A

Major Project

On

SOCIAL DISTANCE DETECTION AND ALERT SYSTEM USING YOLOv3 MODEL

Submitted to

Jawaharlal Nehru Technological University, Hyderabad

In partial fulfillment of the requirements for the award of Degree

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

BY

Ch. Savitha (197R5A0523)

P. Durga Prasad (187R1A05M5)

A. Raghuvaran Reddy (187R1A05P4)

Under the esteemed Guidance of

G. LATHA

(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, NewDelhi)
Recognized Under Section 2(f) & 12(B) of the UGC Act.1956,
Kandlakoya (V), Medchal Road, Hyderabad-501401.

2018-2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled **"SOCIAL DISTANCE DETECTION AND ALERT SYSTEM USING YOLOv3 MODEL"** being submitted by **CH. SAVITHA (197R5A0523), P. DURGA PRASAD (187R1A05M5), A. RAGHUVARAN REDDY (187R1A05P4)** in partial fulfillment of the requirements for the award of the degree of B. Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of Bonafied work carried out by him/her under our guidance and supervision during the year 2021-2022.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

GADEPAKA LATHA
(Associate Professor)
INTERNAL GUIDE

Dr. A. RAJI REDDY
DIRECTOR

Dr. K. SRUJAN RAJU
HOD

EXTERNAL EXAMINER

Submitted on viva voce Examination held on _____

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **G. LATHA** for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) Coordinators: **Mr. A. Uday Kiran, Mr. J. Narasimha Rao, Dr. T.S. Mastan Rao, Mrs. G. Latha, Mr. A. Kiran Kumar,** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to the Head of the Department **Dr. K. Srujan Raju** for providing excellent infrastructure and a nice atmosphere for completing this project successfully.

We are obliged to our Director **Dr. A. Raji Reddy** for being cooperative throughout the course of this project. We would like to express our sincere gratitude to our Chairman Sri. **Ch. Gopal Reddy** for his encouragement throughout the course of this project

The guidance and support received from all the members of **CMR TECHNICAL CAMPUS** who contributed and who are contributing to this project, was vital for the success of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement without which this assignment would not be possible. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project.

CH. SAVITHA (197R5A0523)

P. DURGA PRASAD (187R1A05M5)

A. RAGHUVARAN REDDY (187R1A05P4)

ABSTRACT

With no doubt, the infectious diseases pandemic has put the world to a halt. The world we lived in a few months prior is completely different from what it is now. The virus is spreading quickly and is a danger to the human race. Seeing the necessity of the hour one must always take certain precautions of which one being social distancing. Maintaining social distancing during this pandemic situation is necessary to ensure a slowdown in the growth rate of new cases. Our manuscript focuses on detecting if the people around are maintaining social distancing or not. Using our own self-developed YOLO v3 for detecting the frame of a person and displaying labels, they are marked as safe or unsafe if the distance is less than a certain value. This system can be used for monitoring people via video surveillance in CCTV. Our model achieved an accuracy of 92.8 %.

LIST OF FIGURES

LIST OF SCREENSHOTS

TABLE OF CONTENTS **PAGE NO'S**

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

The project titled as “SOCIAL DISTANCE DETECTION AND ALERT SYSTEM USING YOLOv3 MODEL”. The absence of any active therapeutic agents and the lack of immunity against COVID-19 increases the vulnerability of the population. Since there are no vaccines available, social distancing is the only feasible approach to fight against this pandemic. Motivated by this notion, this article proposes a deep learning-based framework for automating the task of monitoring social distancing using surveillance video.

1.2 PROJECT PURPOSE

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 curb the virus.

1.3 PROJECT FEATURES

This system can be used in CCTV for surveillance of people during pandemics Focusing on surveillance of public places and detecting whether the people are maintaining social distancing or not. Social Distancing is the only best option for us to protect ourselves from diseases, not limited to COVID-19, where no medicinal antidote has been prepared, and that may be transmitted.

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

2.1 PROBLEM DEFINITION

The system presented is for analyzing social distancing by calculating the distance between people in order to slow down the spread of the virus.

This system utilizes input from video frames to figure out the distance between individuals to alleviate the effect of this pandemic.

The results and outcomes obtained by the system show that evaluation of the distance between multiple individuals and determining if rules are violated or not.

2.2 EXISTING SYSTEM

In existing System determining whether persons during an outbreak can use social distancing and associated self-protective behaviors. The differential game is used as a mitigating tool to research the possible utility of social distancing by measuring the equilibrium actions under several cost functions. Following outbreak detection, computational techniques are used to measure the cumulative expense of an infection under equilibrium practices because of the period until mass vaccination. The main parameters in the study are the specific number of reproductions and the underlying efficacy of social distancing. To slow the spread of the COVID-19 virus via airborne transmission a" social distancing" approach of around 1.83 m.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

- It has Outbreak in identifying person using mitigating tool.

2.3 PROPOSED SYSTEM

The proposed framework utilizes the YOLO v3 object detection model to segregate humans from the background and approach to track the identified people with the help of bounding boxes and assigned IDs. The results of the YOLO v3 model are further compared with other popular state-of-the-art models, e.g., faster region-based CNN (convolution neural network).

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

- It is used to track persons while moving easily.
- It will be helpful for maintain social distance.
- By this there is a chance to prevent Covid-19 and upcoming Omicron virus.

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis, the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.4.1 ECONOMIC FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

- Operating system : windows, linux
- Processor : intel i7
- Ram : minimum 4 gb
- Hard disk : minimum 250gb

2.5.2 SOFTWARE REQUIREMENTS

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what the areas of strength and deficit are and how to tackle them.

- Python idel 3.7 version (or)
- Anaconda Environment 3.7
- Jupiter Notebook (editor)

3. ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture describes how the application is going to function. This describes how the user's request is taken as input and how the output is delivered to them. The detailed architecture is explained below.

