# Real Time Social Distancing Violation Detection System


Project Team

| | |
|---|---|
| Syed Ali Hashim | 18P-0148 |
| Salman Arif | 18P-0075 |


Session 2018-2022


Supervised by

Mr. Muhammad Amin

**Department of Computer Science**

**National University of Computer and Emerging Sciences**
**Peshawar, Pakistan**


**June, 2022**

# Student’s Declaration

We declare that this project titled “*Real Time Social Distancing Violation Detection System*”, submitted as requirement for the award of degree of Bachelors in Computer Science, does not contain any material previously submitted for a degree in any university; and that to the best of our knowledge, it does not contain any materials previously published or written by another person except where due reference is made in the text.

We understand that the management of Department of Computer Science, National University of Computer and Emerging Sciences, has a zero tolerance policy towards plagiarism. Therefore, We, as authors of the above-mentioned thesis, solemnly declare that no portion of our thesis has been plagiarized and any material used in the thesis from other sources is properly referenced.

We further understand that if we are found guilty of any form of plagiarism in the thesis work even after graduation, the University reserves the right to revoke our BS degree.

Syed Ali Hashim Signature: ____________________

Salman Arif Signature: ____________________

____________________________________

Verified by Plagiarism Cell Officer
Dated:

# Certificate of Approval

The Department of Computer Science, National University of Computer and Emerging Sciences, accepts this thesis titled *Real Time Social Distancing Violation Detection System*, submitted by Syed Ali Hashim (18P-0148), and Salman Arif (18P-0075), in its current form, and it is satisfying the dissertation requirements for the award of Bachelors Degree in Computer Science.

**Supervisor**

Mr. Muhammad Amin Signature: ________________

______________________________

Zeeshan Khan

FYP Coordinator
National University of Computer and Emerging Sciences, Peshawar

______________________________

Dr. Hafeez ur Rehman

HoD of Department of Computer Science
National University of Computer and Emerging Sciences

# Acknowledgements

We are very thankful to Almighty Allah for giving us a lot of courage to choose this project and for progressing in it with great dignity and grace. All the group members played a significant role in the progress of this project. We would like to grab the opportunity to thank Sir Muhammad Amin who motivated us to a selection of this project. He encouraged us and showed us the way the project can be implemented. He has been kind to us since the beginning of our project. Despite his very busy schedule, he was always available to us and gave us time whenever we wanted to meet him. His guidance and experience proved very helpful to us to make progress in our project. We are highly thankful to all of our teachers especially Dr. Mohammad Nouman and Sir Muhammad Amin who had been guiding us throughout our project work. Their knowledge, guidance, and training enabled us to carry out this development work more efficiently. We would like to thank all the FYP panel for giving us useful suggestions about increasing the quality of our project. Their suggestions and motivation helped us a great deal.

Syed Ali Hashim

Salman Arif

# Abstract

Due to COVID-19, The need of social distancing is mandatory to control the spread of the disease. People are advised to limit their interactions with other people as much as they can to reduce the chance of spreading the covid-19 virus with physical or close contact. In this system we will see how python, computer vision and deep learning model can be used to monitor social distancing at public places and workplaces or education institutes. To make sure the people follow social distancing protocols in crowded places, A real time social distancing detection software that can monitor if people are following the social distancing protocols by taking real time video streams from the camera. Social distancing violation detection offers an opportunity for real-time measuring and analysing the physical distance between pedestrians using real time videos in public spaces. Our solution focuses on detecting people through real time camera feeds and detect the social distancing violations by calculating the distance between people in real time. Our system will also give real time results by pointing out violators in real time and forming red bounding boxes around them. A violation threshold is defined and when the number of violations are exceeded from that 180cm threshold then yellow bounding box will appear on that person and if two persons further get closer to each other then red bounding box will appear.
**Keywords:** Social distancing surveillance, Deep Learning, Computer Vision, CCTV,Camera Calibration, Person Detection, Pedestrian Tracking

# Contents


**1 Preliminaries and Introduction** **1**

1.1 Introduction . . . . . . . . . . 1

1.2 Background . . . . . . . . . . 2

1.3 Problem Statement . . . . . . . . . . 2

1.4 Objective . . . . . . . . . . 3

1.5 Project Domain . . . . . . . . . . 3

**2 Review of Literature** **5**

2.1 Introduction . . . . . . . . . . 5

2.2 Techniques Used . . . . . . . . . . 5

2.2.1 Faster-RCNN . . . . . . . . . . 5

2.2.2 You Only Look Once(YOLO) . . . . . . . . . . 6

2.2.3 Solid State Detector(SSD) . . . . . . . . . . 6

2.2.4 Spatio-temporal-based Trajectory Approach . . . . . . . . . . 6

2.2.5 Solid State Detector(SSD) with mobilenetv2 . . . . . . . . . . 6

2.2.6 Haar Cascade based eye and face detector . . . . . . . . . . 7

2.2.7 Distance measurement method based on perspective transformation 7

2.2.8 Mobilenetv2 with transfer learning . . . . . . . . . . 7

2.2.9 3D object detection from bird eye view . . . . . . . . . . 8

2.3 Analysis of Research Papers . . . . . . . . . . 8

**3 System Analysis and Design** **9**

3.1 Introduction . . . . . . . . . . 9

3.2 Use Case Diagram . . . . . . . . . . 9

3.3 Flow Chart . . . . . . . . . . 11

3.4 Gantt Chart . . . . . . . . . . 11
3.5 Activity Diagram . . . . . . . . . . 12
3.6 Class Diagram . . . . . . . . . . 13

**4 Proposed Methodology** **15**
4.1 Introduction . . . . . . . . . . 15
4.1.1 Project Phases . . . . . . . . . . 15
4.2 YOLOv4 Object Detection Model . . . . . . . . . . 16
4.2.1 Function to detect pedestrians . . . . . . . . . . 17
4.3 Perspective Transformation . . . . . . . . . . 18
4.3.1 Function for transforming Region of Interest(ROI) . . . . . . . . . . 19
4.3.2 Function for transforming centroids . . . . . . . . . . 19
4.4 Bird-eye View Generation . . . . . . . . . . 19
4.5 Distance Estimation . . . . . . . . . . 20
4.5.1 Function for calculating distance . . . . . . . . . . 20

**5 Results and Analysis** **21**
5.1 Introduction . . . . . . . . . . 21
5.2 Detection Results . . . . . . . . . . 22
5.3 Perspective Transformation . . . . . . . . . . 22
5.4 Final social distancing and bird eye view . . . . . . . . . . 23
5.5 Model Evaluation . . . . . . . . . . 24
5.5.1 Precision and Recall . . . . . . . . . . 24
5.5.2 Confusion matrix . . . . . . . . . . 25
5.5.2.1 Precision . . . . . . . . . . 26
5.5.2.2 Recall . . . . . . . . . . 26

**6 Problem Discussion** **29**
6.1 Distance from 2D image . . . . . . . . . . 29
6.2 Problem in perspective transformation . . . . . . . . . . 29
6.3 Camera panning problem . . . . . . . . . . 30

**7 Conclusions and Future Work** **31**

7.1 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31
7.2 Future Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31

**References** **35**

# List of Figures


3.1 Use Case Diagram . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
3.2 Flow Chart Diagram . . . . . . . . . . . . . . . . . . . . . . . . . . 11
3.3 Gantt Chart . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
3.4 Activity Diagram . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
3.5 Class Diagram . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

4.1 Architecture of YOLOv4 [12] . . . . . . . . . . . . . . . . . . . . . . 16
4.2 Pixel manipulation in perspective transformation . . . . . . . . . . . . . 18

5.1 The figure represents the detected people in the frame. along with the probability of that object being a person. . . . . . . . . . . . . . . . . . . 22
5.2 This figure represents the perspective transformed view of the original frame in figure 5.1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
5.3 Violation count after performing social distancing analysis . . . . . . . . 23
5.4 This graph represents precision-recall curve of our model . . . . . . . . . 24
5.5 This graph represents best precision-recall values of our model . . . . . . 25
5.6 Confusion matrix of our model . . . . . . . . . . . . . . . . . . . . . . 27

# List of Tables

2.1 Analysis of Research Papers. . . . . . . . . . . . . . . . . . . . . . . . . . 8

# Chapter 1

# Preliminaries and Introduction

## 1.1 Introduction

The spread of COVID-19 virus and the large scale lock downs across the globe has given rise to an alarming situation. While there is an urgent need to resume operations at the work and education sectors, the safety of people working there cannot be compromised. It has been recommended by the World Health Organization that keeping a distance of around 2 meters between yourself and another person reduces the spread of COVID-19. One of the non-pharmaceutical solution that can reduce the transmission of this highly infectious disease is social distancing. The most effective way of monitoring social distancing violations is to do it using technology thus we propose a system that will monitor the social distancing violations using cameras.

