Social Distance Detector

Submitted in partial fulfillment of the requirements
of the degree of

B. E. Computer Engineering

By

Dhruvi Kakadiya	**172032**	**62**
Ritik Lodha	**172088**	**63**
Nehal Panara	**172072**	**17**
Onkar Pednekar	**172075**	**78**

Supervisor(s):

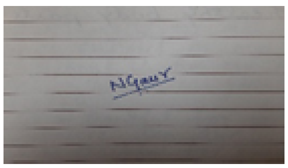
Name of the Supervisor
Nidhi Gaur
Assistant Professor

Department of Computer Engineering
St. Francis Institute of Technology
(Engineering College)
University of Mumbai

2020-21

CERTIFICATE

This is to certify that the project entitled **“Social Distance Detector”** is a bonafide work of **“Nehal Panara (17), Dhruvi Kakadiya (62), Ritik Lodha (63), Onkar Pednekar (78)”** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of B.E. in Computer Engineering.



(Ms. Nidhi Gaur)
Supervisor/Guide

Dr. Kavita Sonwane

Head of Department

Dr. Sincy George

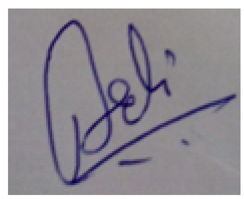
Principal

Project Report Approval for B.E.

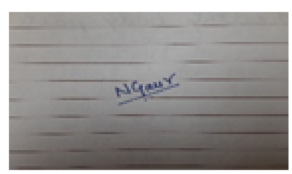
This project report entitled (***Social Distance Detector***) by (**Onkar pednekar(78), Nehal panara(17), Dhruvi Kakadiya (62), Ritik lodha(63**) is approved for the degree of ***B.E. in Computer Engineering.***

Examiners

Dr. Arti Deshpande



Ms. Nidhi Gaur

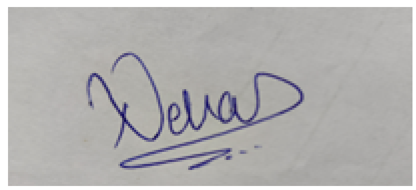


Date: 29/05/2021

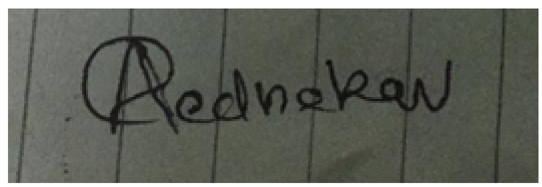
Place: Mumbai

Declaration

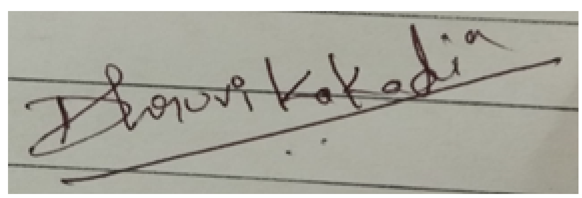
I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.



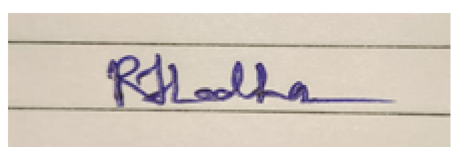
(Nehal R Panara Roll No.17)



(Onkar Pednekar Roll No.78)



(Dhruvi Kakadia Roll No.62)



(Ritik Lodha Roll No. 63)

Date: 29/05/2021

Abstract

Social distancing has been proven as an effective measure against the spread of the infectious COronaVIrus Disease 2019 (COVID-19). However, individuals are not used to tracking the required 6-feet (2-meters) distance between themselves and their surroundings. An active surveillance system capable of detecting distances between individuals and warning them can slow down the spread of the deadly disease. Furthermore, measuring social density in a region of interest (ROI) and modulating inflow can decrease social distance violation occurrence chance. On the other hand, recording data and labeling individuals who do not follow the measures will breach individuals' rights in free-societies. Here we propose an Python based real-time social distance detection and warning system considering four important ethical factors: (1) the system should never record/cache data, (2) the warnings should not target the individuals, (3) no human supervisor should be in the detection/warning loop, and (4) the code should be open-source and accessible to the public. Against this backdrop, we propose using a web camera and deep learning-based real-time object detectors to measure social distance. If a violation is detected, a non-intrusive audio-visual warning signal is emitted without targeting the individual who breached the social distance measure. Also, if the social density is over a critical value, the system sends a control signal to modulate inflow into the ROI. We tested the proposed method across real-world datasets to measure its generality and performance.

CONTENTS

List of Figures

List of Tables

List of Abbreviations

Sr. No.	Abbreviation	Expanded form
1.	NMS	Non-max suppression
2.	SORT	SINGE ONLINE and REAL TIME tracking technique
3.	CNN	Convolutional neural network
4.	DNN	Deep neural network
5.	YOLO	You Only Look Once
6.	WHO	World Health Organization
7.	RRT-PCR	Real-time reverse transcription-polymerase chain reaction
8.	ROI	Region of Interest

Chapter 1

Introduction

Coronavirus is an infectious disease caused by the coronavirus-2 extreme acute respiratory syndrome. The dis-ease was first detected in Wuhan, China in December,which has contributed to a spread across the world.When in close contact, the virus spreads mainly between individuals, including by tiny droplets formed when sneezing or coughing. Droplets falling on the ground will pass through the air through the body of a human.For the first three days, the infection is the most infectious. Many typical symptoms include nausea, dry cough, and fatigue. Severe and harmful human consequences have contributed to a worldwide halt. Many such signs may include a sore throat and headache. It takes a fortnight for a person with mild symptoms to get healed. The duration of recovery for individuals with severe symptoms depends on the extent,along with an individual's immune capability. The main diagnostic approach is from a nasopharyngeal swab by a real-time reverse transcription-polymerase chain reaction (RRT-PCR). Chest CT imaging is also useful for the diagnosis of people with an elevated probability of infection based on signs and risk factors. Seeing the devastating spread of the disease, the World Health Organization(WHO) suggested favoring the term social distancing.To slow down the rate of spread of the disease it is necessary to maintain physical distance. Maintaining a distance of two meters between two individuals is a must to remain safe and get back to the world we lived a few months back. After the COVID-19 pandemic, the CDC changed the concept of social distancing as keeping out of congregate environments, preventing public meetings, and preserving, when appropriate, a gap of around six feet or two meters from everyone. Recent findings have shown that droplets from a sneeze or a deep breath will be more than six meters during exercise. And hence maintaining the norm of social distancing is a necessity and also in our bent to live a safer and healthier life.

1.1 Description:

Our work proposes to determine whether or not an individual is following the rule of social distance. The findings are verified using both alive streams as well as a video feed. By measuring the gap of two frames of people from the centroids, we can understand whether or not a person is maintaining social distance. Also, they are labelled as safe and unsafe. After detecting the person who is not following the social distance a system based message will be generated and sent to that person.

1.2 Problem Formulation:

Social distancing is arguably the most effective non pharmaceutical way to prevent the spread of a disease. Our goal is to develop a software which can be used to detect distance between a group of people using a webcam or from video . As per the option selected, our system will find the distance between the people and generate a green bounding box if it is permissible, and a red bounding box if social distance is not followed and send the person notifications. To achieve the desired output, we would be implementing various techniques of OpenCV, Computer Vision, and Deep Learning.

1.3 Motivation:

Social Distancing – the term that has taken the world by storm and is transforming the way we live. Social distancing has become a mantra around the world, transcending languages and cultures.

This way of living has been forced upon us by the fastest growing pandemic the world has ever seen – COVID-19. As per the World Health Organization (WHO), COVID-19 has so far infected almost 4 million people and claimed over 230K lives globally. Around 213 countries have been affected so far by the deadly virus.

The biggest cause of concern is that COVID-19 spreads from person to person through contact or if you're within close proximity of an infected person. Given how densely populated some areas are, this has been quite a challenge.

So this got me thinking – I want to build a tool that can potentially detect where each person is in real-time, and return a bounding box that turns red if the distance between two people is dangerously close. This can be used by governments to analyze the movement of people and alert them if the situation turns serious.

1.4 Proposed Solution:

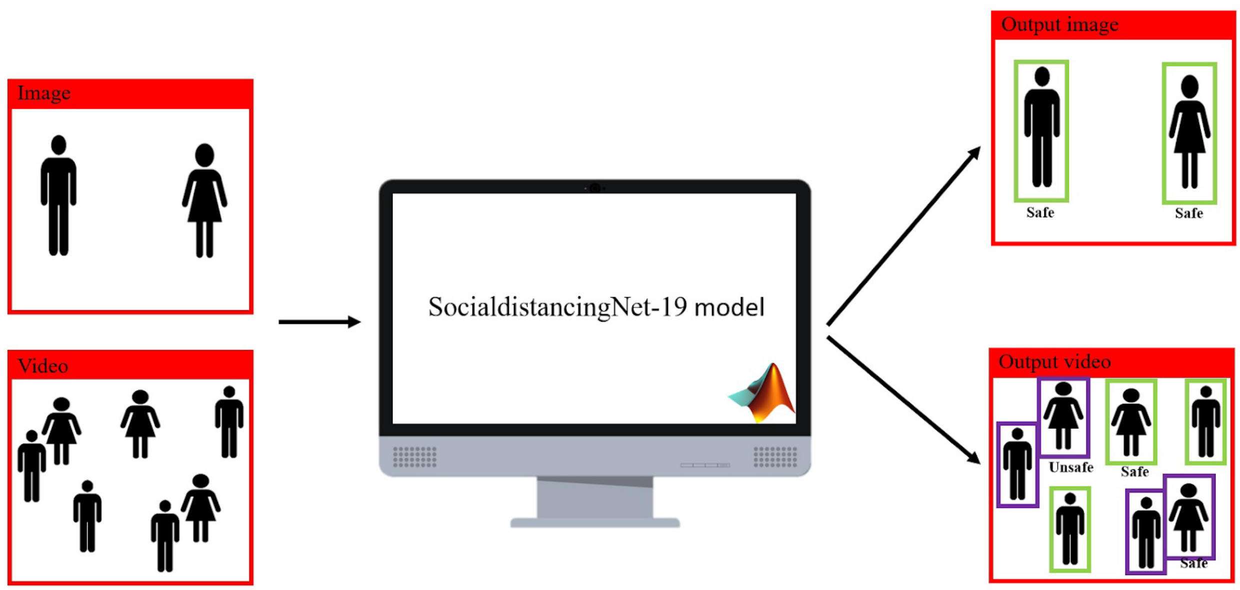


Fig 1.4.1 Social DistancingNet-19 Model

The proposed system focuses on how to identify the person on image/video stream whether the social distancing is maintained or not with the help of computer vision and deep learning algorithm by using the OpenCV, Tensor flow library.