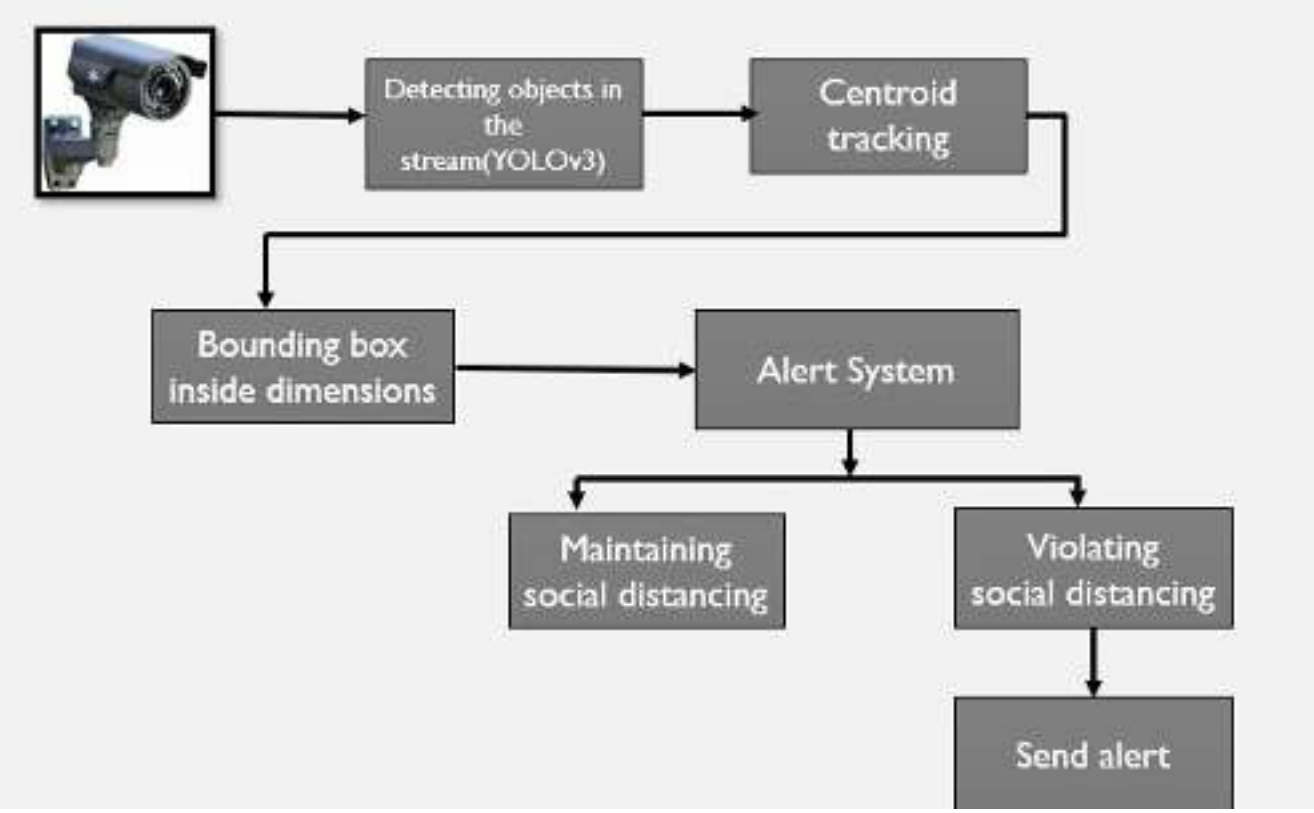

Fig 3.1: Architecture

3.2 MODULES DESCRIPTION

3.2.1 YOLOv3

The YOLOv3 module is an object detection 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 network. First, the frame is detected. We then compute bounding box coordinates and then derived the center of the bounding box. Using the box coordinates, the top-left coordinates are derived. After which the frame is pre-processed giving three results, which are confidence, bounding box, and centroids of each person.

3.2.2 Centroid Tracking

Centroid Tracking is a module, which is used for distance computation. The Euclidean distance is calculated and used to find the distance between centroids. After the comparison of the distance the centroids of two individuals, it is compared with the minimum distance in terms of pixels.
Violation's detection module detects the distance violations of an individuals. The pairs are marked as red or green depending on if they have violated social distancing or not.
Then voice alert is sent, if distance is violated. 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 to find the best model as per input size and set of training images and optimize it to handle larger data sets than the current dataset.