Our main goal is to help authorities to maintain a safe social distancing protocols by using an AI based tool to monitor public areas, workplaces, education institutes to analyze and detect any social distancing violations. It will be a real time monitoring system that will use CCTV cameras firstly the video feed will be analyzed to detect the persons in the frame then it will calculate the distance between every person detected in the frame those who are in a safe distance away form others will be assigned green boxes and the people who were found violating the minimum distance mark will be assigned red boxes.

## 1.2 Background

In this section we will see some previous work done on real time detection of humans and estimating distance from CCTV cameras. Some recent work focusing on the detection of humans involving deep learning and computer vision. Localization is performed on the detected objects and then they are classified based on their shape with the help of a deep learning model.The recent advancements in CNN and deep learning based techniques have achieved a better performance in objects recognition. It is very effective in detecting different objects from different inputs sources. The reason for its success is working on large datasets such as ImageNet.

The real-time object detection algorithms which are developed by using models like R-CNN and YOLO. YOLO is a prominent technique in object detection because of its speed and accuracy.

## 1.3 Problem Statement

Measuring object distance from 2D images while the camera view is changing limits the accuracy of the surveillance system.

## 1.4 Objective

Minimize the risk of infection by monitoring the social distancing violations using bird eye view,which will be very effective measure of calculating distance between persons.

## 1.5 Project Domain

Effectively control the spread of COVID-19 in a public or workplace.

# Chapter 2

# Review of Literature

## 2.1 Introduction

In this chapter some of the existing techniques will be discussed which will give more insights and details of this project.Some state of the state of the art object detectors are used to detect pedestrians from videos.Faster-RCNN based objector.[2],YOLOv4[5],SSD[9] and some of the techniques which includes work on perspective transformation,3D to 2D mapping and recognition,distance calculation are [10] [7]

## 2.2 Techniques Used

### 2.2.1 Faster-RCNN

Faster R-CNN is a deep CNN model which is used for object detection and it appears to the user as a unified network. It predicts the objects based on their locations.It consists of two modules. First module works on proposing regions and the task of second module is to detect objects from that particular region. It has been used to detect persons from a live camera feed.The limitation of proposed technique in these papers[2],[11] is that it was working only on overhead view which are not very common so it reduces the usability of system.

### 2.2.2 You Only Look Once(YOLO)

YOLO[19] is the most widely used detection model in the research papers that we have came across till now .[14] Their model was pre-trained on coco data set but some of the occlusions were detected in detections of persons which makes it quite inefficient in crowded places and also there is no alarm system to notify the user of the violations of risky area.

### 2.2.3 Solid State Detector(SSD)

SSD object detector was used in this approach using mobilenetv2 as backbone.The SSD object detector works by extracting feature from generated maps and then it applies filters based on convolutional network to detect objects.[4]This study used SSD as their model but Their model was trained on limited dataset which results in limited accuracy of model and no alarming or notification system is setup to avoid social distancing.

### 2.2.4 Spatio-temporal-based Trajectory Approach

They used Faster Rcnn as their object detection model and they proposed and spatio-temporal based approach to detect person[15].This technique is quite efficient in terms of accuracy but becomes computationally costly in terms of use as a system.They proposed an online system which is not common and expensive for commercial use.

### 2.2.5 Solid State Detector(SSD) with mobilenetv2

[1]This study proposed SSD object detector with mobilenetv2 as backbone for person detection.Their model was performing well on low crowded area but it was not performing well and giving missed detections on densely populated places with accuracy of 60 percent which was quite low.It was also observed that their proposed Region of interest was small in which only few people can adjust which makes it quite inefficient in large organizations.

### 2.2.6 Haar Cascade based eye and face detector

This study[3] focuses on face and eye detection mechanism to classify incoming object as human.They used social distancing tags which are piece of hardwares integrated with remote system and fixed on users who want to monitor social distancing.Haar cascade classifier is used as face and eye detector which has got low accuracy of 80.61 percent because of limited dataset they used.The system is costly because of those SD tags and it also becomes inapplicable in large crowded areas.

### 2.2.7 Distance measurement method based on perspective transformation

This study [6] proposes a measurement method which is the recognition of 2D images and inverse perspective transformation.In this method they are performing the image greying process and then extracting the eigenvalues of measured object. The conversion of distance coordinates between the world coordinate system, the camera coordinate system and the image coordinate system is realized by the inverse perspective transformation method. Prior knowledge base is used as measure for generating the required information.

### 2.2.8 Mobilenetv2 with transfer learning

This paper [8] uses mobilenetv2 as their model for detection of persons on crowded workplaces with additional training on masks dataset for checking whether violator is wearing mask or not.Distance measurement is done using euclidean distance formula on the persons's centroid points returned by model.Their system's limitations are the missed detections in crowded place, masks were not properly identified and distance calculation which is affected by poor caliberation of ROI.

### 2.2.9 3D object detection from bird eye view

This system [7] proposed their own model BEVDet, which shares a similar framework with the up-to-date bird eye view segmentation algorithms.It is modularly designed with an image-view encoder for encoding features in image view,a view transformer for transforming the feature from image view into bird eye view, a bird eye view encoder for further encoding features in the bird eye view perspective and a task- specific head for performing 3D object detection in the BEV space. By taking advantage of this modular design, we can reuse a mass of existing works which have been proved effective in other areas but still expect a long way to upgrade this task of 3D object detection

## 2.3 Analysis of Research Papers

Table 2.1: Analysis of Research Papers.

| Ref No. | Model | Accuracy | Limitation |
|---|---|---|---|
| [2] | Faster R-CNN | 96% | Only works on over head views |
| [14] | YOLO | 89% | Limited dataset |
| [4] | SSD and Mobilenet v2 | 60% | Low Accuracy |
| [15] | Spatio-temporal-based approach | 94% | Computationally costly |
| [1] | SocialdistancingNet-19 | 93% | Could not detect still objects |
| [3] | Haar Cascade based Approach | 80.61% | Costly to install |
| End of Table | | | |

# Chapter 3

# System Analysis and Design

## 3.1 Introduction

In this chapter, we will analyze our project in terms of Project management, System analysis, and design. Project management includes different stages such as planning, analysis, design, deployment, and maintenance. In system analysis, we have made different use cases for our project so that we can design our project based on these use cases. A flow chart and activity diagram are given so that a proper workflow should be followed to assemble this project.We will also Discuss our Proposed Methodology in this section as well.

## 3.2 Use Case Diagram

A use case diagram is the primary form of system/software requirements for a new software program underdeveloped.It is helpful in summarizing the detail of modelling.The user will login and will start camera feed after that only persons will be detected from live feed and social distancing will be calculated.Persons violating certain threshold will be warned by alarm and email will be sent to person in charge.