Approach

1. Detects humans in the frame with yolov3.
2. Calculates the distance between every human who is detected in the frame.

3. Shows how many people are at High, Low and Not at risk.

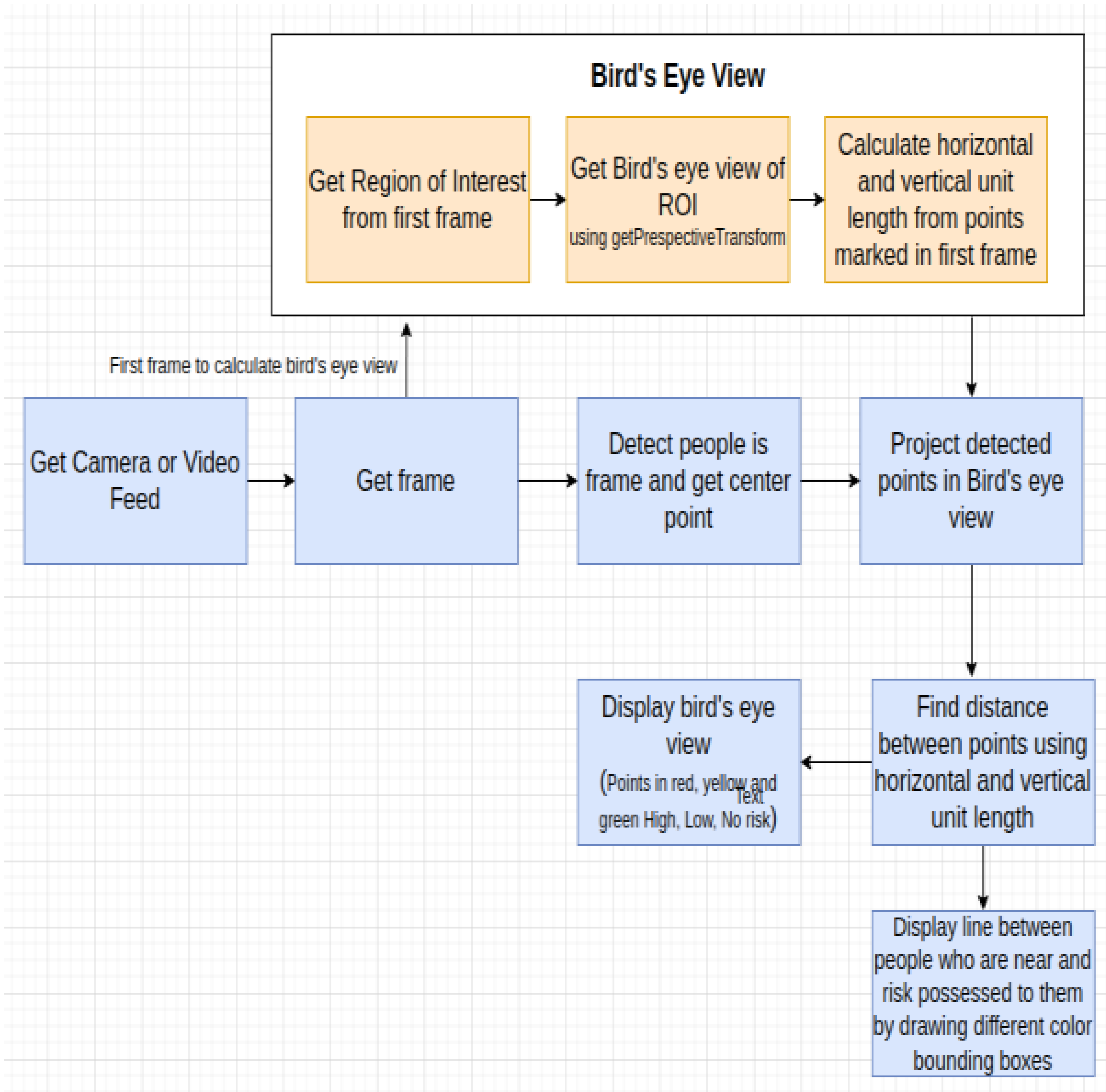


Fig 1.4.2 Flow diagram of social distancing detector model

Camera Perspective Transformation or Camera Calibration

As the input video may be taken from an arbitrary perspective view, the first step is to transform perspective of view to a bird's-eye (top-down) view. As the input frames are monocular (taken from a single camera), the simplest transformation method involves selecting four points in the perspective view which define ROI where we want to monitor social distancing and mapping them to the corners of a rectangle in the bird's-eye view. Also these points should form parallel lines in the real world if seen from above (bird's eye view). This assumes that every person is standing on the same flat ground plane. This top view or bird eye view has the property that points are distributed uniformly horizontally and vertically (scale for horizontal and vertical direction will be different).From this mapping, we can derive a transformation that can be applied to the entire perspective image.

Above image shows how we can select the Region of Interest (ROI) and this is a one time step. We draw 8 points on the first frame using the mouse click event. First four points will define ROI where we want to monitor social distance. Next 3 points will define a 180 cm(unit length) distance in horizontal and vertical direction and those should form parallel lines with ROI. In the above image we can see point 5 and point 6 define 180 cm in real life in horizontal direction and point 5 and point 7 define 180 cm in real life in a vertical direction. As we can see ROI formed by the first 4 points has different length in horizontal and vertical direction, so the number of pixels in 180 cm for horizontal and vertical direction will be different in the rectangle (bird's eye view) formed after transformation.
So from point 5, 6, 7 we are calculating the scale factor in the horizontal and vertical direction of the bird's eye view, e.g. how many pixels correspond to 180 cm in real life.

Detection

The second step is to detect pedestrians and draw a bounding box around each pedestrian. To clean up the output bounding boxes, we apply minimal post-processing such as non-max suppression (NMS) and various rule-based heuristics, so as to minimize the risk of over fitting.

Distance Calculation

Now we have a bounding box for each person in the frame. We need to estimate the person 's location in frame. i.e we can take the bottom center point of the bounding box as a person's location in frame. Then we estimate (x,y) location in the bird's eye view by applying transformation to the bottom center point of each person's bounding box, resulting in their position in the bird's eye view. Last step is to compute the bird's eye view distance between every pair of people and scale the distances by the scaling factor in horizontal and vertical direction estimated from calibration.

Software and libraries required

1. Python Opencv numpy argparse
2. Working

Running the program will give you a frame (first frame) where you need to draw ROI and distance scale. To get ROI and distance scale points from first frame Code to transform perspective to Bird's eye view (Top view) and to calculate horizontal and vertical 180 cm distance in Bird's eye view

ROI and Scale points' selection for the first frame. The second step is to detect pedestrians and draw a bounding box around each pedestrian. To detect humans in video and get bounding box details.

Now we have a bounding box for each person in the frame. We need to estimate the person's location in the frame. i.e we can take the bottom center point of the bounding box as a person 's location in frame. Then we estimate (x,y) location in the bird's eye view by applying transformation to the bottom center point of each person's bounding box, resulting in their position in the bird's eye view. To calculate the bottom center point for all bounding boxes and projecting those points in Bird's eye view. Last step is to compute the bird's eye view distance between every pair of people (Point) and scale the distances by the scaling factor in horizontal and vertical direction estimated from calibration.

1.5 Scope of the project:

Social distancing is deliberately increasing the physical space between people to avoid spreading illness. Staying at least six feet away from other people lessens your chances of contracting COVID-19. We can use OpenCV and YOLO to monitor/analyze whether people are maintaining social distance or not. It could be used for just keeping a track whether people are maintaining social distance.

Social distance detector can be used in:

Hospitals

Hospitals can use social distance detectors to monitor if people are adhering to social distance or not. Using CCTV cameras installed at hospital premises, health authorities can track if visitors, patients or health workers are maintaining a distance or not. If somebody is found violating the distance, alerts will be sent to the concerned authorities to take action.

Offices

The risk of COVID-19 will not be ending soon; therefore, offices after the lockdown can use this app to ensure that social distance is maintained until the risk of COVID-19 does not go away. Offices can add face data of their employees and if anyone is found not following distance can receive an alert or notification.

Schools

Using a social distance detector, schools can manage social distance efficiently. If students or teachers are found violating the distance within school premises, an alert is raised and notification is sent to concerned authorities.

Retail shops

Retail stores can use the social distance detector to ensure that visitors maintain a specific distance from each other. Cameras connected to the app would track the in-store activity and trigger voice alert to aware people.

Chapter 2

Review of Literature

Sr no.	Title of the paper	Author	Source	year
1.	Real time social distancing detector using Social distancing Net-19 deep learning network.	Rinkal Keniya, Ninad Mehendale	IEEE	2020
2.	Social distancing detection system with artificial intelligence using computer vision and deep learning	Vinitha , Valentina	IRJET	Aug 2020
3.	DeepSocial : Social distancing monitoring and infection risk assessment in COVID-19 pandemic.	Mahdi Rezaei, Mohsen Azarmi	IEEE	2020

Table: 2.1

1. Various research work has been done on social distance using different techniques.
Real time social distancing detector using Social distancing Net-19 deep learning network :-

- Detection:-
  YOLO model (You Only Look Once).
  The YOLO detection model is similar to the single stage detector model.
  YOLO has three tuning parameters.
  1. Network input sizes
  2. Anchored Box
  3. Feature Extraction Network.
- Distance Calculation:-
  Compute bounding box coordinates and derive the center of the bounding box.The euclidean distance is calculated between the centers to verify if people are in the social distancing limit.

2. Social distance detection with artificial intelligence using computer vision and deep learning :-
APPROACH:-

- Detects humans in frame with YOLO v3.
- Calculates the distance between every human who is detected in frame.
- Shows how many people are at High ,Low and Not at risk.

DETECTION:-
Draw the bounding box around each pedestrian.they apply minimal post-processing such as non-max suppression(NMS).
DISTANCE CALCULATION:-
Distance is calculated using bird's eye view method.Scale the distance by scaling factor in horizontal and vertical distance estimated from calibration.