3.3 UML DIAGRAMS

3.3.1 USECASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

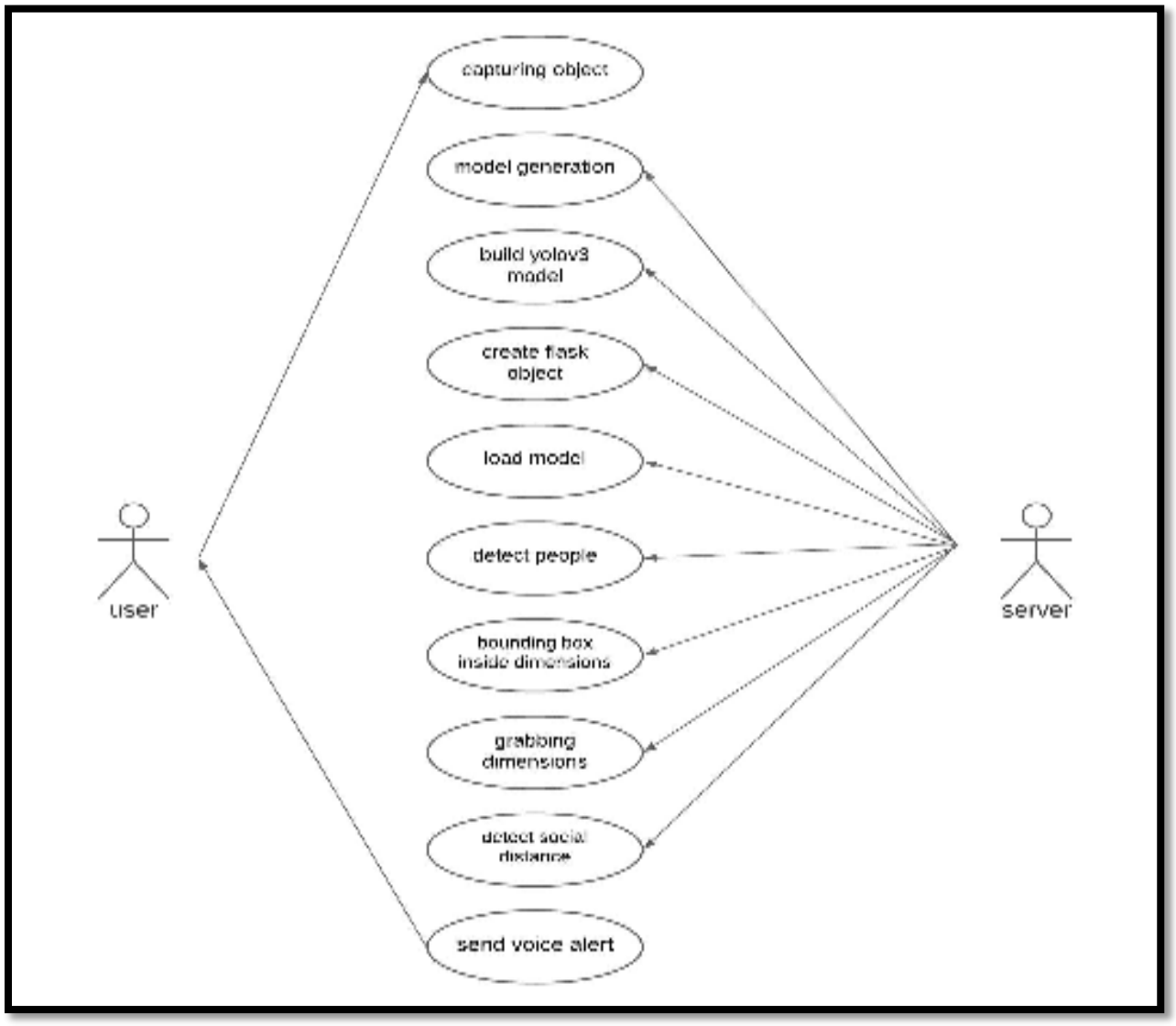

Fig 3.3.1 Usecase diagram

3.3.2 CLASS DIAGRAM

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.

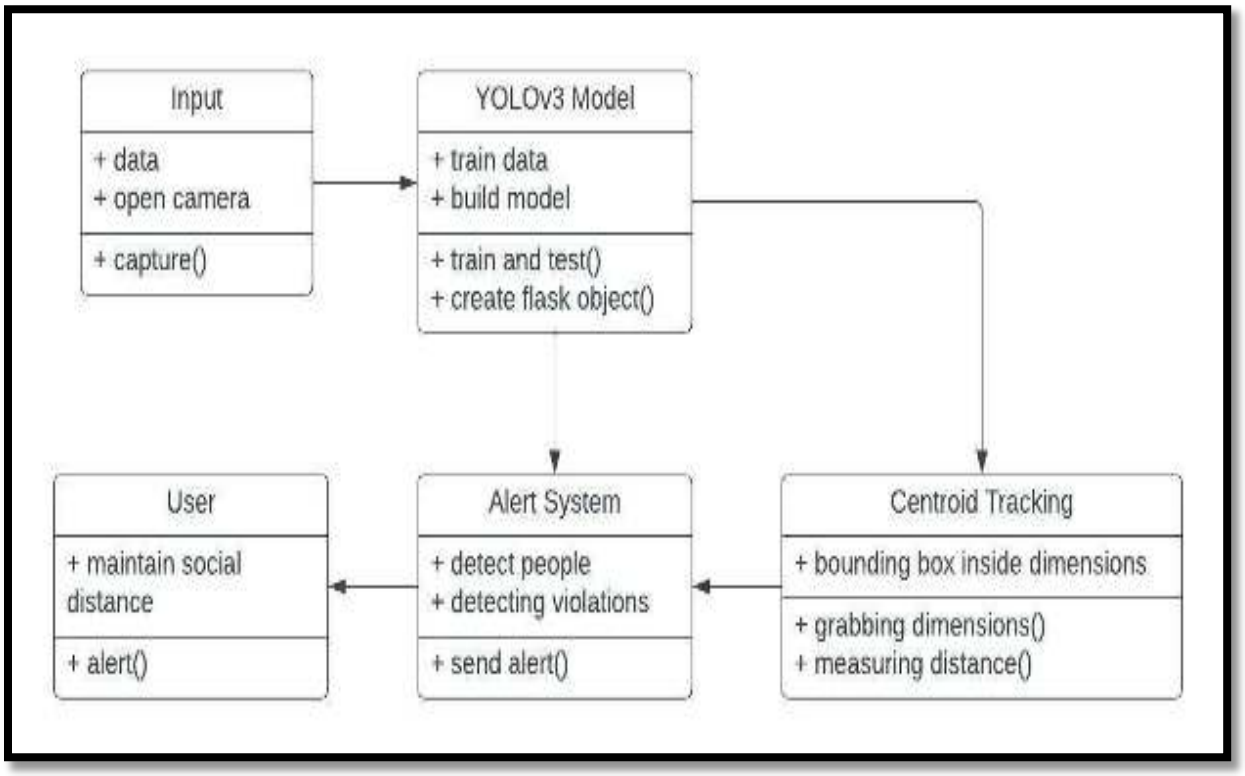

Fig 3.3.2 Class diagram

3.3.3 SEQUENCE DIAGRAM

A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented systematically. Different objects in the sequence diagram interact with each other by passing "messages".