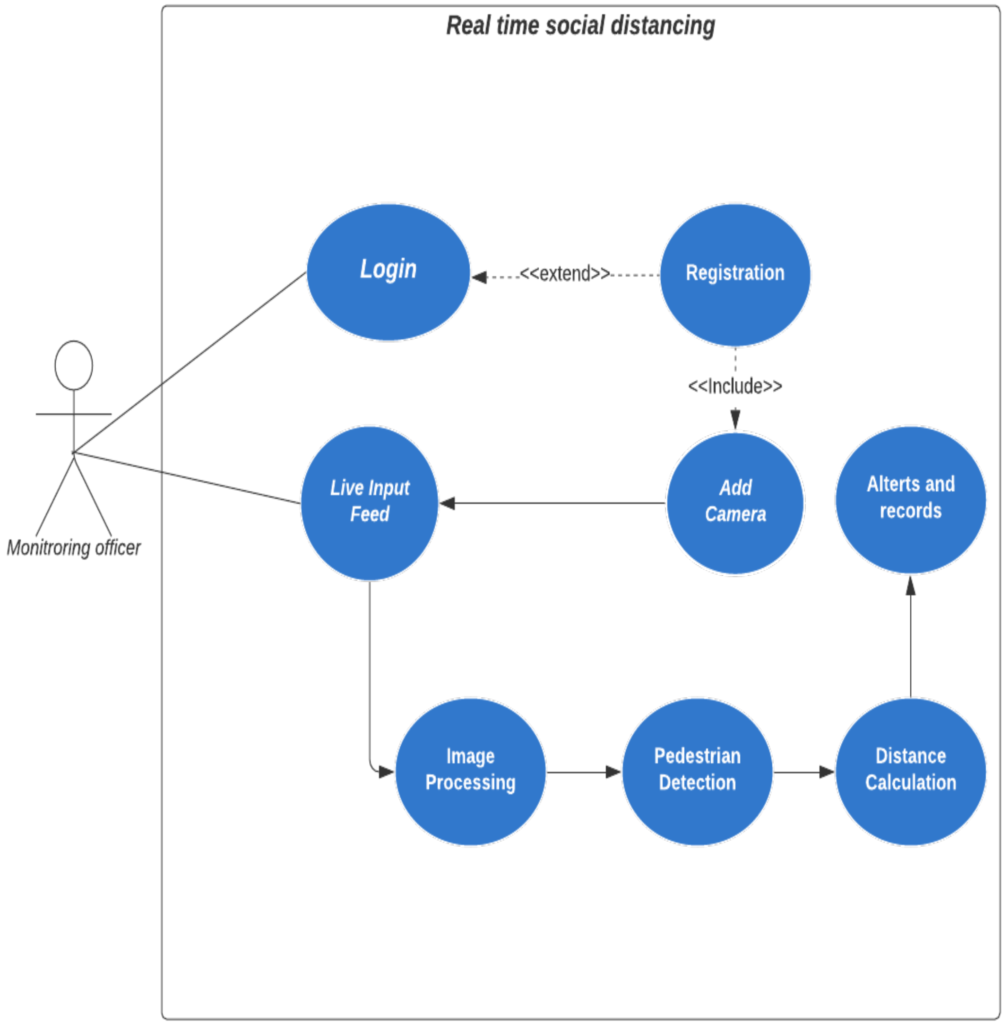


Figure 3.1: Use Case Diagram

## 3.3 Flow Chart

A flow chart is used in designing and documenting the project. This helps us to visualize what is going on in the project.

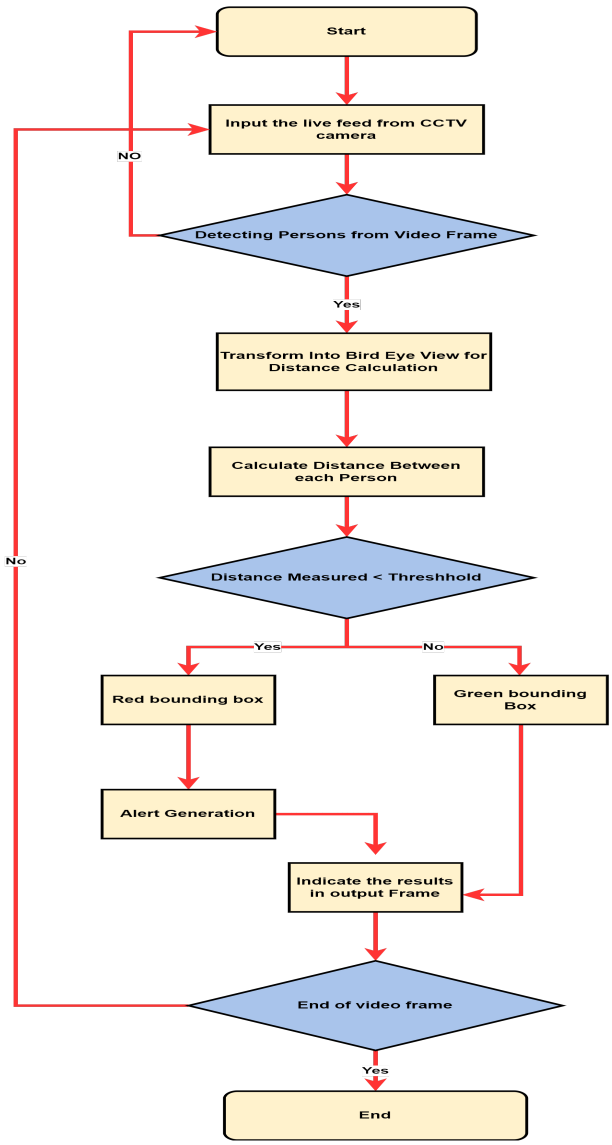


Figure 3.2: Flow Chart Diagram

## 3.4 Gantt Chart

Gantt charts are used here for planning and scheduling the project according to the deliverable time. This helps us to determine that how much time one process should take to complete, and what order should be followed. The deliverable of FYP is shown below in figure 3.3. This explains the same working of the project as described in Flowchart and use case diagram