3. DEEPSOCIAL : Social distancing monitoring and infection risk assessment in COVID-19 pandemic:-

APPROACH :

- Detection
- Tracking
- Inter distance Estimation

- DETECTION:

  Modern DNN based detectors consist of two section:-

  1. A backbone for extracting features.
  2. Head for predicting classes and location of objects.

  Two stage detectors are used.

  1. Detector extracts a set of object proposals by a selective search.
  2. Resizes them to fixed size before feeding them to the CNN model.

  (Same as R-CNN based detectors.)

- TRACKING:

  SINGE ONLINE and REAL TIME (SORT) tracking technique as framework for Kalman filter along with hungerian optimization technique to track people.

Chapter 3

System Analysis

3.1 Functional Requirements:

Detection

The second step is to detect pedestrians and draw a bounding box around each pedestrian. To clean up the output bounding boxes, we apply minimal post-processing such as non-max suppression (NMS) and various rule-based heuristics, so as to minimize the risk of over fitting.

Distance Calculation

Now we have a bounding box for each person in the frame. We need to estimate the person 's location in the frame. i.e we can take the bottom center point of the bounding box as a person 's location in frame. Then we estimate (x,y) location in the bird's eye view by applying transformation to the bottom center point of each person's bounding box, resulting in their position in the bird's eye view. Last step is to compute the bird's eye view distance between every pair of people and scale the distances by the scaling factor in horizontal and vertical direction estimated from calibration.

Software and libraries required

1. Python Opencv numpy argparse
2. Working

Running the program will give you a frame (first frame) where you need to draw ROI and distance scale. To get ROI and distance scale points from first frame Code to transform perspective to Bird's eye view (Top view) and to calculate horizontal and vertical 180 cm distance in Bird's eye view

ROI and Scale points' selection for the first frame. The second step is to detect pedestrians and draw a bounding box around each pedestrian. To detect humans in video and get bounding box details.

Now we have a bounding box for each person in the frame. We need to estimate the person 's location in the frame. i.e we can take the bottom center point of the bounding box as a person 's location in frame. Then we estimate (x,y) location in the bird's eye view by applying transformation to the bottom center point of each person's bounding box, resulting in their position in the bird's eye view. To calculate the bottom center point for all bounding boxes and projecting those points in Bird's eye view. Last step is to compute the bird's eye view distance between every pair of people (Point) and scale the distances by the scaling factor in horizontal and vertical direction estimated from calibration

.3.2 Non Functional Requirements:

Security requirements

The database contains sensitive information of all the staff. Therefore, optimal security measures must be taken to ensure data is safe from unauthorized users.

Safety requirements

If there is extensive damage to the camera that is used to capture live footage it may cause error in face detection .

3.3 Specific Requirements :

3.3.1 Hardware Requirements :

Components	Specifications
Screen Resolution	1024x768 display 5400 RPM hard disk
System/Processor	Intel® Core ™ i3-2330M or more than CPU @ 2.20 GHz.
Memory	4 GB of RAM
Hard Disk	500 GB of HDD or higher recommended
Supported Arch	X86 and x64

Table 3.3.1

3.3.2 Software Requirements:

Components	**Specifications**
Operating System	Windows 2010 (64-bit)
Supported software	YOLO Spyder Open cv Tensor Flow