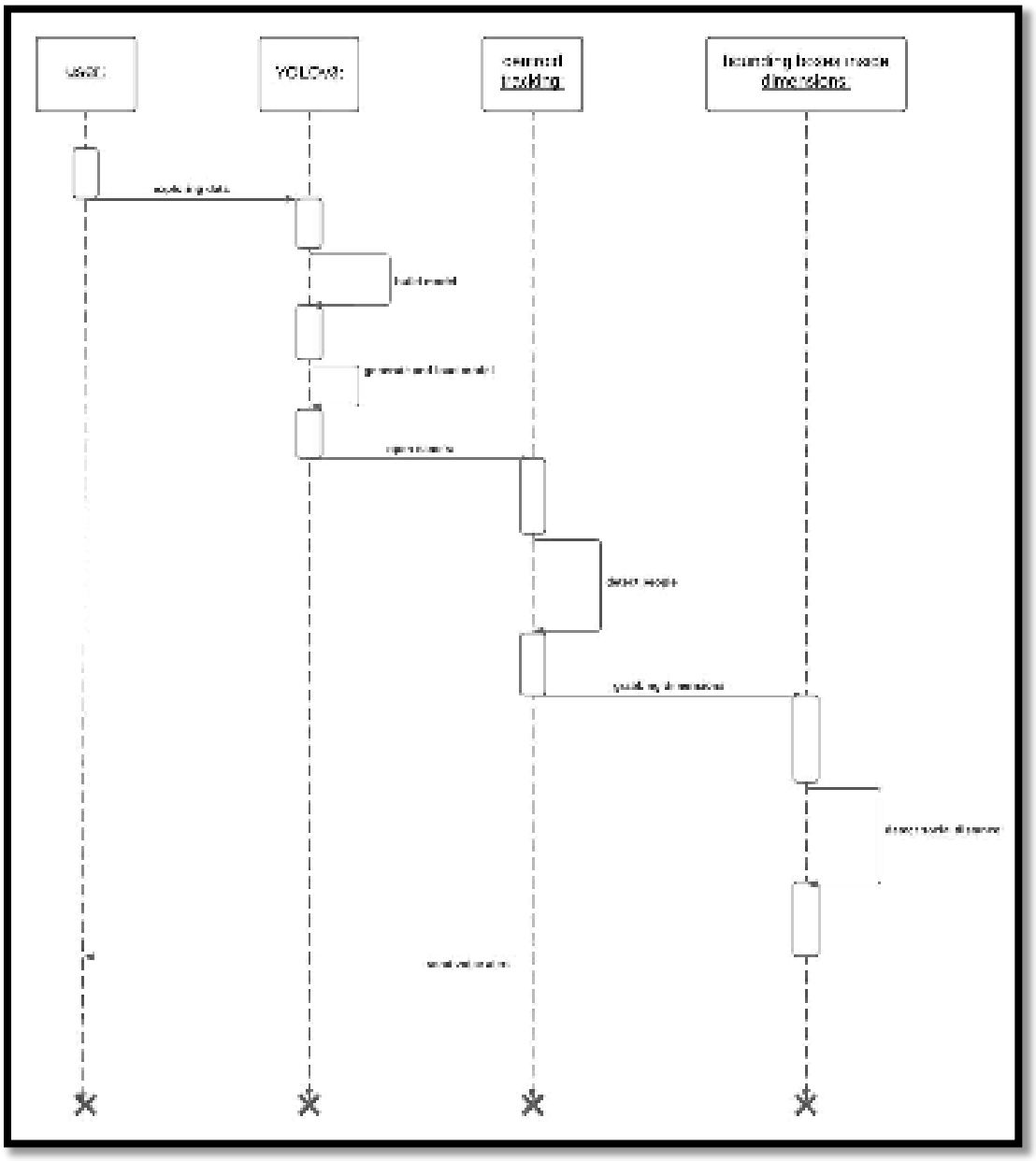

Fig 3.3.3 Sequence diagram

3.3.4 ACTIVITY DIAGRAM

The process flows in the system are captured in the activity diagram. Similar to a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions

