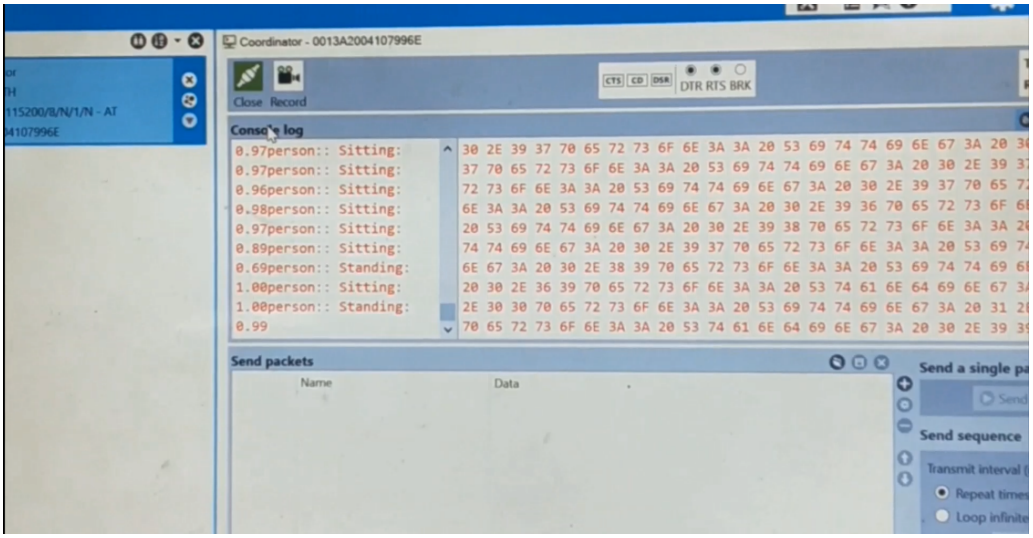


Figure 8.2: End device side

both LoRa module with minimum loss of information and without wasting bandwidth. The information sent is in bytes in the highest speed of 48Kbps, while all the other modes are also tested.

Chapter 9

Results

The flow of work is divided into parts where, computer vision, communication, flight of drone are dealt seperately as shown in the figure 9.1.

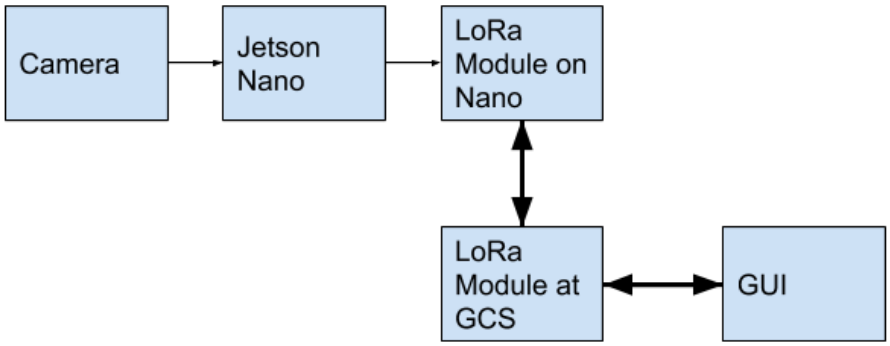


Figure 9.1: Block diagram of flow of work.

A drone with a 3.5kg thrust is assembled with four rotors. It is controlled using a PixHawk (PX4) flight controller, can be seen in figure 3.3. Autonomous flight was achieved for hovering of the flight at a particular altitude, it was done through dronekit, python script was run on Jetson Nano and the communication between Nano, PX4 is carried out by USB port. PID tuning of the drone was done after mounting all the components of the system to compensate the CG shift effects.
The development of computer vision systems to go on with the drone are carried out parallely, initially YOLO was trained on a VOC 2007 dataset with single label of "person", its training and test loss are shown in the figure 4.2. The final aim was to segregate the help required based on the pose of the person, to estimate pose initially the ratio of the bounding box's height and width was considered, results for the same were depicted in the figure 4.3.

Figure 9.2 shows the flight, on board computer vision processing of video feed and estimation of pose.
Later to counter the inaccuracies in estimation of pose and to increase speed, MobileNets were used as the base CNN for the poseNet. The points predicted on the human body were used to estimate pose, even though it worked when the person is facing the camera, it fails to work when the same person did not face camera directly. The different scenarios of pose estimation can be seen in figures 5.3, 5.4, 5.5, where the images taken are from the newspapers during 2018 Kerala Floods.
The details of estimated poses and the level of criticality is displayed on a web based GUI as shown in the figure 6.2.
The dataset the models are trained on and the date given by the camera on drone to the models differ a lot in setting and premise. It lead to a need to collect new data to train and test. Dataset collected is augmented with Gaussian blur and synthetic noise added images as shown in figures 7.4, 7.5.

Figure 9.2: Manual flight and on-board processing.

To summarize the video feed is taken by the Jetson Nano using a camera during a flight and apply poseNet on it to estimate the pose of the person, the estimated information along with the number of people and their geo-location is sent to a Ground Control Station (GCS) which is already setup using LoRa communication module (XBee/Gamma-868) and displayed on an

user friendly web based GUI. Help can be initiated from the GCS by sending required number of personnel to the location acquired through LoRa.