Table 3.3.2

3.4 Use-Case Diagrams and description

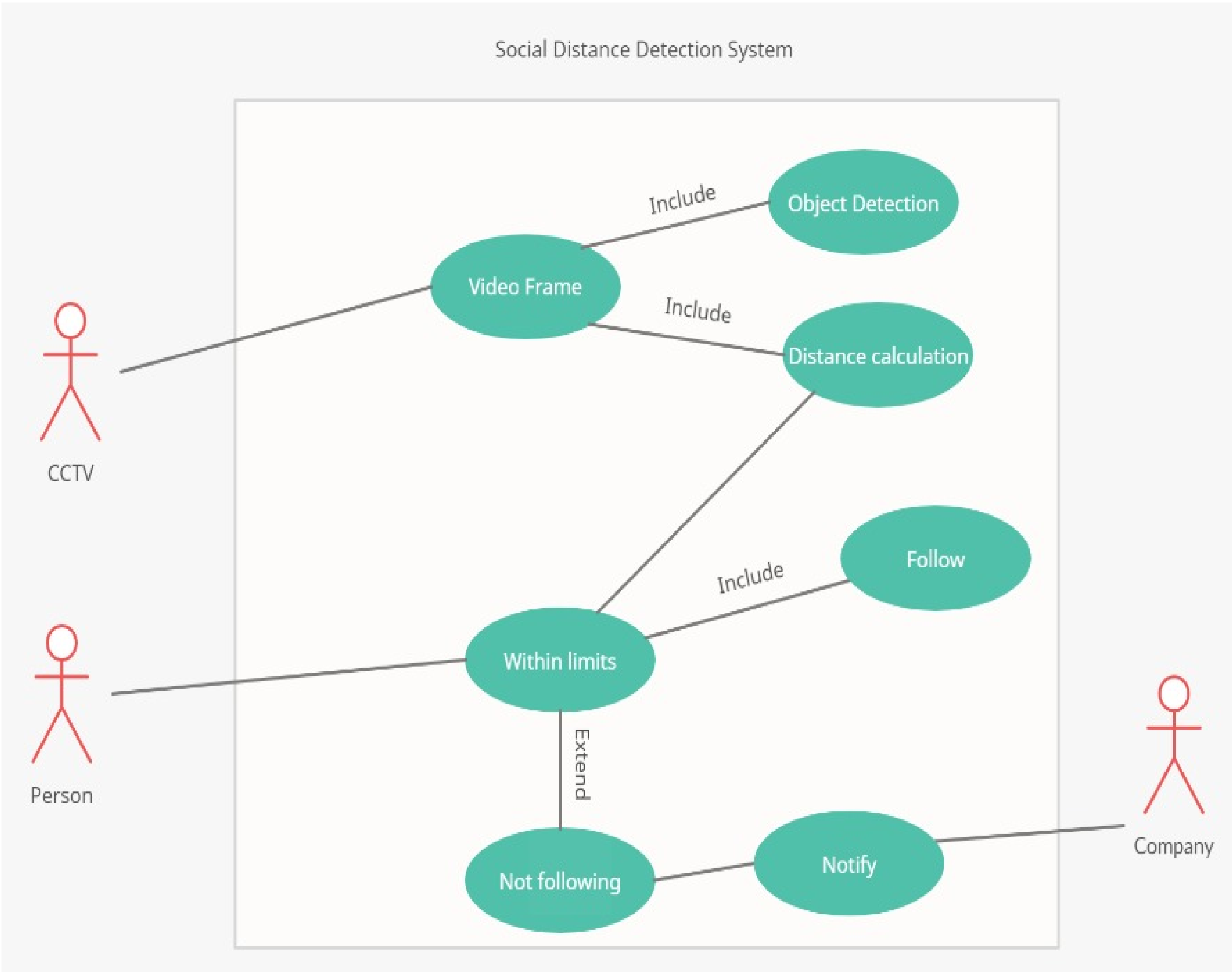


Fig:3.3.2 Use-Case Diagrams

Chapter 4

Analysis Modeling

4.1 Activity Diagrams / Class Diagram

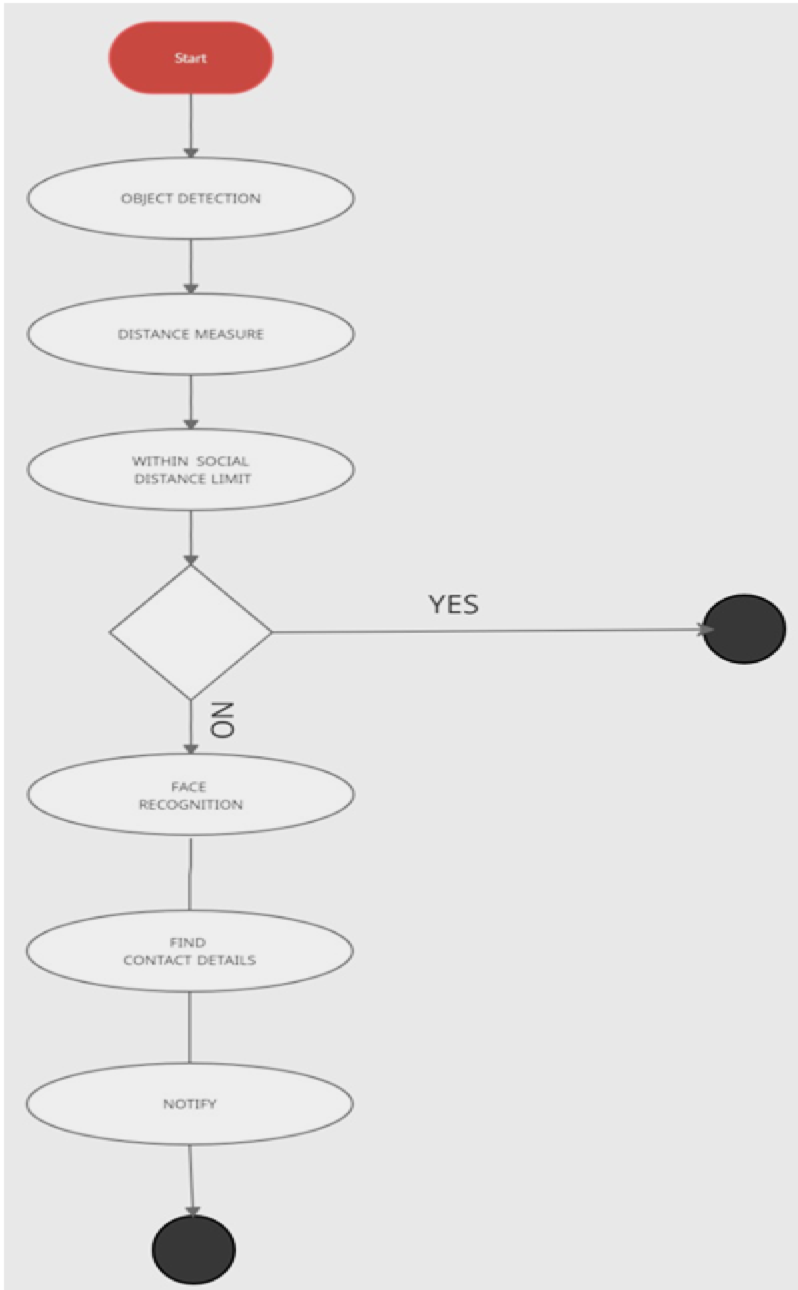


Fig 4.1.1 Activity Diagram

4.2 TimeLine Chart

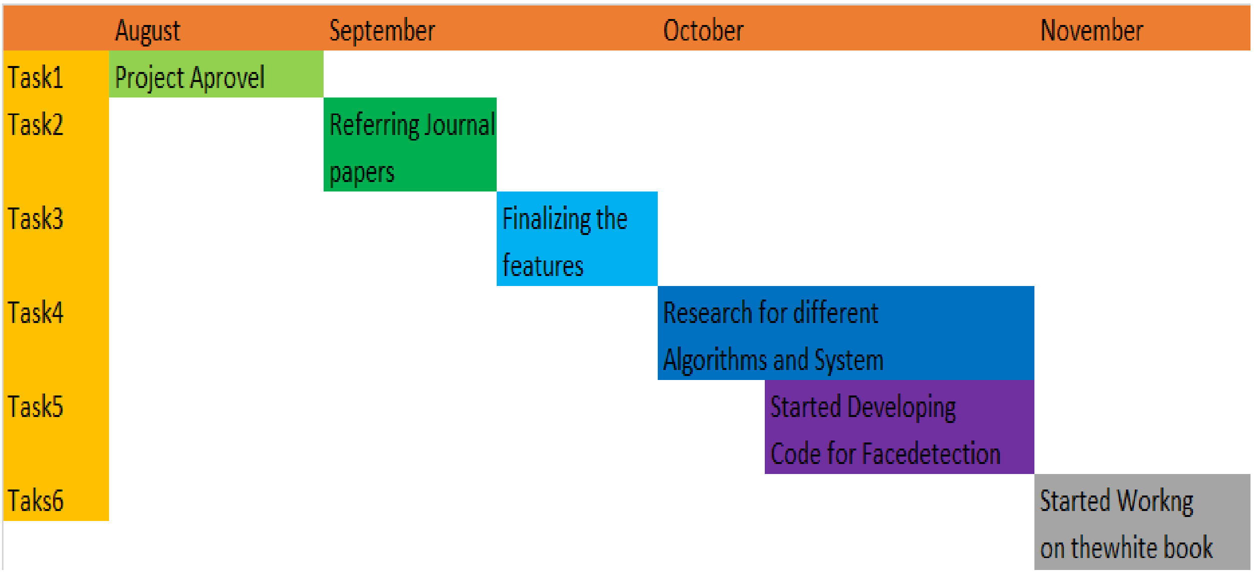


Fig 4.2.1 TimeLine Chart

Chapter 5

Design

5.1 **Architectural Design**

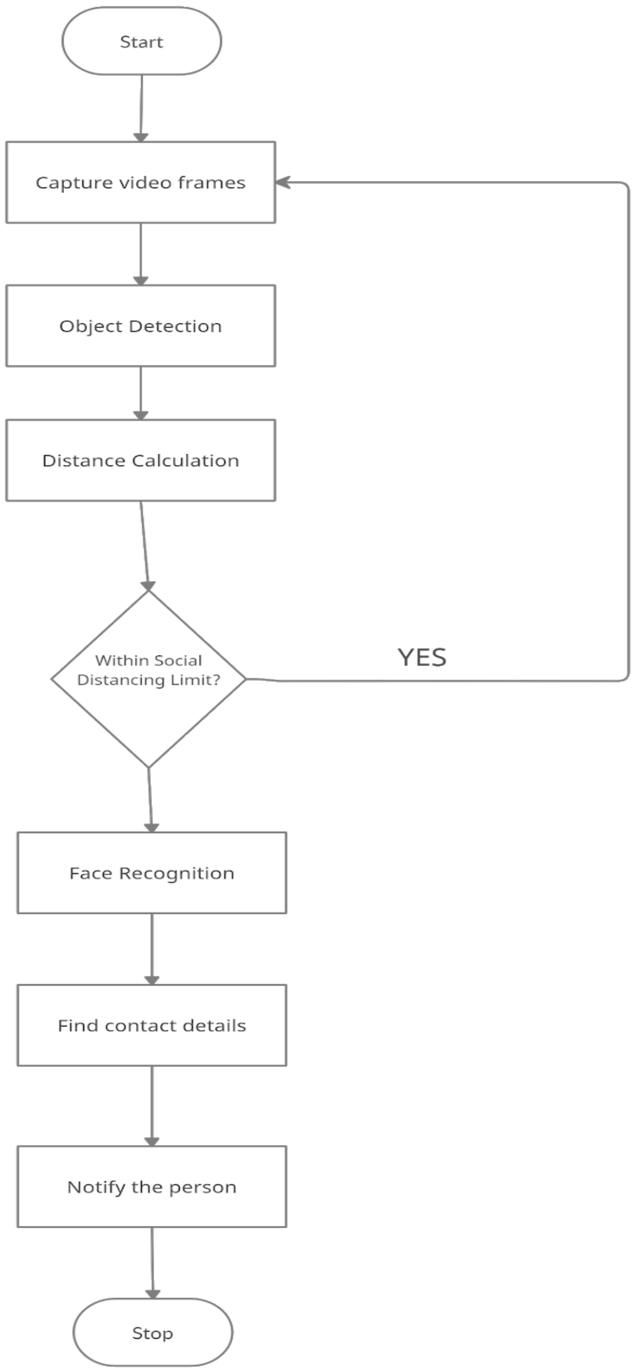


Fig 5.1.1 Architectural Design

User Interface :-

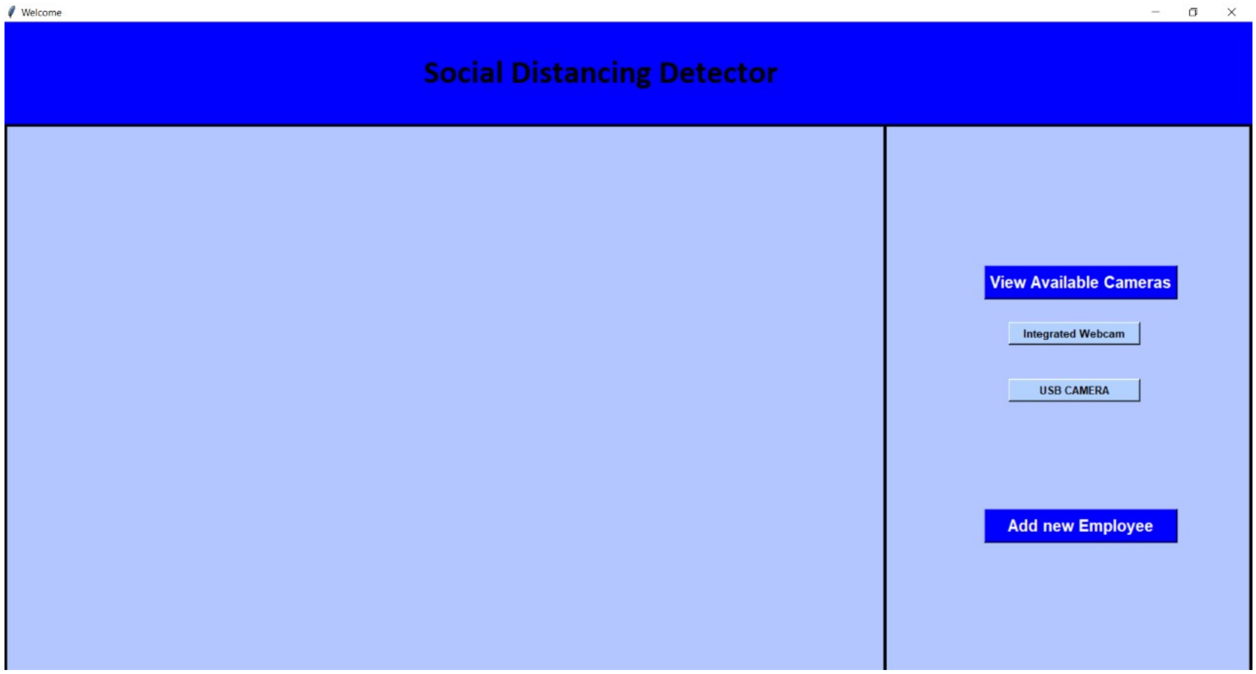


Fig. 5.1.2 Social distance detector GUI

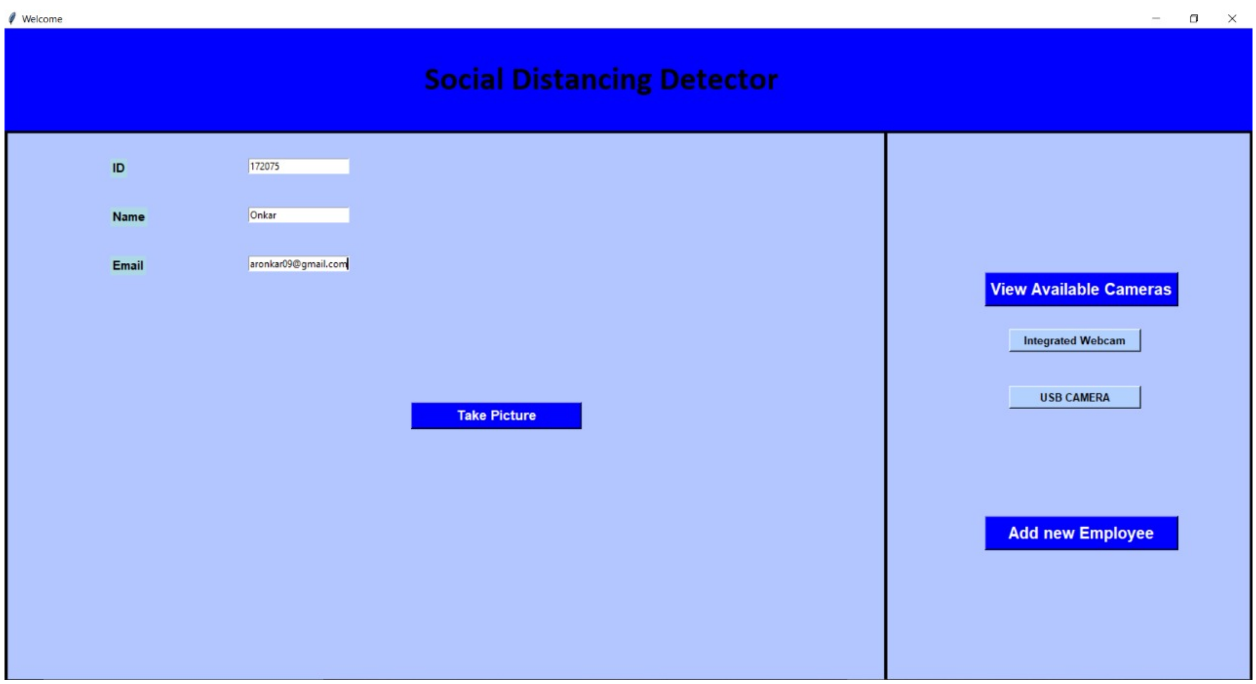


Fig.5.1.3 Enter employee Details

Chapter 6

Implementation

6.1 **Algorithms / Methods Used**

We used the SocialdistancingNet-19 architecture for the training purpose. Box labels were used to create the data for training and evaluation purposes. A rectangular box was used to mark the object. This network comprises 2 subnetworks- feature extraction and feature detection. The feature extraction was carried out by a pre-trained convolutional neural network (CNN) model. We also used a reduced ResNet-50, MobileNet-V2 and ResNet-18 network. The detection of sub-networks of small CNN is compared to feature extraction and is composed of a few convolutional layers specific to the YOLO object detection model. The YOLO detection model is similar to the single-stage detector model. This algorithm views object recognition as a problem of regression, taking a given input image or video stream and concurrently knowing the bounding box coordinates and the corresponding labels of class probabilities. YOLO has three tuning parameters, network input sizes, anchored box, and feature extraction networks. First, the frame is detected. We then computed bounding box coordinates and then derived the center of the bounding box. Using the box coordinates the top-left coordinates are derived. Afterwhich the frame is pre-processed giving three results which are condensed, bounding box, and centroids of each person. The euclidean distance is calculated and used to end the distance between centroids. After the comparison of the distance between the centroids of two individuals, it is compared with the minimum distance in terms of pixels. The pairs are marked as red or green depending on if they have violated social distance or not.

The user specifies the input size and number of classes while choosing a network. With the minimum size for a network, the size of the training image and the computational cost was optimized. We tried the best model as per input size and set of training images and optimized it to handle larger data sets than the current dataset. SocialdistancingNet-19 has an ar-chitecture of 19 layers. The network is fed with an in-put an image of dimension 224x224x3. Then it is further passed through a convolution, batch normalization and ReLU (Rectication Linear Unit) layer each of dimension 112x112x64. After that it is passed through a single max pooling layer, two convolution layers, two batch normalization layers, two ReLU layers and a single ad-dition layer. Each of these layers were of dimension 56x56x64. Further it was passed through single con-volution, batch normalization and ReLU (RecticationLinear Unit) layers, each of dimension