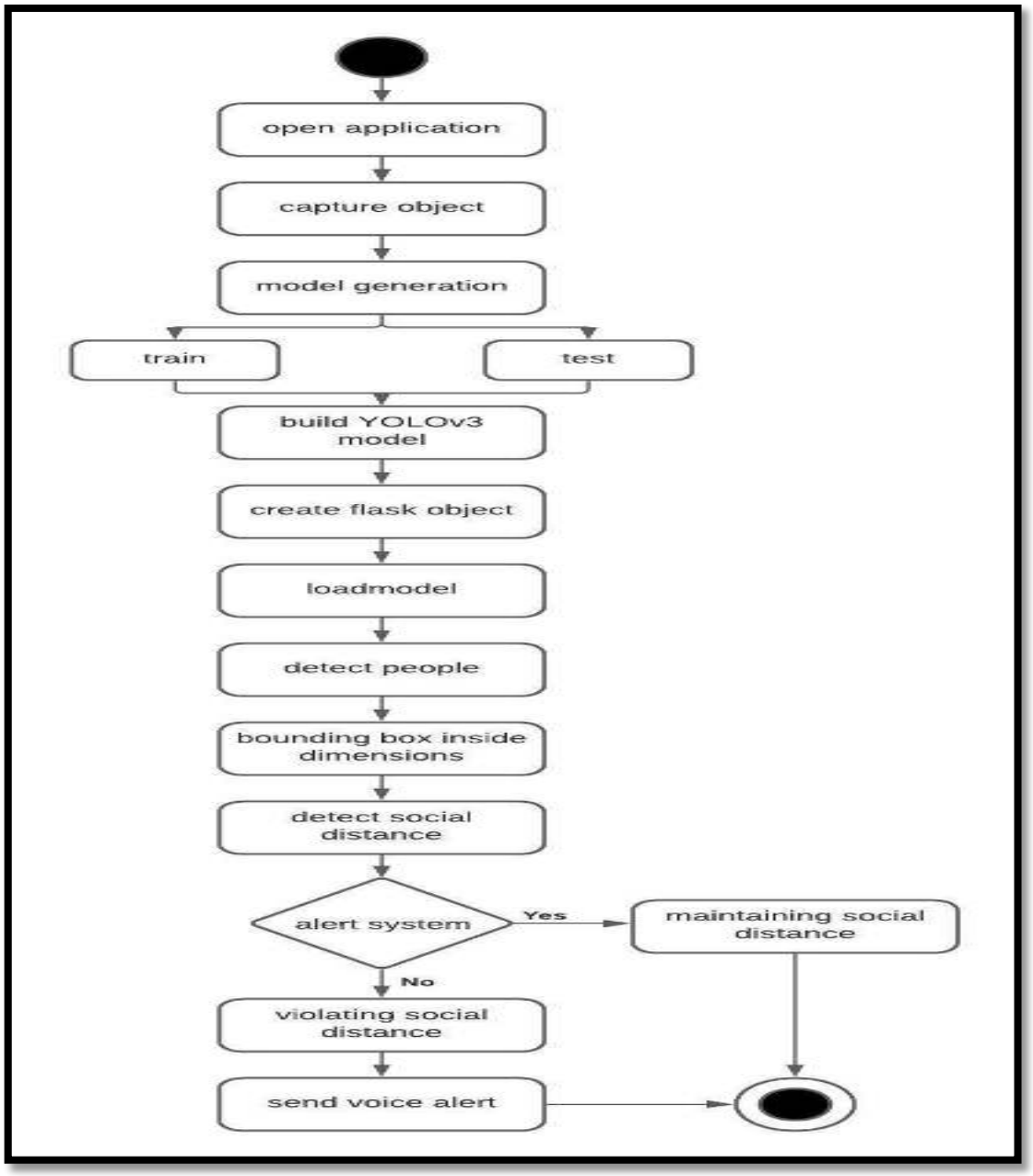

Fig 3.3.4 Activity diagram

3.3.5 DATAFLOW DIAGRAM

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

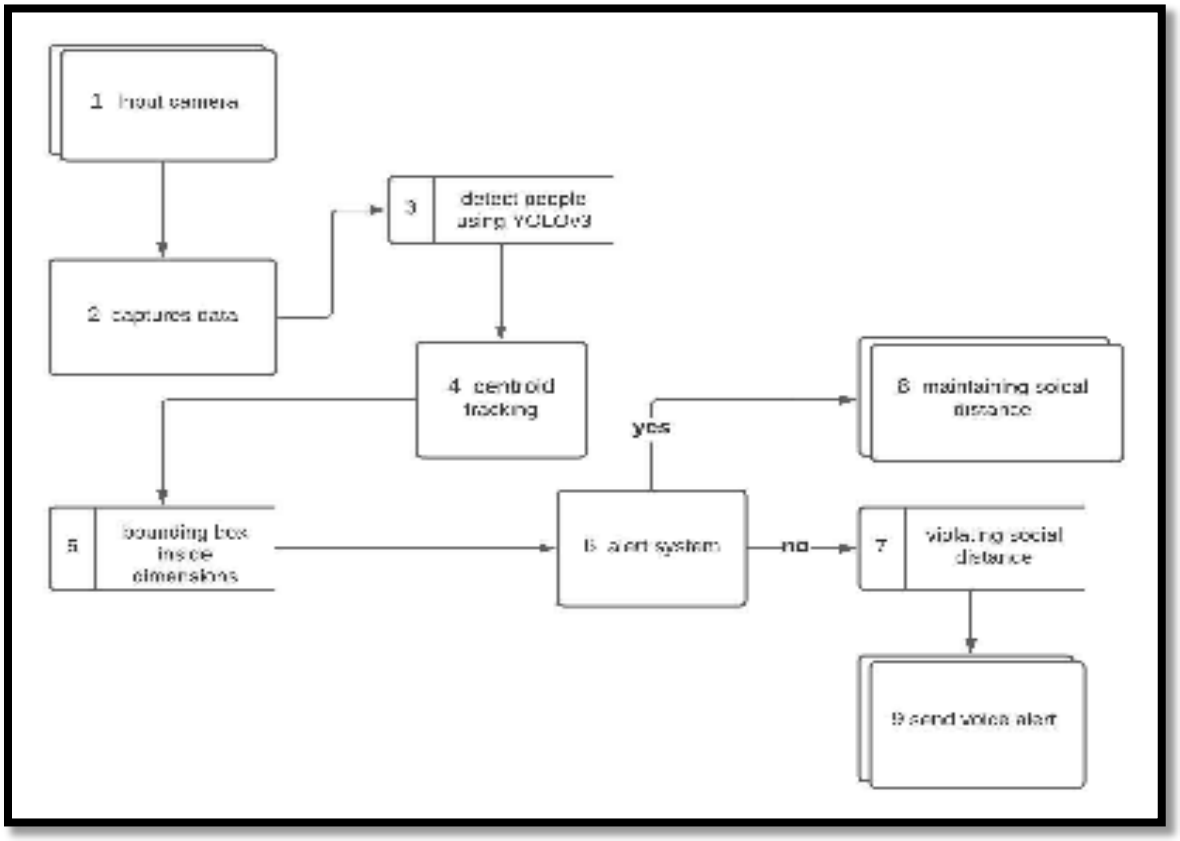

Fig 3.3.5 Dataflow diagram

4. IMPLEMENTATION

4. IMPLEMENTATION

4.1 SAMPLE CODE

```
# Import packages
from tensorflow. keras. applications.mobilenet_v2 import preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
import numpy as np
import imutils
import time
import cv2
import math
from playsound import playsound

labelsPath = "coco.names"
LABELS = open(labelsPath).read().strip().split("\n")
np.random.seed(42)
COLORS = np.random.randint(0, 255, size=(len(LABELS), 3),
                           dtype="uint8")
weightsPath = "yolo-coco/yolov3.weights"
configPath = "yolo-coco/yolov3.cfg"

net = cv2.dnn.readNetFromDarknet(configPath, weightsPath)

# face mask classification
confidence_threshold = 0.4

# load our serialized face detector model from disk
print("[INFO] loading face detector model...")
prototxtPath = "face_detector/deploy.prototxt"
```

```
weightsPath = "face_detector/res10_300x300_ssd_iter_140000.caffemodel"
faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)

# load the face mask detector model from disk
model_store_dir= "classifier.model"
maskNet = load_model(model_store_dir)

cap = cv2.VideoCapture(0)
while (cap.isOpened()):
    ret, image = cap.read()

    if ret == False:
        break

    image = cv2.resize(image, (640, 360))
    (H, W) = image.shape[:2]
    ln = net.getLayerNames()
    ln = [ln[i[0] - 1] for i in net.getUnconnectedOutLayers()]
    blob = cv2.dnn.blobFromImage(image, 1/255.0, (416, 416), swapRB=True, crop=False)
    net.setInput(blob)
    start = time.time()
    layerOutputs = net.forward(ln)
    end = time.time()
    print("Time taken to predict the image: {:.6f}seconds".format(end-start))
    boxes = []
    confidences = []
    classIDs = []

    for output in layerOutputs:
        for detection in output:
            scores = detection[5:]
```

```
        classID = np.argmax(scores)
        confidence = scores[classID]
        if confidence > 0.1 and classID == 0:
            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, 0.5, 0.3)
ind = []
for i in range(0, len(classIDs)):
    if (classIDs[i] == 0):
        ind.append(i)

a = []
b = []
#color = (0, 255, 0)
if len(idxs) > 0:
    for i in idxs.flatten():
        (x, y) = (boxes[i][0], boxes[i][1])
        (w, h) = (boxes[i][2], boxes[i][3])
        a.append(x)
        b.append(y)
        #cv2.rectangle(image, (x, y), (x + w, y + h), color, 2)

distance = []
nsd = []
for i in range(0, len(a) - 1):
```

```
    for k in range(1, len(a)):
        if (k == i):
            break
        else:
            x_dist = (a[k] - a[i])
            y_dist = (b[k] - b[i])