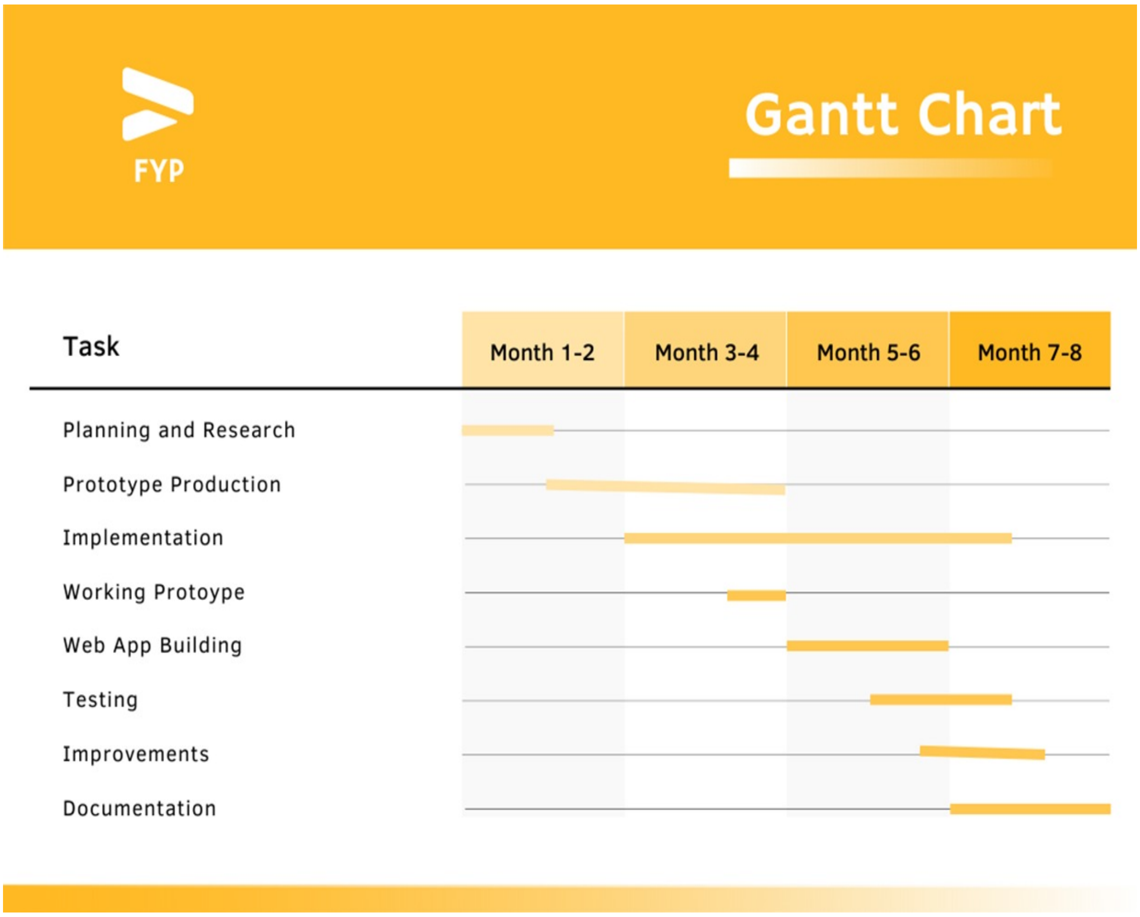


Figure 3.3: Gantt Chart

## 3.5 Activity Diagram

For comprehensive view of our project we use an activity diagram.An activity diagram is a behavioral diagram i.e. it shows the behavior of a system.In the below figure first step is getting video feed from user from which our trained object detector model will detect persons from that feed.After that if persons are detected correctly then distance between them is calculated using euclidean distance formula.If excessive amount of violations are detected then alerts will be send to system and email to person in charge of violations.

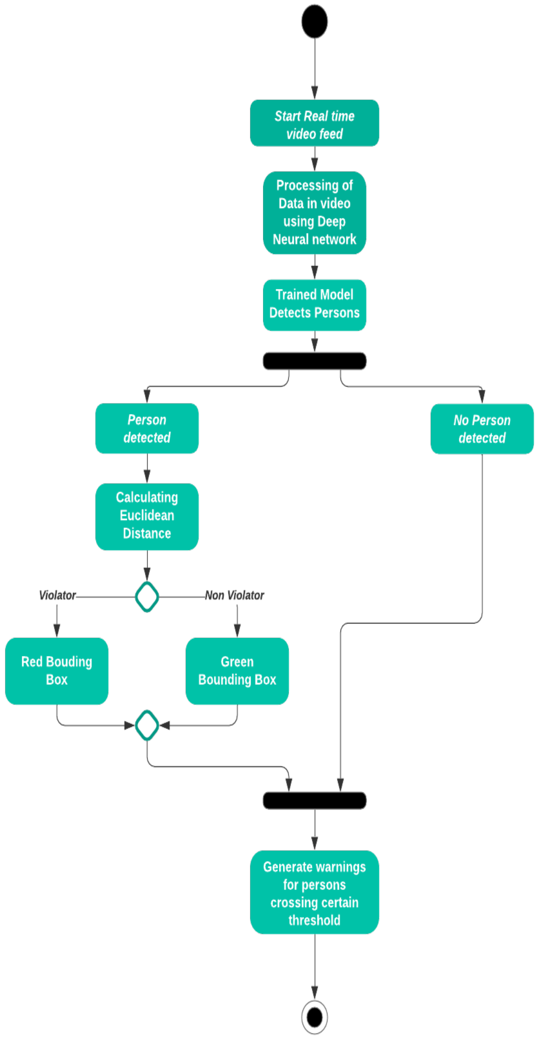


Figure 3.4: Activity Diagram

## 3.6 Class Diagram

For the modeling, we have used a class diagram, which basically describes the static view of our project.In our Case we have three classes named Person detector,Distance calculation and warning generation system.At first our Person detector class contains some methods which will be used for person detection specifically.Distance will be calculated by calculating distance between centroids of detected person from bird eye view.If violations are found then alerts will be generated to avoid social distancing violations.

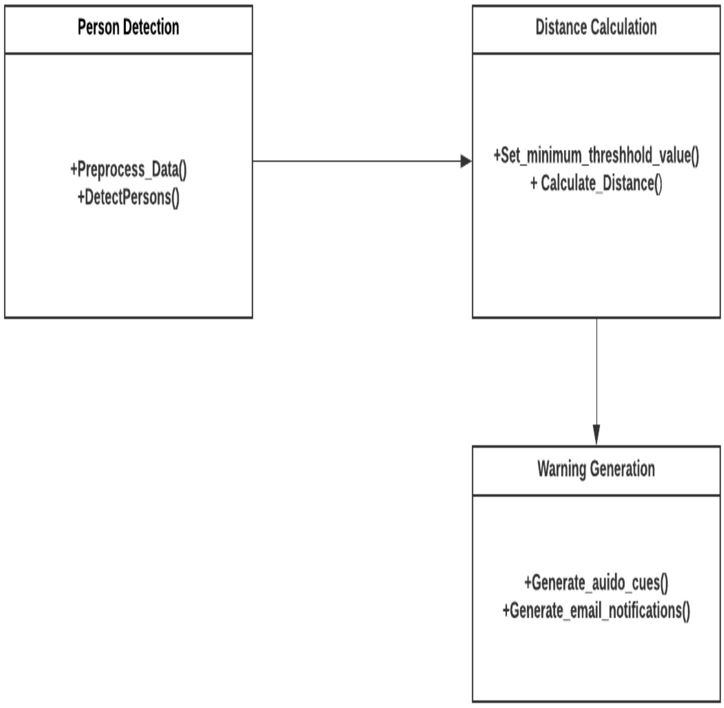


Figure 3.5: Class Diagram

# Chapter 4

# Proposed Methodology

## 4.1 Introduction

In this chapter we will discuss our detailed methodology that will be carried out in our project. First of all dataset of pedestrians will be collected from different datasets hosting sites like google open images and Kaggle.We are using YOLOv4 pre-trained model for pedestrian detection and we are using transfer learning approach to perform additional training for our object detector. We will fine tune our model for better accuracy. After successfully detecting pedestrians from live video feed we will transform our respective region of interest into bird eye view that will assist us in calculation of distance between persons. When persons are found violating minimum distance threshold then violation record of person will be displayed on screen.

### 4.1.1 Project Phases

- Detection
- Perspective Transformation
- Bird-eye Generation
- Distance Estimation

## 4.2 YOLOv4 Object Detection Model

YOLOV4 is real time object detection model that was published in 2020 which achieved state of the art performance on COCO dataset.The main working if YOLOV4 is that it breaks down the object detection tasks into two pieces by using regression to identify object positioning through bounding boxes and classification to identify class of object.YOLOv4 is the most efficient and state of the art deep CNN model that uses several layers to identify and detect objects. it is one of the most optimized model for real time detections[17]. We used YOLO to detected people from the video frame YOLO returns us a list of dimensions of detected objects along with the height and width of the bounding boxed that it makes around the detected objects.
The architecture of YOLOV4 is shown below in figure 4.1