56x56x32. Then it was passed through a global average pooling layer of dimension 1x1x32. And at the end was nally passed through a fully connected and a softmax layer each of dimension 1x1x10. And then we received the class-sication output. The reduced computational cost was having 224x224x3 which was the bare minimum size required to run any network. Image resizing was the only pre-processing operation required before training. Then, the estimated anchor boxes were used for object training to account for resizing before the training. Also, the estimated anchor resizes. This was done to transform the process with the number of anchor boxes estimated in the resized images. And later stored in the processed data directly. Activation layer 40 of ReLU (Rectication Linear Unit) is generally selected for the feature extraction layer and we refresh the activation layer with the detected sub-network. The feature extraction layer outputs the feature maps and down samples by the factor of 16. The amount of downsampling was good to maintain the tread between the special resolution and strength of the extracted feature. This feature extracted downs to the encoder with a stronger image feature that was used to estimate the cost of the special resolution. Data augmentation was carried out to improve the accuracy by randomly transforming the data while training. Data augmentation added more variety during training. And actually, increases the number of labels in the training data samples. The use of transform augmentation during the training al-
low random keeping of images. The associated box labels are also ipped horizontally. Augmentation is not performed for the validation and test data and hence
evaluation can be carried out unbiasedly since the data is unmodified.

6.2 Working of the project

Object Tracing:

1. Outputs recognized objects together with the bounding boxes.
2. These boxes are then measured via their pixel distance .
3. A centroid is calculated for that particular object
4. After location of centroid Each object box is assigned a particular ID’s

Distance Measurement:

1. Since It is not possible for the objects to remain static .
2. The object in continuous motion needs to calculate the distance all the time to maintain social distance.

Steps:

1. Add the Social distance detector software to your existing Camera.
2. Start getting violations data and alerts.
3. Send sms alerts to Admin and violators.

Social Distancing Detector uses existing IP cameras to identify if people are following social distance. The Social Distancing Computer Vision system finds the gap between two persons detected in the camera. The platform generates notifications and calls an external alarm (via speaker) to warn if anyone is found violating laws.