            d = math.sqrt(x_dist * x_dist + y_dist * y_dist)
            distance.append(d)
            if (d <= 100.0):
                nsd.append(i)
                nsd.append(k)
            nsd = list(dict.fromkeys(nsd))

color = (0, 0, 255)
for i in nsd:
    (x, y) = (boxes[i][0], boxes[i][1])
    (w, h) = (boxes[i][2], boxes[i][3])
    cv2.rectangle(image, (x, y), (x + w, y + h), color, 2)
    text = "Alert"
    playsound('sound/alert.mp3')
    cv2.putText(image, text, (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)
color = (138, 68, 38)
if len(idxs) > 0:
    for i in idxs.flatten():
        if (i in nsd):
            break
        else:
            (x, y) = (boxes[i][0], boxes[i][1])
            (w, h) = (boxes[i][2], boxes[i][3])
            cv2.rectangle(image, (x, y), (x + w, y + h), color, 2)
            text = 'OK'
```

```
            cv2.putText(image, text, (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)

    (h, w) = image.shape[:2]
    blob = cv2.dnn.blobFromImage(image, 1.0, (416, 416), (104.0, 177.0, 123.0))
    faceNet.setInput(blob)
    detections = faceNet.forward()
    cv2.imshow("Image", image)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

5. RESULTS

5. RESULTS

5.1 COMMAND PROMPT

Open the Command prompt option.

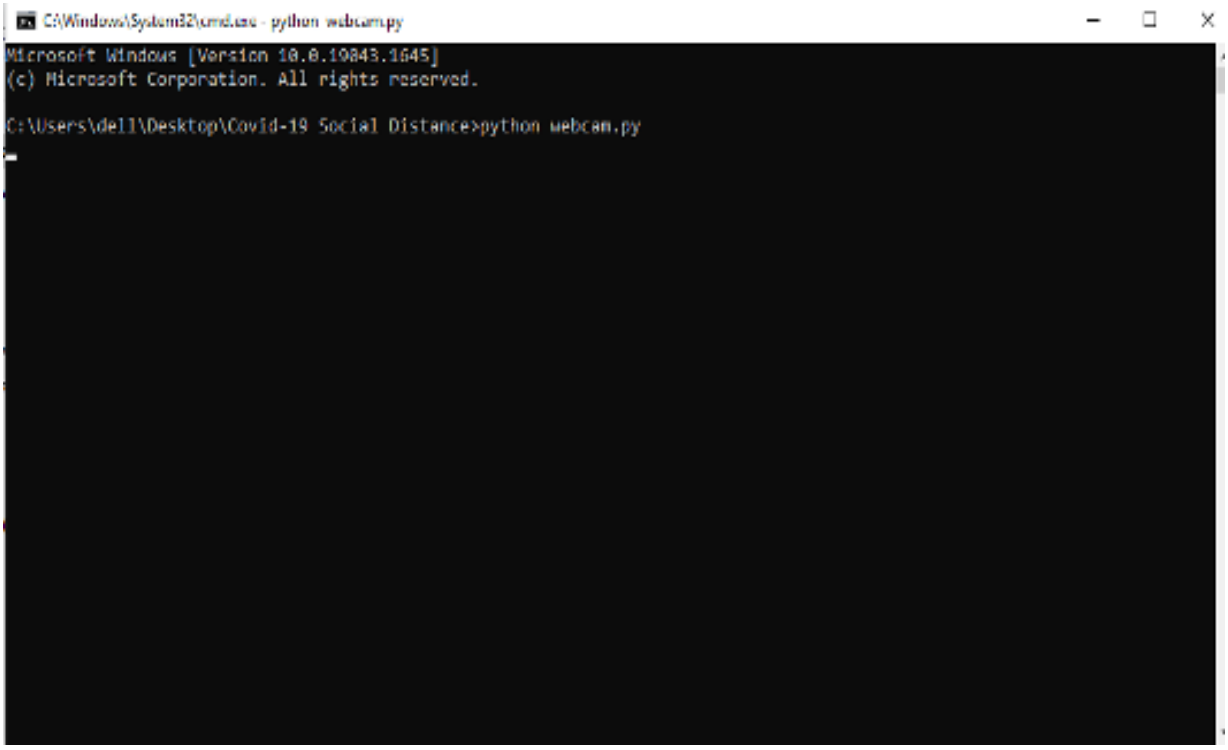

Fig 5.1 Command prompt

5.2 OUTPUT SCREENSHOTS

People maintaining social distance will be bounded in blue color bounded boxes.

People violating rule of maintaining social distance will be bounded in red color bounded boxes and an alert will be sent.

Fig 5.2.1: maintaining social distance

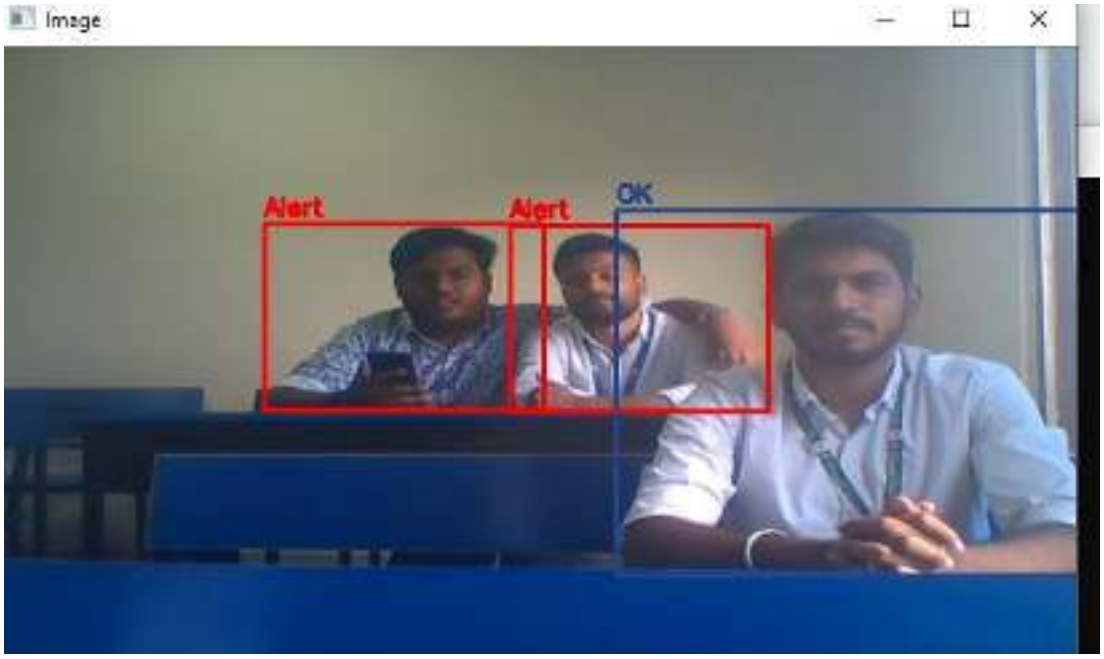

Fig 5.2.2: distance violations and alert sound

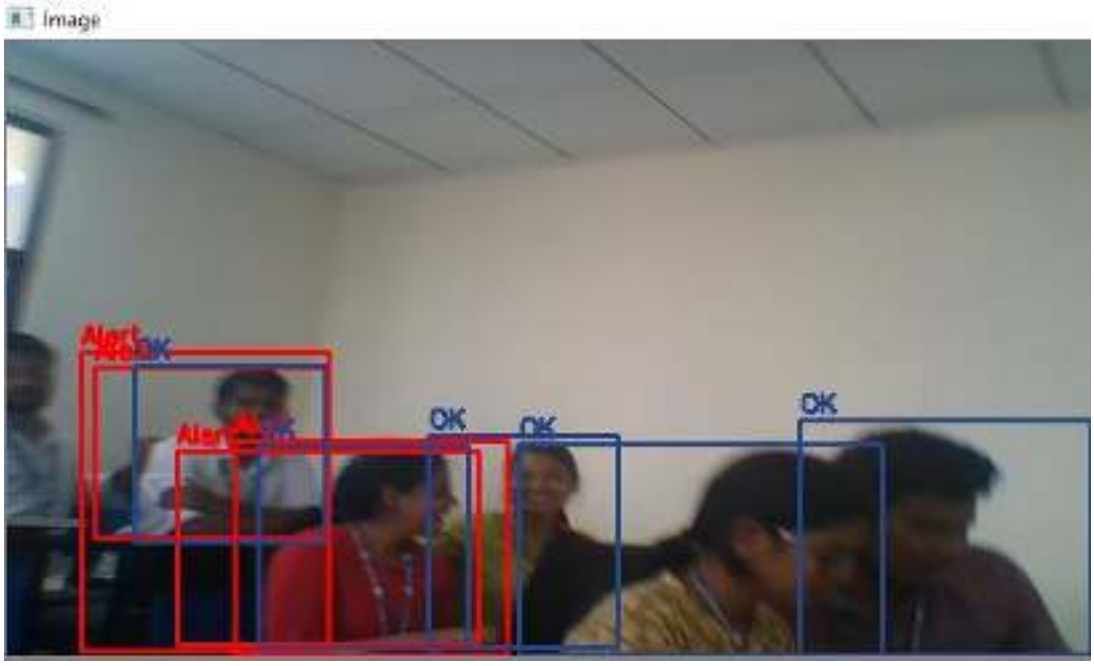

Fig 5.2.3: crowded people in college

Fig 2.2.4: crowded people in restaurants

Fig 5.2.5: crowded people in shopping malls

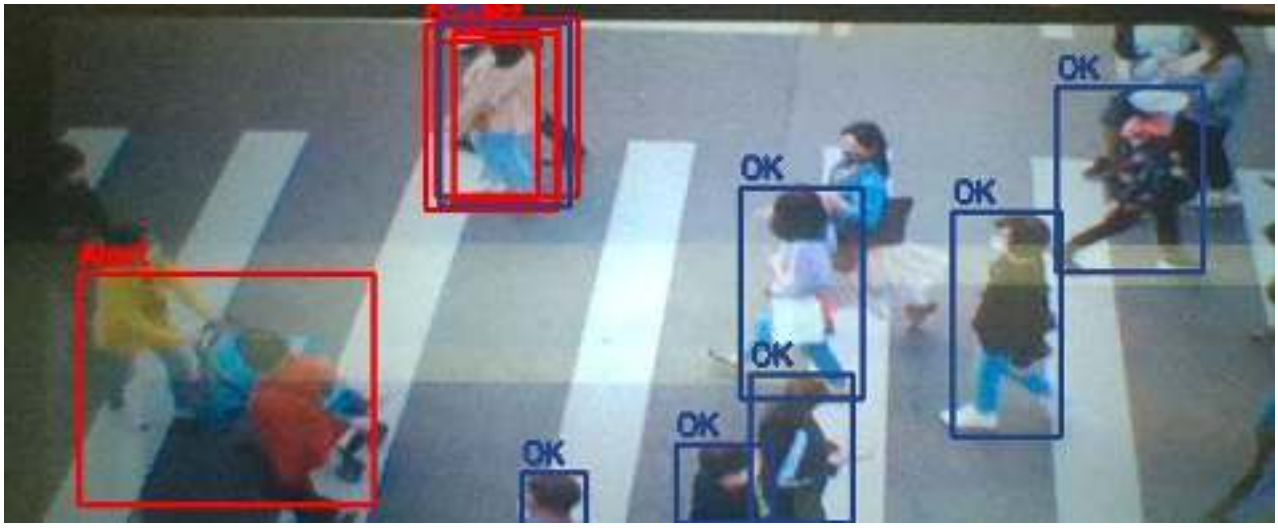

Fig 5.2.6: crowded people on roadside

6. TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be exercised.
Output	: identified classes of application outputs must be exercised.
Systems/Procedures	: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.2.4 SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6.2.5 WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

6.2.6 BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

6.2.7 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

- **Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

6.3 TEST CASES

TEST CASE ID	TEST CASE NAME	DESCRIPTION	INPUT	EXPECTED OUPUT	ACTUAL OUTPUT	RESULT
01	DISTANCE DETECTION 1	TO CHECK SOCIAL DISTANCE BETWEEN INDIVIDUALS	CROWDED PEOLPE	VIOLATING RULE	MAINTAINING SOCIAL DISTANCE	FAIL
02	DISTANCE DETECTION 2	TO CHECK SOCIAL DISTANCE BETWEEN INDIVIDUALS	PEOPLE WITH DISTANCE	MAINTAINING SOCIAL DISTANCE	MAINTAINING SOCIAL DISTANCE	PASS