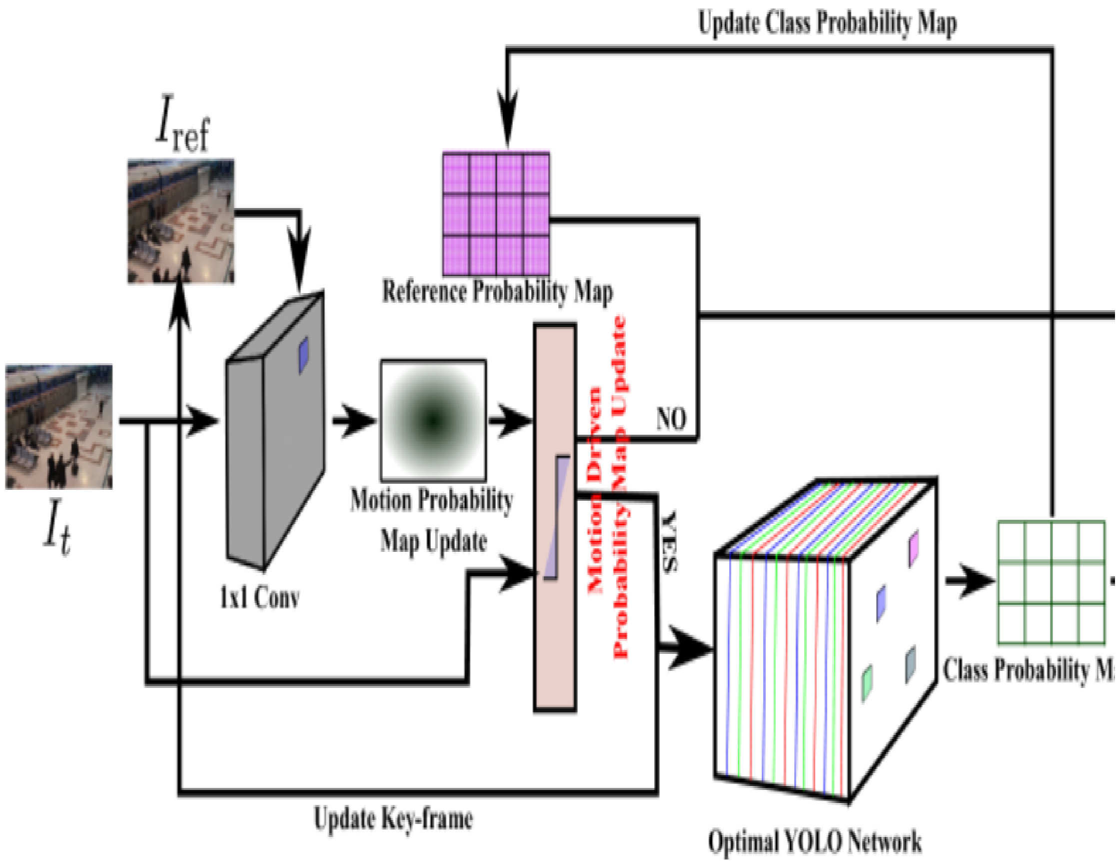


Figure 4.1: Architecture of YOLOv4 [12]

### 4.2.1 Function to detect pedestrians

```python
# Function to detect persons
def detect_pedestrians(frame, net,ln1,H,W):
    blob = cv2.dnn.blobFromImage(frame, 1 / 255.0, (416, 416), swapRB=True, crop=False)
    net.setInput(blob)

    output_layer = net.forward(ln1)
    boxes = []
    confidences = []
    classIDs = []
    #looping over each output layer of yolo
    for output in output_layer:
        for detection in output:
            scores = detection[5:]
            classID = np.argmax(scores)
            confidence = scores[classID]
            # filtering out only persons class in frame
            if classID == 0:

                if confidence > confid:
                    box = detection[0:4] * np.array([W, H, W, H])
                    (centerX, centerY, width, height) = box.astype("int")

                    x = int(centerX - (width / 2))
                    y = int(centerY - (height / 2))

                    boxes.append([x, y, int(width), int(height)])
                    confidences.append(float(confidence))
                    classIDs.append(classID)

    idxs = cv2.dnn.NMSBoxes(boxes, confidences, confid, thresh)
    font = cv2.FONT_HERSHEY_PLAIN
    output_results = []
    for i in range(len(boxes)):
```

```
        if i in idxs:
            output_results.append(boxes[i])
            x, y, w, h = boxes[i]
    return output_results
```

## 4.3 Perspective Transformation

After Detection of people using YOLOv4 the next step was to estimate the distance between the detected objects[18]. For this we used an OpenCV[16],[13] library Perspective Transformation. Calculating the distance from a 2D image in not possible as there is no information about how far or close an object is from the camera and how close or far two objects are from each other if they are standing the same line. To handle this issue we used perspective transformation.In Perspective Transformation, we can change the view of a given image or video to get better information on the required information.In Perspective Transformation[20], we need to give points in the picture where we want to gather information by changing perspective. We also need to give points inside when we want to express our image. Then, we get a point of view from two sets of given points and wrap it up with the original image. It gives us a over head view like we get from a drone so we could accurately estimate the distance between two objects.ROI and detected centroids will be transformed in the end.

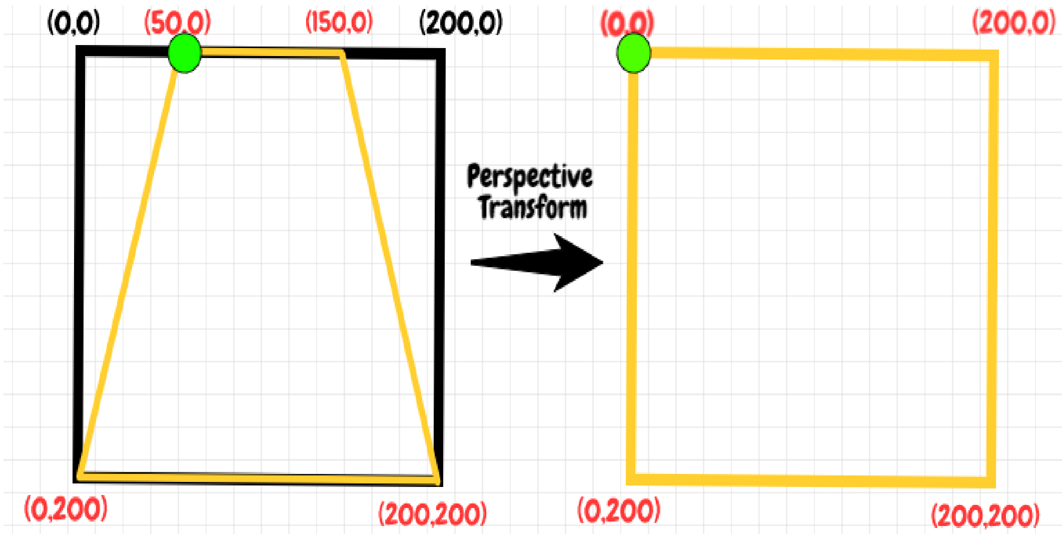


Figure 4.2: Pixel manipulation in perspective transformation

### 4.3.1 Function for transforming Region of Interest(ROI)

```
def Roi_transformation(points):
    source = np.float32(np.array(points[:4]))
    destination = np.float32([[0, H], [W, H], [W, 0], [0, 0]])
    perspective_transform =  cv2.getPerspectiveTransform(source,
   destination)
    return perspective_transform
```

### 4.3.2 Function for transforming centroids

```
transformed_roi = Roi_transformation(points)
output_list = yolo.detect_pedestrians(frame, net,ln1,H,W)
def get_transformed_points(output_list, transformed_roi):
    #centroid points list initialized here
    centroid_points = []
    for box in output_list:
        points = np.array([[[int(output_list[0]+(output_list
   [2]*0.5)),int(output_list[1]+output_list[3])]]] , dtype="float32
   ")
        # here perspective transformation is performed on all
   detected persons
        bd_pnt = cv2.perspectiveTransform(points,
   prespective_transform)[0][0]
        pt = [int(bd_pnt[0]), int(bd_pnt[1])]
        centroid_points.append(pt)
    return centroid_points
```

## 4.4 Bird-eye View Generation

Bird-eye view is an over head drone like view of a picture or a video. After the prospective transformation we have the transformed dimensions of the detected objects now we need to plot those dimensions on a plane. For that we used red and green dots and lines to show violators and people at safe distance from each other.