Chapter 7

Results and Discussion

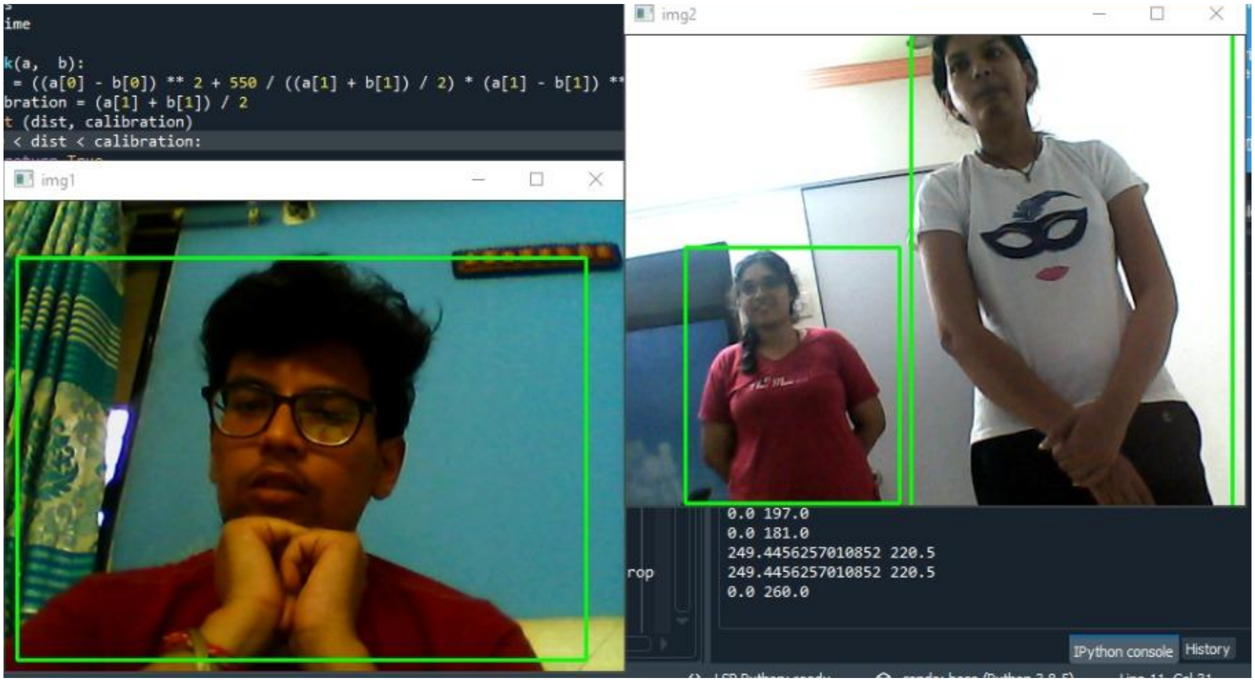


Fig 7.1. Training for distance calculation

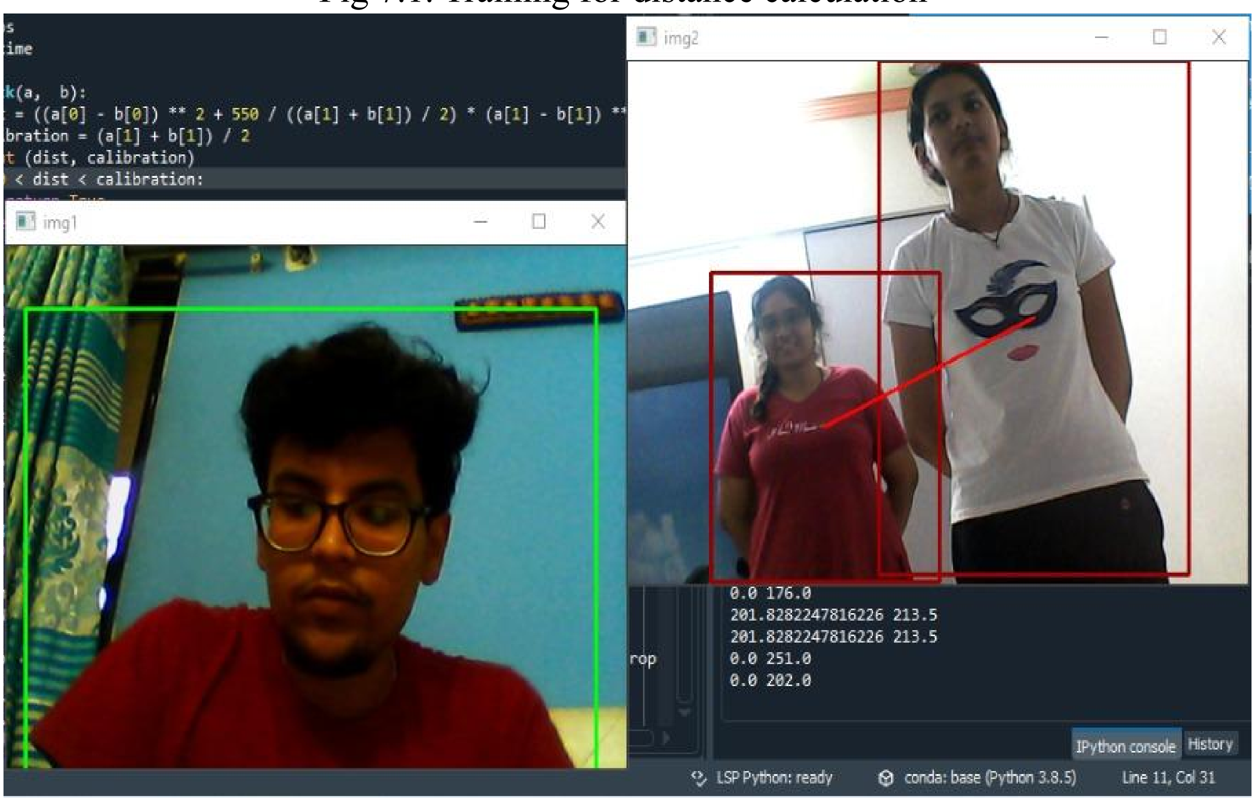


Fig 7.2. Training for distance calculation & Facial recognition

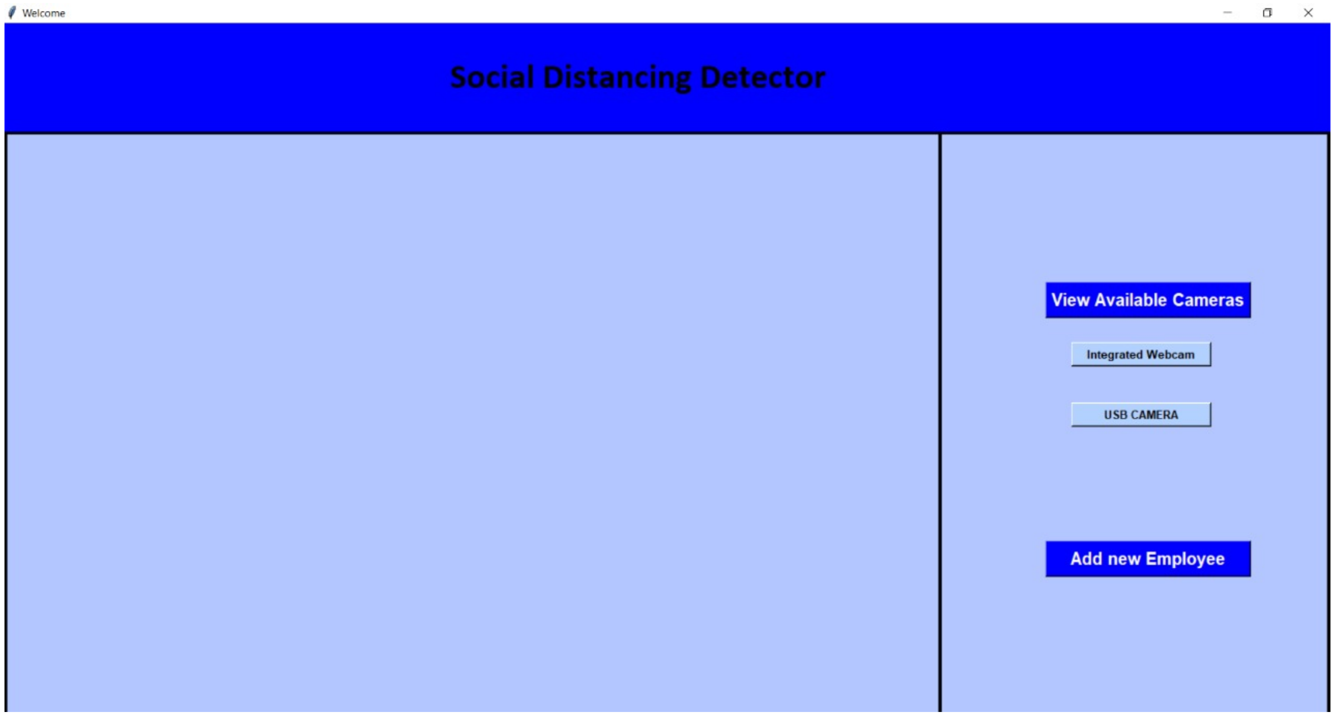


Fig 7.3. Social distance detector GUI

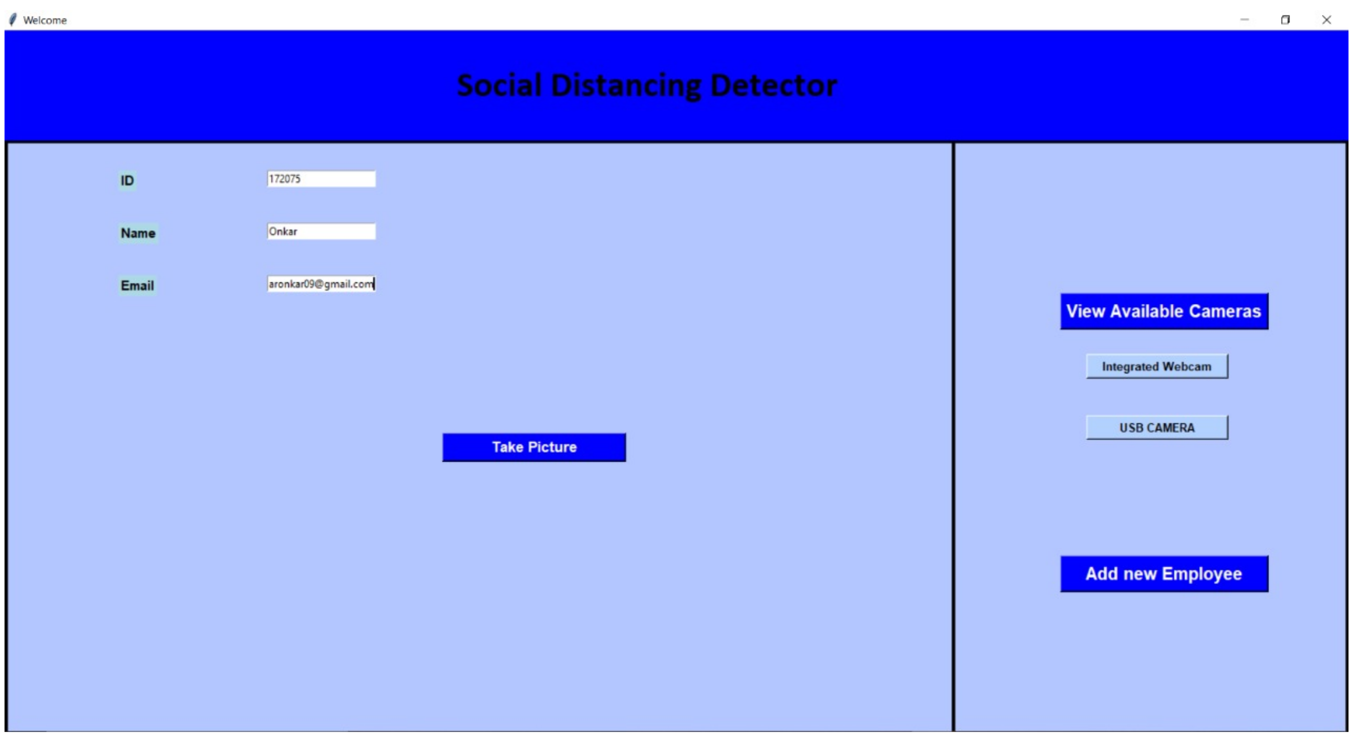


Fig 7.4. Enter employee Details

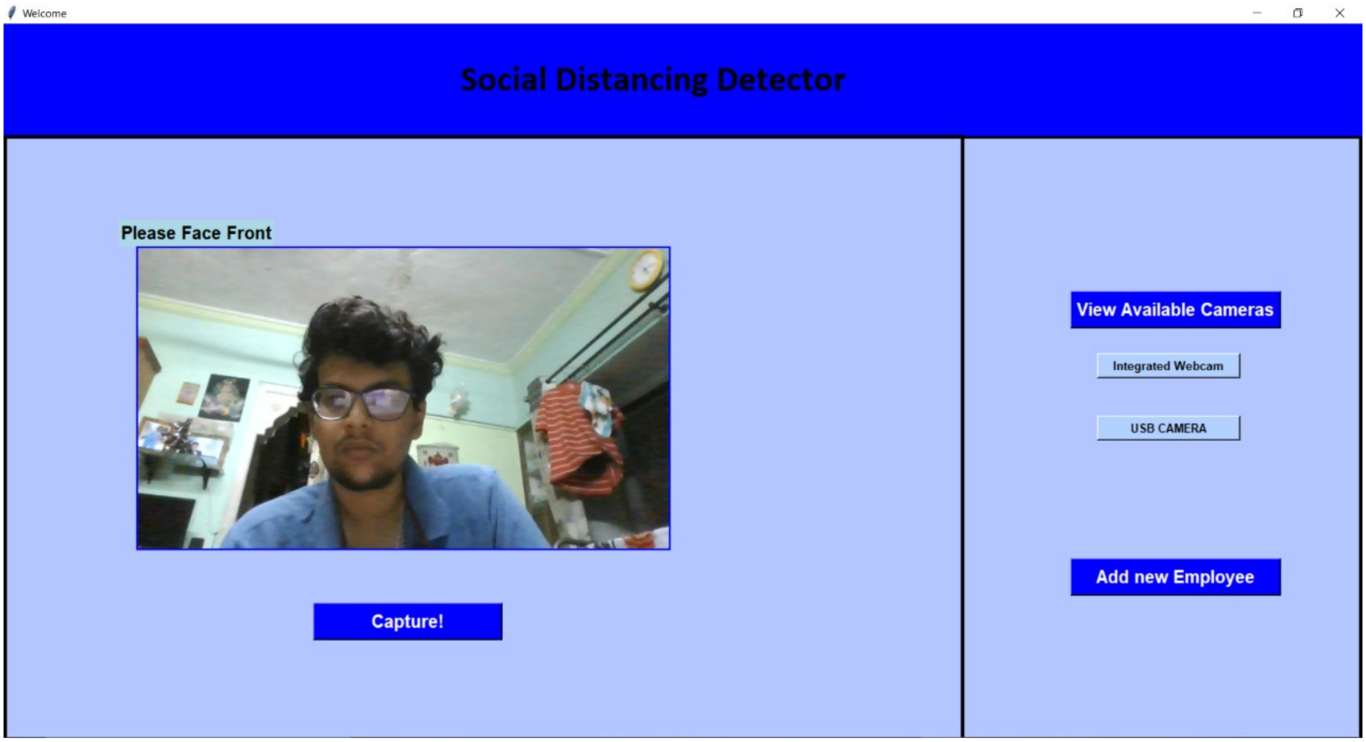


Fig 7.5. Capture Employees Front profile

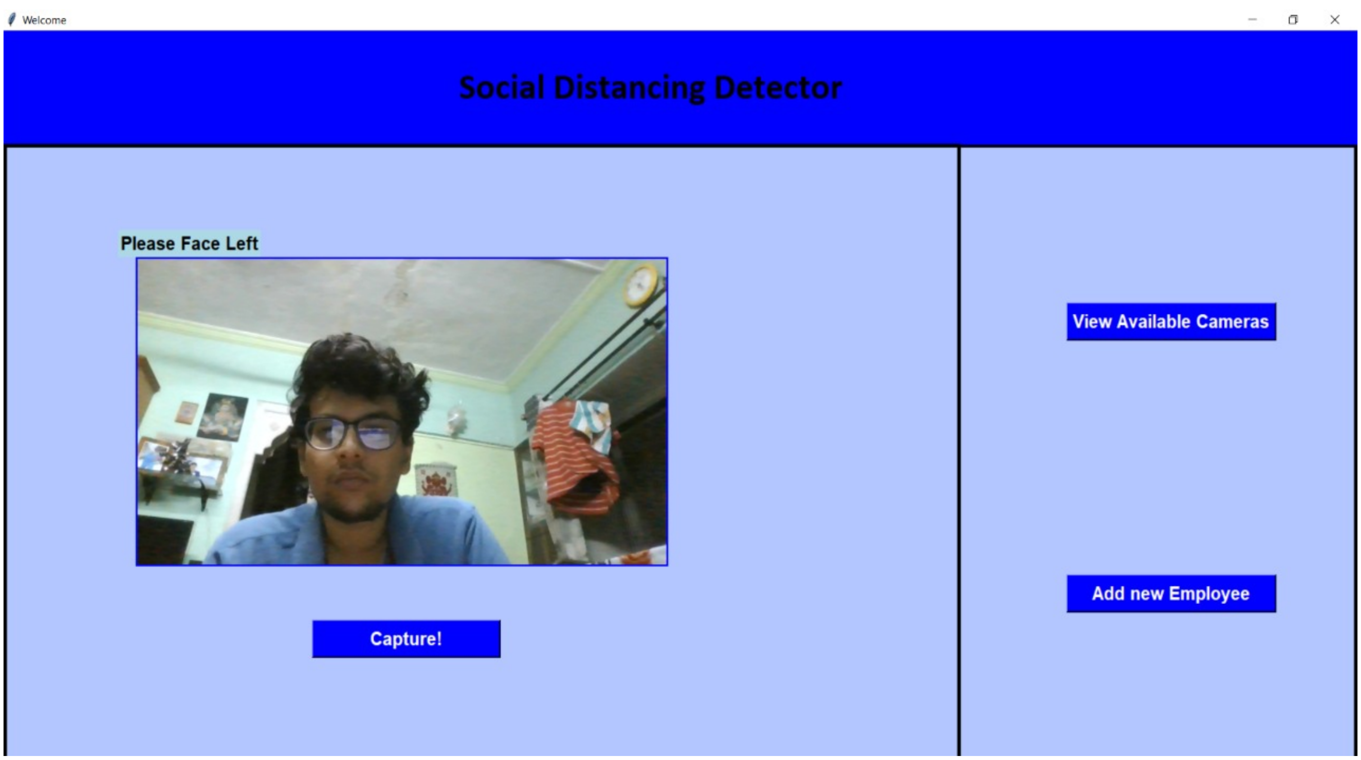


Fig 7.6. Capture Employees Left profile

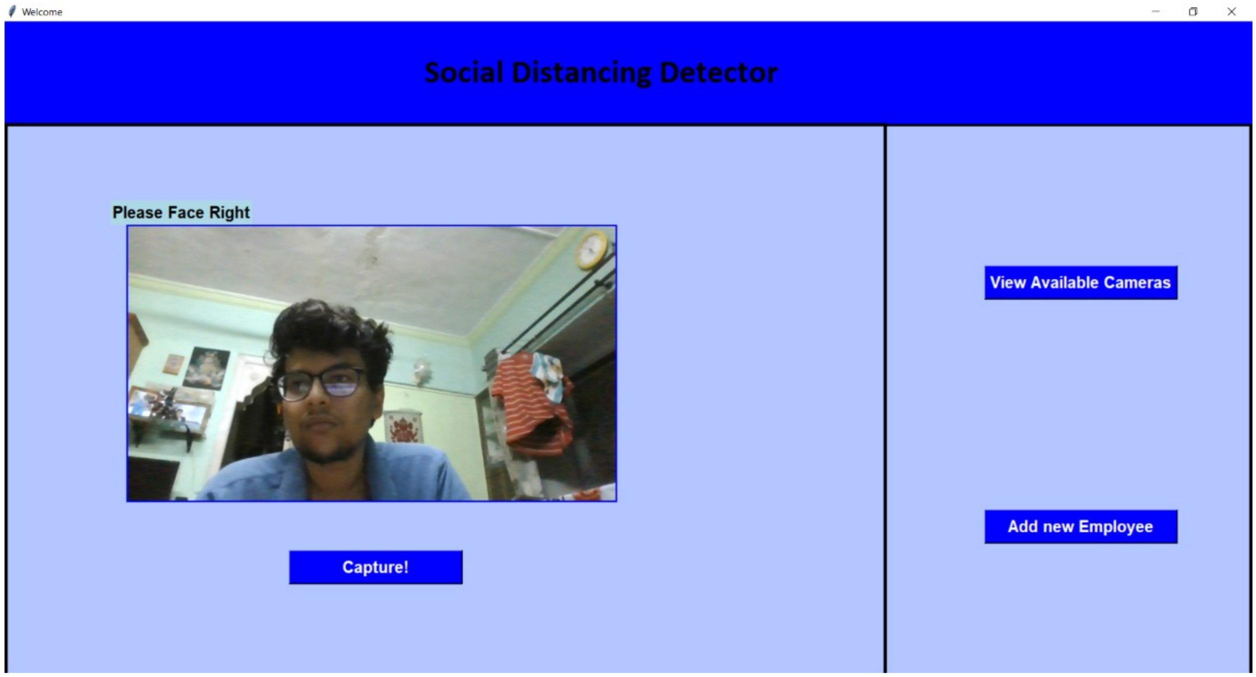


Fig 7.7. Capture Employees Right profile

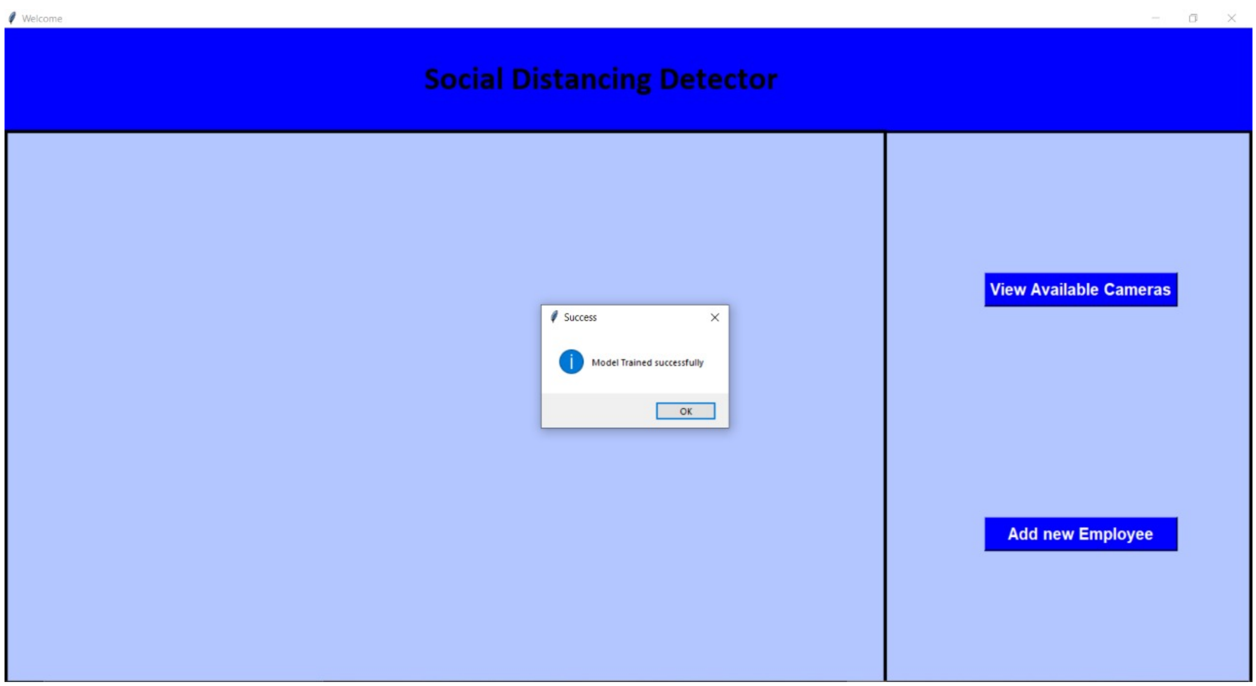


Fig 7.8. Employee Data stored successfully

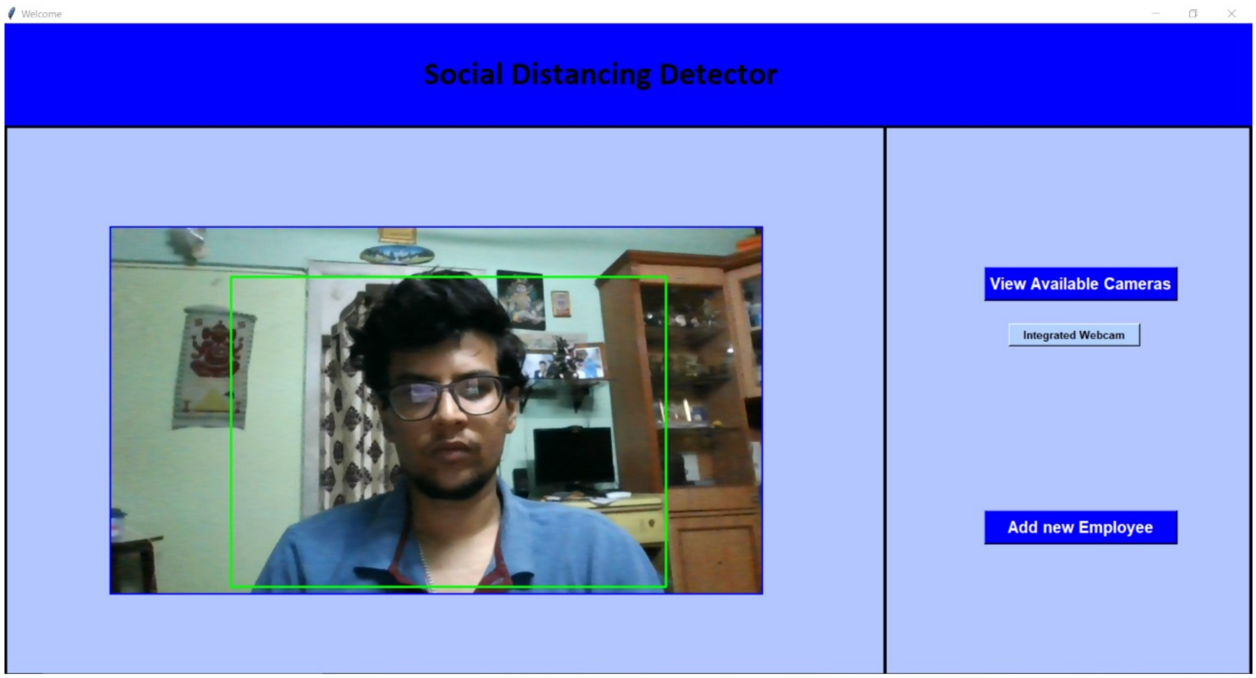


Fig 7.9. Object detection Using YOLO

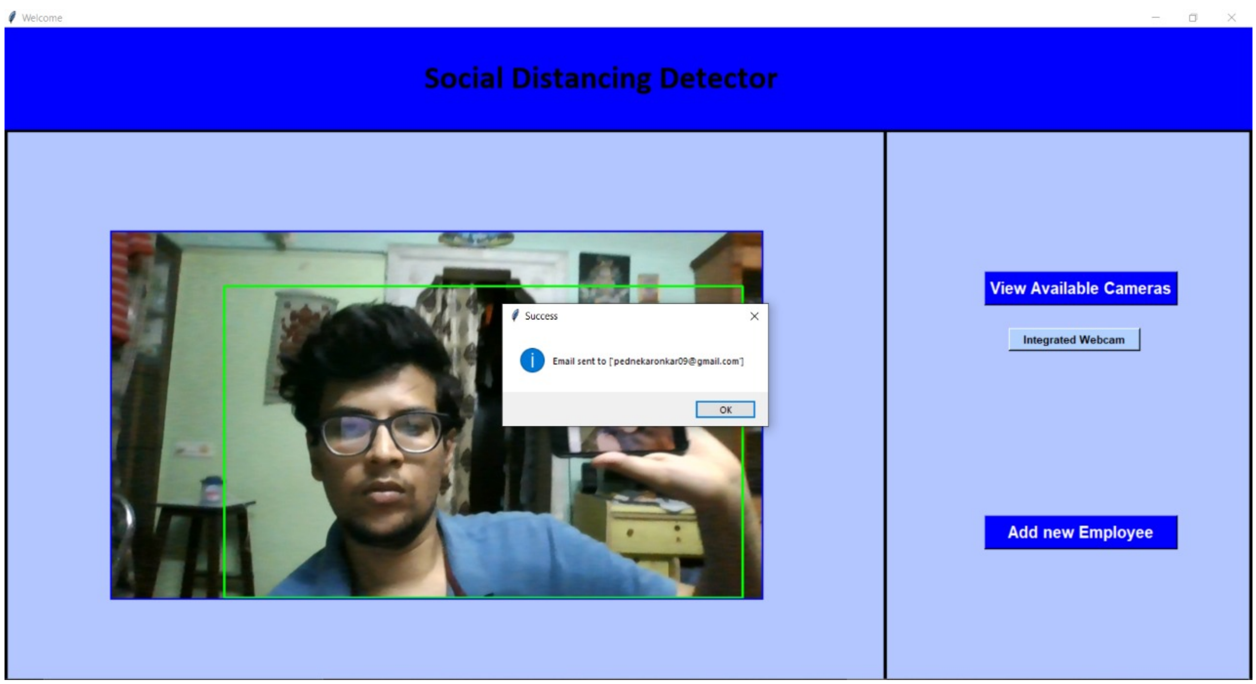


Fig 7.10. Employee Notified

Frames	TP	FP	FN	Prec.	Rec.
1	5	0	3	1	0.62
2	7	0	3	1	0.7
3	6	0	1	1	0.86
4	6	0	1	1	0.86
5	8	0	1	1	0.89
6	7	0	0	1	1
7	7	0	0	1	1
8	4	0	1	1	0.8
9	5	0	0	1	1
10	5	0	0	1	1
11	9	0	2	1	0.82
12	9	0	1	1	0.9
13	9	0	0	1	1
14	8	0	1	1	0.89
15	7	0	2	1	0.78

Fig.7.11 YOLO v3 performance evaluation results

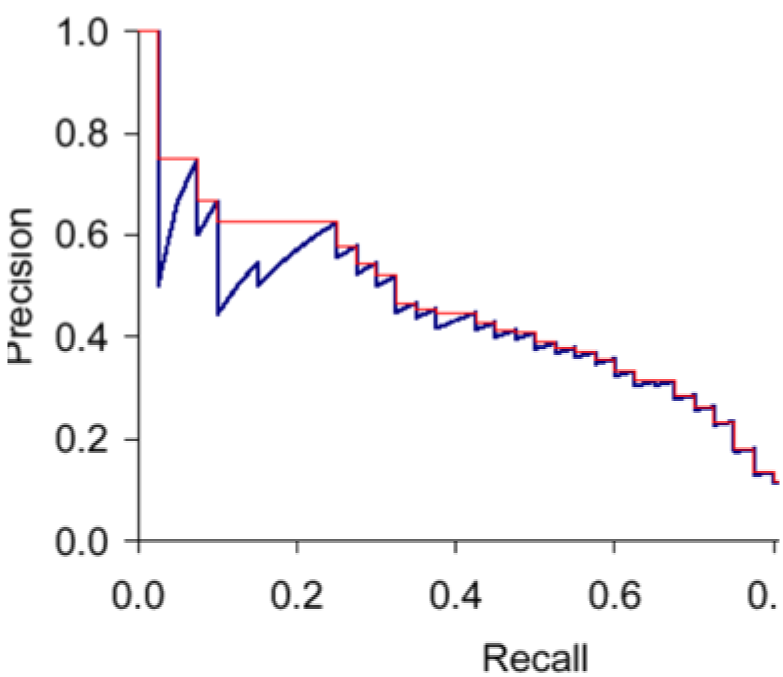


Fig. 7.12 Recall Precision graph

Chapter 9

Conclusion

The emerging trends and the availability of intelligent technologies make us develop new models that help to satisfy the needs of the emerging world. So we have developed a novel social distancing detector which can possibly contribute to public healthcare. The model proposes an efficient real-time deep learning based framework to automate the process of monitoring the social distancing via object detection and tracking approaches, where each individual is identified in the real-time with the