Chapter 10

Discussion and Conclusion

The drone being a friction-less system (other than air resistance), requires a good controller such as PID, neural networks. Utmost care has to be taken for the placement of additional components on the drone, as this might effect the CG. It is advisable to make CG of drone coincide with the geometric centre of the drone.
The project carried out is done as different segments such as computer vision, drone control, and communication. Even though these all segments of work are individually good a performance, the integration carried out is mediocre as the situation being dealt is a disaster management. There is a lot of work which can be done in the autonomous navigation of drone, as a small part is explored in the project i.e., hovering of the drone. A robust integration can be achieved with more emphasis on communication and autonomous navigation.
More data can be collected during the flight of drone and can be used as test data to benchmark the computer vision models such as PoseNet, already sampled data collected from Central Workshop can be labeled with 17 pose points and can be trained for better results.
PoseNet used so far fails to estimate when a person is not facing the camera directly, this is a bottleneck faced during the development. This issue can be tackled to develop a robust system.
More rigorous flight tests can be carried out to test the system under different conditions and constrains.
The developed system can not only be used in a disaster management but can be extended its usage to various other surveillance applications such as crowd management, human tracking.

References

[1] Meera, Ajith Anil, et al. "Obstacle-aware adaptive informative path planning for uav-based target search." 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019.

[2] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[3] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767 (2018).

[4] Lin, Tsung-Yi, et al. "Microsoft coco: Common objects in context." European conference on computer vision. Springer, Cham, 2014.

[5] Everingham, Mark, et al. "The pascal visual object classes (voc) challenge." International journal of computer vision 88.2 (2010): 303-338.

[6] Anton A. Zhilenkov1, Ignat R. Epifantsev. "System of Autonomous Navigation of the Drone in Difficult Conditions of the Forest Trails"

[7] India Today article dated August 24, 2018.

[8] Liang Zhao,Charles E. Thorpe."Stereo- and Neural Network-Based Pedestrian Detection"

[9] Kobi Levi and Yair Weiss. "Learning Object Detection from a Small Number of Examples: the Importance of Good Features."

[10] `https://developer.nvidia.com/embedded/jetson-nano-developer-kit`

[11] D. Câmara, "Cavalry to the rescue: Drones fleet to help rescuers operations over disasters scenarios".

[12] Rivas, Alberto, et al. "Detection of cattle using drones and convolutional neural networks." Sensors 18.7 (2018): 2048.

[13] Constantine P.Papageorgiou, Michael Oren, Tomaso Poggio."A General Framework for Object Detection."

[14] C.Ambika bhuvaneswari, M.Muthumari. "Design and realization of radio communication using LoRa & XBee module for an e-Bike."

[15] Sharma V, You I, Pau G, Collotta M, Lim JD, Kim JN, "LoRaWAN-Based Energy-Efficient Surveillance by Drones for Intelligent Transportation Systems."

[16] Aras, Emekcan, et al. "Exploring the security vulnerabilities of LoRa."

[17] Meera, Ajith Anil, "Obstacle-aware Adaptive Informative Path Planning for UAV-based Target Search."

[18] Alex Kendall, Matthew Grimes, Roberto Cipolla, "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization" ; The IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2938-2946

[19] Mikkel Cornelius Nielsen ; Mari Hovem Leonhardsen ; Ingrid Schjølberg, "Evaluation of PoseNet for 6-DOF Underwater Pose Estimation"

[20] N Jlidi, A Snoun, T Bouchrika, "PTLHAR: PoseNet and transfer learning for human activities recognition based on body articulations"

[21] EM Park, IS Kim, SK Jung, "Hand gesture recognition assisted by human pose information"

[22] Umer Rafi, Ilya Kostrikov, Juergen Gall, Bastian Leibe, "An Efficient Convolutional Network for Human Pose Estimation"

[23] Baohua Qiang, Shihao Zhang, Yongsong Zhan, "Improved Convolutional Pose Machines for Human Pose Estimation Using Image Sensor Data"

[24] Pengjie Tang, Hanli Wang, Sam Kwong, "G-MS2F: GoogLeNet based multi-stage feature fusion of deep CNN for scene recognition"

[25] L. Apvrille, T. Tanzi and J. Dugelay, "Autonomous drones for assisting rescue services within the context of natural disasters".

[26] Martín Abadi et al. "TensorFlow: Large-scale machine learning on heterogeneous systems, 2015." Software available from tensorflow.org.

[27] Bondi, Elizabeth, et al. "Spot poachers in action: Augmenting conservation drones with automatic detection in near real time."

[28] Hanhirova, Jussi, et al. "Latency and throughput characterization of convolutional neural networks for mobile computer vision." Proceedings of the 9th ACM Multimedia Systems Conference. ACM, 2018.

[29] Martín Abadi et al. "TensorFlow: Large-scale machine learning on heterogeneous systems", 2015. Software available from `tensorflow.org`.

[30] Scott, Judy, and Carlton Scott. "Drone delivery models for healthcare." Proceedings of the 50th Hawaii international conference on system sciences. 2017.

[31] World Focus Magazine `http://www.worldfocus.in/magazine/disaster-management-in-india/`

[32] Comptroller and Auditor General of India Report 5 of year 2013.

[33] India Today article dated August 24, 2018. *https://www.indiatoday.in/india/kerala/story/death-toll-in-kerala-floods-rises-to-417-36-people-still-missing-1322764-2018-08-24*

[34] Observer Research Foundation article dated 10 September 2018 *https://www.orfonline.org/expert-speak/43901-improved-disaster-management-saves-kerala-despite-lack-of-preparedness/*

[35] eCalc - xcopterCalc - the most reliable Multicopter Calculator on the Web `https://ecalc.ch/xcoptercalc.php`