## 4.5 Distance Estimation

After performing pedestrian detection we are going to calculate the distance between the people using perspective transformation.This is our novel approach of calculating distance as bird eye transform the detected objects on 2D plane from which distance can be easily calculated by using any suitable formula.Once the detected objects are plotted on 2d plane based on their centroids then we used the pythagoras theorem to calculate distance between objects.The pythagoras theorem between two points p1 and p2 is given by

$$d(\mathbf{p1},\mathbf{p2}) = \sqrt{(p1)^2 + (p2)^2}$$

$Eq.4.1 Pythagoras Theorem$

### 4.5.1 Function for calculating distance

Following Pythagoras theorem function is used to calculate distance between person's transformed points:

```
def cal_distance(p1, p2, distance_w, distance_h):
    #Here distance_w and distance_h are the two pixel to cm ratio
   points on image
    h = abs(p2[1]-p1[1])
    w = abs(p2[0]-p1[0])

    dis_w = float((w/distance_w)*180)
    dis_h = float((h/distance_h)*180)

    return int(np.sqrt(((dis_h)**2) + ((dis_w)**2)))
```

# Chapter 5

# Results and Analysis

## 5.1 Introduction

In this chapter we will show the final results produced by our system which includes the detection of people by using yolov4 algorithm, perspective transformation using OpenCV,Bird eye view generation and the distance estimation results.
In the final output we have shown the results of violations on output frame on which safe people will be shown on green boxes,people who are violating the minimum threshold are considered high risk will be shown in red bounding boxes,people who are violating second minimum threshold will be shown in yellow bounding box as low risk people. Our whole violation record which will be shown on output frame are classified as:

- Total Violations
- High Risk People
- Low Risk People
- Safe People

## 5.2 Detection Results

The first phase of our project was to detect the people from each video frame using YOLO and form a bounding box around each detected object.

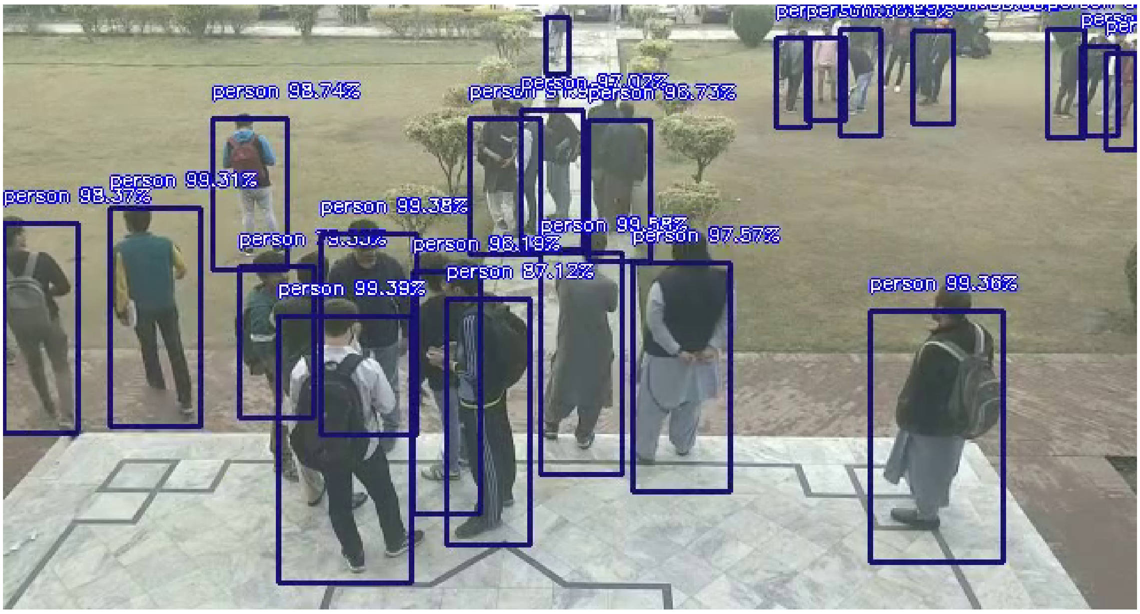


Figure 5.1: The figure represents the detected people in the frame. along with the probability of that object being a person.

## 5.3 Perspective Transformation

In this phase we perform the persepective transformation on the frame using OpenCV which give us the over head view of the frame.This can be done by selecting the roi points on image frame which are treated as source points while destination points are image height and width and by performing perspective transformation roi points will be stretched and hence all objects will be observed as being viewed from top down perspective.

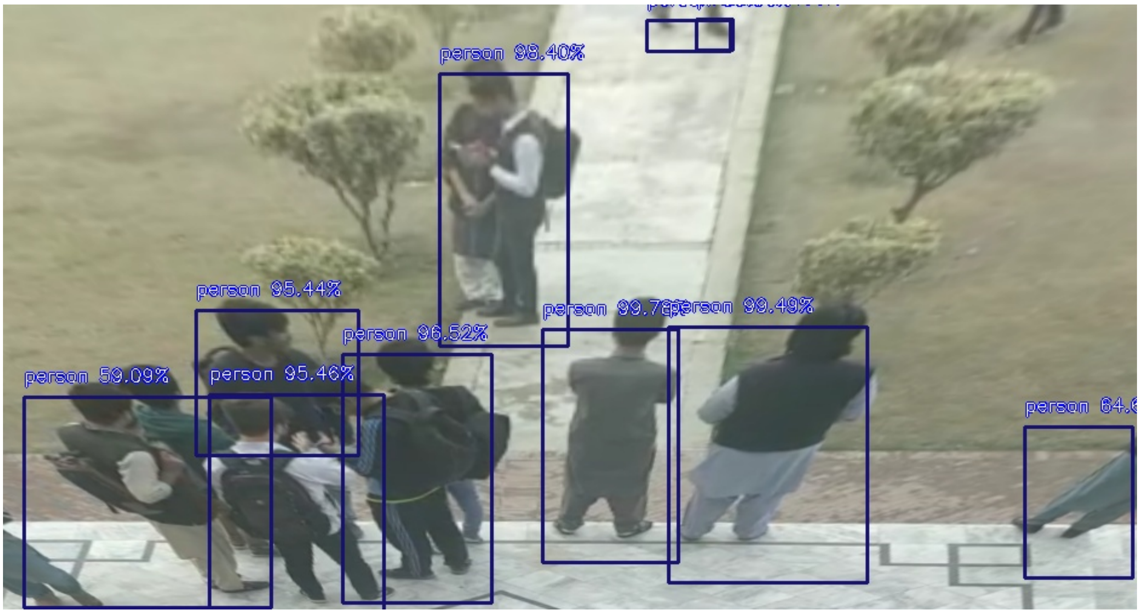


Figure 5.2: This figure represents the perspective transformed view of the original frame in figure 5.1

## 5.4 Final social distancing and bird eye view

Our final social distancing view with its corresponding bird eye view are shown below.Here the output frame contains all the violation record and all the persons detected in region of interest are shown on bird eye view frame as colored dots which are used as points for distance calculation between detected persons.