help of bounding boxes. Identifying the clusters or groups of people satisfying the closeness property computed with the help of Bird's eye view approach. The number of violations is confirmed by computing the number of groups formed and violation index terms computed as the ratio of the number of people to the number of groups. The extensive trials were conducted with popular state-of-the-art object detection models Faster RCNN, SSD, and YOLO v3, since this approach is highly sensitive to the spatial location of the camera, the same approach can be fine tuned to better adjust with the corresponding field of view. This system works very effectively and efficiently in identifying the social distance between the people and generating the alert that can be handled and monitored.

We evaluate the performance of the trained model by mAP detection metrics. Fig 11. shows precision (Prec), recall (Rec), false positives (FP), true positives (TP), false negatives (FN), and with IoU threshold values. To calculate precision and recall we use the TP, FP, and FN . By summarizing the evaluation results based on the mAP, we can see that the model exhibited overall good performance, network size 416 with IoU threshold 0.5 have the high mAP value of 76.8%.

Our work distinguishes the social distancing pattern and classifies them as a violation of social distancing or maintaining the social distancing norm. Additionally, it also displays labels as per the object detection. The classier was then implemented for live video streams and images also. This system can be used in CCTV for surveillance of people during pandemics. Mass screening is possible and hence can be used in crowded places like railway stations, bus stops, markets, streets, mall entrances, schools, colleges, etc. By monitoring the distance between two individuals, we can make sure that an individual is maintaining social distancing in the right way which will enable us to cure the virus.