03	DISTANCE DETECTION 3	TO CHECK SOCIAL DISTANCE BETWEEN INDIVIDUALS	PEOPLE WITH DISTANCE	MAINTAINING SOCIAL DISTANCE	VIOLATING RULE	FAIL
04	DISTANCE DETECTION 4	TO CHECK SOCIAL DISTANCE BETWEEN INDIVIDUALS	CROWDED PEOPLE	VIOLATING RULE	VIOLATING RULE	PASS

7. CONCLUSION

7. CONCLUSION

7.1 PROJECT CONCLUSION

In this project we have used a recent technique in the field of computer vision and also in the deep learning. The proposed system will correctly detect the presence of person is in the safe distance. The system is accurate, since we have used Euclidean distance formula for distance computing. Thus, it makes easier to deploy our model to embedded system like Raspberry Pi, Google Coral etc. We believe that this approach will enlarge the safety of the individuals during the pandemic. Social distance with is performed by using the two labels which were declared for color and title.

- System uses a pre-filmed video of people on a crowded street. The distance between each person can be easily calculated using computer vision. If any set of individuals is found violating the minimum accepted threshold value, they will be indicated with a red bounding box.
- The social distancing patterns are distinguished and classified as "Safe" and "Unsafe" distance. The classifier can be implemented for live video streams and can be used for developing real-time applications. This system can be integrated with CCTV for surveillance of people during pandemics.

7.2 FUTURE ENHANCEMENT

- With advanced technical updates, the system may be capable to trace and detect the violations from an aerial view and we can implement this system in shopping malls, restaurants to find the violations along with distance marks.
- We can predict/detect time at which it gets crowded and heat map can be plotted in an accurate way.
- This publishing is focusing on surveillance of public places and detecting whether the people are maintaining social distancing or not. Social Distancing is the only best option for us to protect ourselves from diseases, not limited to COVID-19, where no medicinal antidote has been prepared, and that may be transmitted through human contact.

8. BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 REFERENCES

- S. Yadav, Deep learning based safe social distancing and face mask detection in public areas for covid-19 safety guidelines adherence
- F. Sener, N. Ikizler-Cinbis, Two-person interaction recognition via spatial multiple instance embedding, Journal of Visual Communication and Image Representation 32, 63 (2015)
- W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A.C. Berg, in European conference on computer vision (Springer, 2016), pp. 21–37
- Yang D. Yurtsever E. Renganathan V. Redmill K.A. Özgüner Ü. A vision-based social distancing and critical density detection system for COVID-19 2020 arXiv:2007.03578