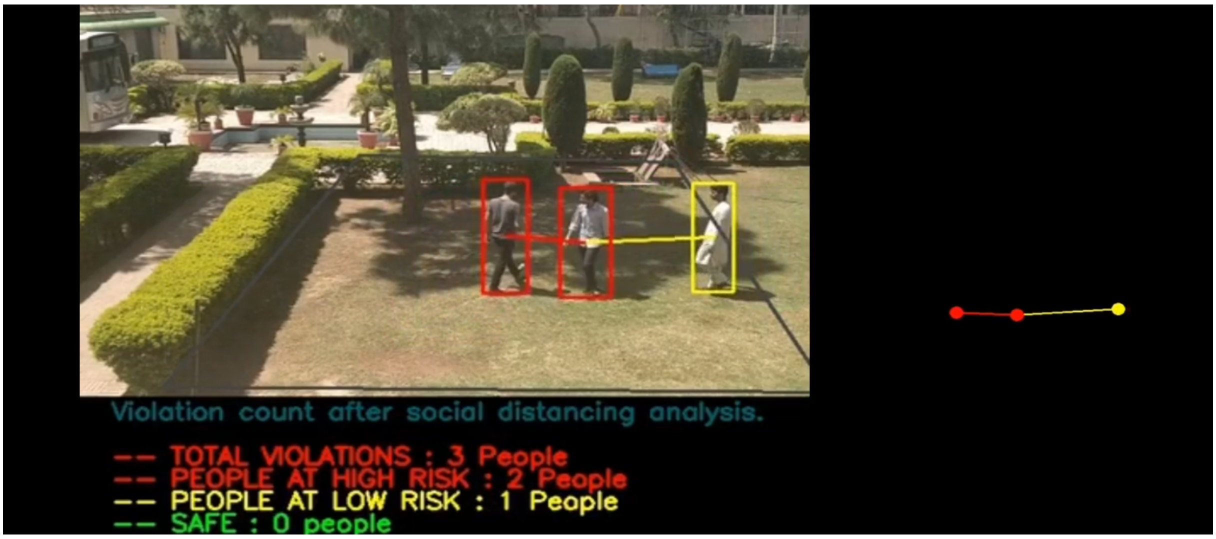


Figure 5.3: Violation count after performing social distancing analysis

## 5.5 Model Evaluation

### 5.5.1 Precision and Recall

We selected YOLOv4 for this project because of its better accuracy and most importantly its capability of detecting objects better in real time which is more important to us as our project works in real time.Here we computed the precision recall curve of our model based on ground truth values and predictions it makes.As our model needs threshold to classify between objects we have generated list of different thresholds and calculated precision and recall values at different threshold values.Increasing precision and recall values show better performance of model.

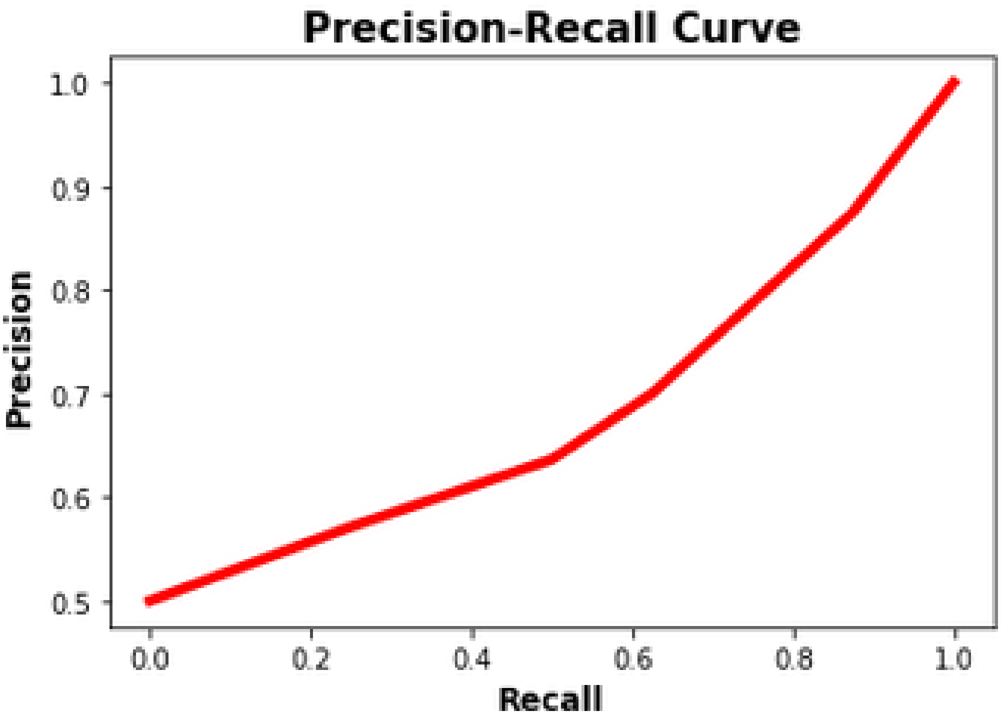


Figure 5.4: This graph represents precision-recall curve of our model

The purpose of this curve is to identify best threshold value to balance precision and recall value. here the blue dot represents the values of precision and recall as 0.87 and 0.83 respectively at threshold value of 0.6

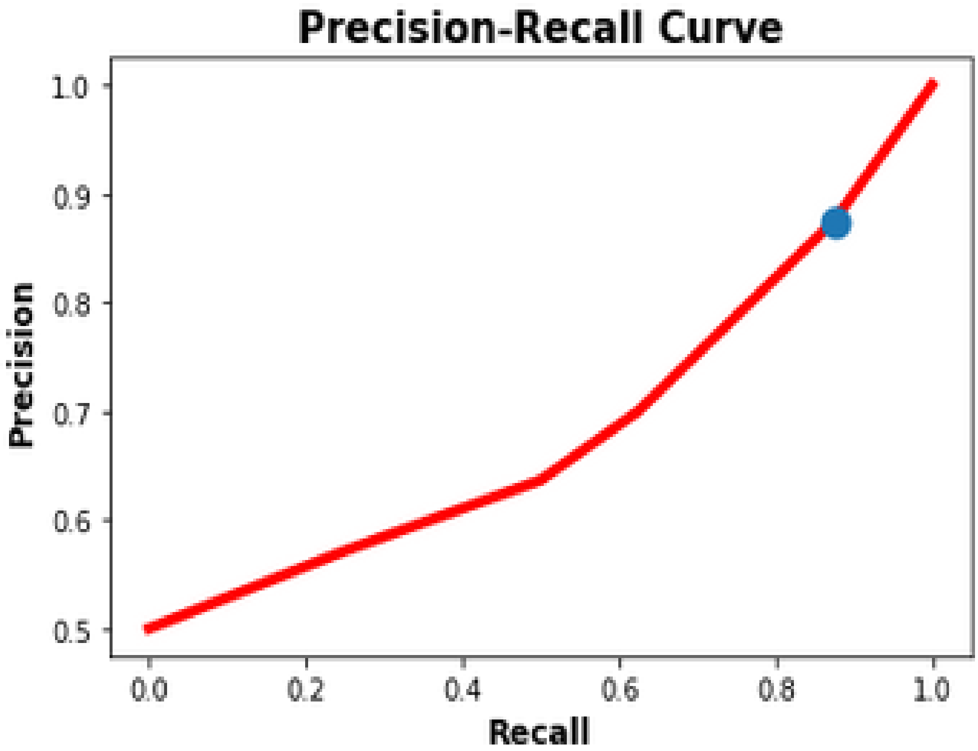


Figure 5.5: This graph represents best precision-recall values of our model

### 5.5.2 Confusion matrix

Object detection models have strengths and weaknesses some have better speed and some have better accuracy we are looking for a bit of both as we have a real time system and we will also be working on CCTV videos which usually do not have a good resolution.We have generated confusion matrix of our model which performs very well regardless of low resolution conditions. We have calculated precision and recall of our system by using the formulas given below respectively:

#### 5.5.2.1 Precision

Precision is defined as the number of true positives over the number of true positives plus the number of false positives, basically it is the ratio between the True Positives and all the Positives.

$$Precision = \frac{TP}{TP + FP}$$

$$Eq.4.2 Precision Formula$$

#### 5.5.2.2 Recall

Recall is defined as the number of true positives over the number of true positives plus the number of false negatives.Recall is the measure of our model correctly identifying True Positives.

$$Recall = \frac{TP}{TP + FN}$$

$$Eq.4.3 Recall Formula$$

Our precision and recall calculated from this confusion matrix is 0.8 and 0.6 respectively.