References

1. Journal Paper,

A. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, et al., "ImageNet Large Scale Visual Recognition Challenge", International Journal of Computer Vision (IJCV), 2015.**(IEEE)**

B. Rezaei, Mahdi & Azarmi, Mohsen. (2020). DeepSOCIAL: Social Distancing Monitoring and Infection Risk Assessment in COVID-19 Pandemic.**(IEEE)**

C. Monitoring Social Distancing for Covid-19 Using OpenCV and Deep Learning Rucha Visal, Atharva Theurkar, Bhairavi Shukla.**(IRJET)**

2. Proceeding paper,

A. S. Ren, K. He, R. Girshick and J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, 2015, [online].**(IEEE)**

B. OpenCV Social Distancing Detector by Adrian Rosebrock on June 1, 2020.**(pyimagesearch)**

C. Monitoring COVID-19 social distance with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques. Narinder Singh Punn, Sanjay Kumar Sonbhadra, Sonali Agarwal. **(arXiv:2005.01385v2 [cs.CV])**

Acknowledgments

We take the opportunity to thank all those people who have helped and guided us through this project and made this experience worthwhile for us. We wish to sincerely thank our reverend Bro. Jose Thuruthiyal and principal Dr. Sincy George for giving us the opportunity for making a project in the Final Year of Engineering. We would also like to thank the HOD of the Computer Department Dr. Kavita Sonawane and all the teaching and non-teaching staff for their immense support and cooperation. Last but not the least we would like to thank Ms. Nidhi Guar for guiding us throughout the project and encouraging us to explore this domain