- Pouw C.A. Toschi F. van Schadewijk F. Corbetta A. Monitoring physical distancing for crowd management: Real-time trajectory and group analysis 2020 arXiv:2007.06962
- K. He X. Zhang S. Ren and J. Sun "Deep residual learning for image recognition" 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 770-778 2016.
- K. Li G. Ding and H. Wang "L-fcn: A lightweight fully convolutional network for biomedical semantic segmentation" 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) pp. 2363-2367 Dec 2018.
- S. Ge, J. Li, Q. Ye and Z. Luo, "Detecting Masked Faces in the Wild with LLE-CNNs," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 426-434, doi: 10.1109/CVPR.2017.53.
- Dr. S. Syed Ameer Abbas, Dr. P. Oliver Jayaprakash, M. Anitha, X. Vinitha Jaini, “Crowd Detection and Management using Cascade classifier on ARMv8 and OpenCV-Python”, Mepco Schlenk Engineering College, Sivakasi, 2017 International Conference on Innovations in Information, Embedded and Communication systems (ICIIECS).

- Joel Joseph Joy, Manali Bhat, Namrata Verma, Milind Jani, "Traffic Management Through Image Processing and Fuzzy Logic", D.J. Sanghvi College of Engineering, Mumbai, India, Proceedings of the Second International Conference on Intelligent Computing and Control Systems (ICICCS 2018), IEEE Xplore Compliant Part Number: CFP18K74-ART; ISBN: 978-1-5386-2842-3.
- A. Haldorai and A. Ramu, Canonical Correlation Analysis Based Hyper Basis Feedforward Neural Network Classification for Urban Sustainability, Neural Processing Letters, Aug. 2020. doi:10.1007/s11063-020-10327-3
- Landing AI Creates an AI Tool to Help Customers Monitor Social Distancing in the Workplace [Onlive] (Access on 4 May 2020).
- Ahmed, I., Ahmad, M., Rodrigues, J. J. P. C., Jeon, G., & Din, S. (2020). A deep learning-based social distance monitoring framework for COVID-19. Sustainable Cities and Society, 102571. doi:10.1016/j.scs.2020.102571
- Dhaya, R. CCTV Surveillance for Unprecedented Violence and Traffic Monitoring. Journal of Innovative Image Processing (JIIP) 2, no. 01 (2020): 25-34.
- Ramadass, Lalitha, Sushanth Arunachalam, and Z. Sagayasree. Applying deep learning algorithms to maintain social distance in public places through drone technology. International Journal of Pervasive Computing and Communications (2020).
- Degadwala, Sheshang, et al. Visual Social Distance Alert System Using Computer Vision & Deep Learning. 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA). IEEE, 202

8.2 WEBSITES

- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7603992/
- https://ieeexplore.ieee.org/document/9673434
- https://pyimagesearch.com/2020/06/01/opencv-social-distancing-detector/

8.3 DRIVE LINK

- https://drive.google.com/drive/folders/18gnZmQb5c5aksDXI2VQm12nI9TJ8T-yV?usp=sharing