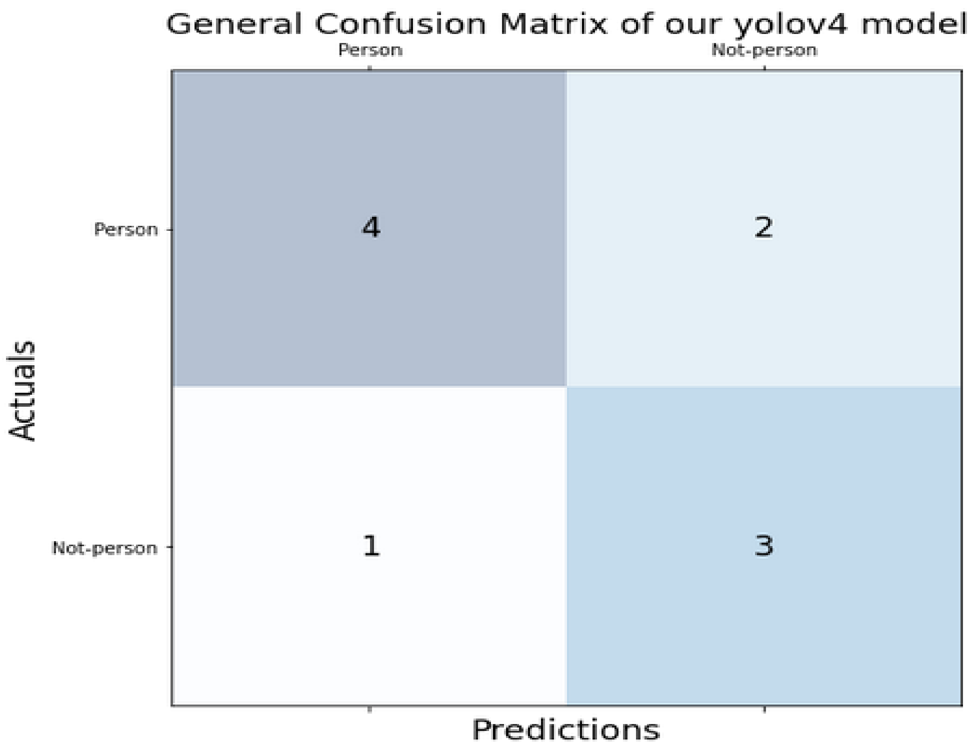


Figure 5.6: Confusion matrix of our model

# Chapter 6

# Problem Discussion

## 6.1 Distance from 2D image

The problems we faced during this project was calculating distance from a 2D image. In a 2D image there is no information about depth so we can not differentiate how much close or far an object is from the camera. For that we did perspective transformation that gave us an over head view which normalized the depth factor.

## 6.2 Problem in perspective transformation

The other problem that that aroused next was if we detect the people in the frame first and then do the perspective transformation then in the transformed frame the position of those people will most likely be changed so the dimensions of detected people given by YOLO would not be accurate. And we could not perform perspective transformation first and then detect people from that frame because it would effect YOLO accuracy because in the perspective transformed frame the image is stretched so miss detections are possible. To address this problem we performed the detection first and then the perspective transformation and then the dimensions we received from YOLO we did perspective transformation of those points as well to match it with the perspective transformed output frame results.

## 6.3 Camera panning problem

Camera panning problem is that when we move the camera from left to right or vice versa the whole frame of the camera is disturbed which also disturbs the ROI(region of interest). To solve that we made the ROI (region of interest) static so it would move with the camera angle.

# Chapter 7

# Conclusions and Future Work

## 7.1 Conclusion

This is a real time surveillance system which detects any social distancing violation and shows results in real time. It can be used in public spaces to ensure safety protocols. Furthermore it can be implemented on a normal CCTV camera feed it does not require any hardware which makes it cost efficient to implement.

## 7.2 Future Work

Future experiments to improve our approach include re-training the model with video feeds from the deployment region as well as testing the performance with newer detection algorithms. Same person tracking in different camera at the same time can be resolved. In addition to that auto selection of ROI (Region of Interest) can be done using computer vison.

# Bibliography


[1] Afiq Harith Ahamad, Norliza Zaini, and Mohd Fuad Abdul Latip. Person detection for social distancing and safety violation alert based on segmented roi. In *2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pages 113–118, 2020.

[2] Imran Ahmed, Misbah Ahmad, and Gwanggil Jeon. Social distance monitoring framework using deep learning architecture to control infection transmission of covid-19 pandemic. *Sustainable Cities and Society*, 69:102777, 2021.

[3] Tareq Alhmiedat and Majed Aborokbah. Social distance monitoring approach using wearable smart tags. *Electronics*, 10(19):2435, 2021.

[4] N Anithadevi, J Abinisha, V Akalya, and V Haripriya. An improved ssd object detection algorithm for safe social distancing and face mask detection in public areas through intelligent video analytics. In *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–7, 2021.

[5] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[6] Xingwen Gao, Kangning Zeng, Ming Hu, Deming Zhao, and Mei Wu. Distance measurement method based on two-dimensional image recognition and inverse perspective transformation. pages 604–607, 2021.

[7] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *CoRR*, abs/2112.11790, 2021.

[8] Prateek Khandelwal, Anuj Khandelwal, Snigdha Agarwal, Deep Thomas, Naveen Xavier, and Arun Raghuraman. Using computer vision to enhance safety of workforce in manufacturing in a post covid world. *arXiv preprint arXiv:2005.05287*, 2020.

[9] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[10] David G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987.

[11] Sergio Saponara, Abdussalam Elhanashi, and Alessio Gagliardi. Implementing a real-time, ai-based, people detection and social distancing measuring system for covid-19. *Journal of Real-Time Image Processing*, 18(6):1937–1947, 2021.

[12] Mohammad Javad Shafiee, Brendan Chywl, Francis Li, and Alexander Wong. Fast yolo: A fast you only look once system for real-time embedded object detection in video. *arXiv preprint arXiv:1709.05943*, 2017.

[13] Juhi Shah, Mahavir Chandaliya, Harsh Bhuta, and Pratik Kanani. Social distancing detection using computer vision. In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pages 1359–1365. IEEE, 2021.

[14] G V Shalini, M Kavitha Margret, M J Sufiya Niraimathi, and S Subashree. Social distancing analyzer using computer vision and deep learning. *Journal of Physics: Conference Series*, 1916(1):012039, may 2021.

[15] Jie Su, Xiaohai He, Linbo Qing, Tong Niu, Yongqiang Cheng, and Yonghong Peng. A novel social distancing analysis in urban public space: A new online spatio-temporal trajectory approach. *Sustainable Cities and Society*, 68:102765, 2021.

[16] Ajay Talele, Aseem Patil, and Bhushan Barse. Detection of real time objects using tensorflow and opencv. *Asian Journal For Convergence In Technology (AJCT) ISSN-2350-1146*, 2019.

[17] Ting-En Tseng, An-Sheng Liu, Po-Hao Hsiao, Cheng-Ming Huang, and Li-Chen Fu. Real-time people detection and tracking for indoor surveillance using multiple top-view depth cameras. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4077–4082. IEEE, 2014.

[18] Shane Tuohy, Diarmaid O'Cualain, Edward Jones, and Martin Glavin. Distance determination for an automobile environment using inverse perspective mapping in opencv. 2010.

[19] Jimin Yu and Wei Zhang. Face mask wearing detection algorithm based on improved yolo-v4. *Sensors*, 21(9):3263, 2021.

[20] Junshu Zhang, Jindong Zhang, Botao Chen, Jian Gao, Shanwei Ji, Xiaolong Zhang, and Zhaodan Wang. A perspective transformation method based on computer vision. In *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pages 765–768. IEEE, 2020.