CROWD MONITORING SYSTEM USING IMAGE PROCESSING

At

Sathyabama Institute of Science and Technology(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering

By

K.Shivram

(Reg.No.38110527)

&

R.VenkataKrishnan

(Reg.No.38110638)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119, TAMILNADU

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI– 600119
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **K.Shivram (38110527) and R.VenkataKrishnan (Reg.No.38110638)** who carried out the project entitled "**Crowd Monitoring System Using Image Processing**" under my supervision from September 2021 to November 2021.

Internal Guide

Dr. P. JEYANTHI.,

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.,

Submitted for Viva voce Examination held on

Internal Examiner **External Examiner**

DECLARATION

K.Shivram and R.VenkataKrishnan hereby declare that the Project Reportentitled **CROWD MONITORING SYSTEM USING IMAGE PROCESSING** is done by me under the guidance of **DR.P.JEYANTHI,.M.E.ph.D,** Associate professor, Dept of CSE at **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degreein Computer Science and Engineering.

DATE:

PLACE:

SIGNATURE OF THECANDIDATE

ABSTRACT

A huge crowd of people visit public places seasonally. Crowd management is critical in these situations in order to avoid public disasters like stampede, confusion, social distance, danger of pandemic etc. Therefore, a need arose in order to manage these crowds to minimize the rate of pandemic and avoid spreading. This framework would be smart and efficient. The framework is based on computer vision with a fairly simple and easy to use interface. The framework is comprised of three layers: sensor, management, and interface layers. The sensor layer is responsible for gathering the data of number of people getting into a room. The management phase is responsible for checking whether the restricted count of a particular room is attained or not. The interface layer is responsible for alerting the security officials with a buzzer to convey that the restricted count has been reached and no more civilians are allowed inside the room. This setup is applicable in closed space like a room, shop, conference room, lift, etc. In wide area, we implement OpenCV concepts using python to measure the distance between each individual and warn them if they exceed the distance limit among the crowd. The assumed distance between two individuals is calculated based on the pixel position of a particular person in the scene. We use image processing to generate the result. If the limit among the individuals exceed, then it will show warning in red color, otherwise they are good to go.

Keywords: OpenCV

LIST OF FIGURES

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

Crowding is a common phenomenon observed during major events such as concerts, festivals, sports, games, and entertainment. One of the most interesting and active research topic in computer vision is the analysis of crowd behavior. Crowd is a group of people gathered in a certain location. Crowd differs in different situations like crowd in a temple will be different from the crowd in a shopping area. Crowd is a group of individuals sharing a common physical location. Now a day's increase in human population tends to increase the crowd in public areas [3].Thus it is required to analyze the surveillance system with several closed circuit Television is used to monitor the crowd. The human eye cannot observe all the cameras simultaneously. Thus an automated technique must be used for continuously monitoring the crowd for a long period. Challenging problems in detecting the desired events automatically are that, simultaneous occurrence of more than one events, large number of data must be processed, occlusions and real time detection. The proposed method can be applied from small group of object. The Internet of Things is a powerful industrial system of radio-frequency identification and wireless devices that have the ability to transfer data over a network without needing human interaction [1]. Analysis of a crowd behavior using surveillance videos is an important issue for public security, as it allows detection of dangerous crowds and where they are headed. [4] Computer vision based crowd analysis algorithms can be divided into three groups people counting, people tracking and crowd behavior analysis. Mainly, IoT consists of three layers, the sensing layer to gather data from real world via existing hardware e.g. sensors, next the network layer to transfer the collected data over wired or wireless network, and the application layer which is responsible for twoway communication between user and systems.

Chapter 2
LITERATURE REVIEW

S. Hashish, and M. Ahmed (2015). "Efficient wireless sensor network rings overlay for crowd management in Arafat area of Makkah." In 2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems, pp. 1-6. Researchers have started utilizing sensor enabled Smartphones as a tag for large scale human sensing. With the increasing usage of Smartphones, more persons can be tracked without providing any tag in future. Some of the Smartphone based location tracking systems require an application to be installed on the client's Smartphone. The installed application obtains the location using GPS sensor of Smartphone and continuously updates the location to the remote server using Internet connection. However, it is rare that users in the large crowd and in remote locations will have the Internet connection all the time. There are many operating systems and versions for Smartphones, which makes the development and distribution of application a difficult task. By embracing the Smart City paradigm, crowd sensing becomes a solution able to cope with air pollution monitoring since this novel paradigm assumes that a significant number of users perform collaborative sensing tasks, thereby collecting data from different populated locations while doing their daily activities. The collected data is periodically transmitted to a central server for data storage and processing.

D. C. Duives, T. van Oijen, and S. P. Hoogendoorn (2020), "Enhancing crowd monitoring system functionality through data fusion: Estimating flow rate from Wi-Fi traces and automated counting system data," IEEE Sensors, vol. 20, no. 21, p. 6032.

With the development of sensor technology, communication technology, and big data science, smart city-oriented intelligent applications have become important services in human life. People's living standard has increased with improved infrastructure and intelligent applications such as smart home furnishing, smart building. With the increasing expansion and prosperity of urban business zones, some people choose to go shopping and seek entertainment in business zones. These large central business zones have become representative of the city image and are the zones with the most economic vitality. Besides,of people. By estimating the number of people, the degree of crowd gathering can be judged for accurate and effective management and planning at a monitored site. Traditional crowd monitoring systems depend on vision-based monitoring technology, the most commonly used method of which is closed-circuit television monitoring. In other words, the signal can be transmitted from the data source to a prearranged specific broadcast device connected to the source.Data fusion is a kind of information fusion technology that associates, correlates, and combines the information from multiple sensor to obtain more timely and accurate decision making support. From low-level data collection to high level services, data fusion offers feasible and high-efficiency support for deep fusion and mining massive multi-source data in heterogeneous networks

Sebastian Bek and Eduardo Monari (2015), "The Crowd Congestion Level A New Measure For Risk Assessment In Video Based Crowd Monitoring", International Journal of Information Technology and Decision Making 3(4):2353- 2361.

There are more categories of crowd counting in video sensing, one is region of interest counting which estimates the total number of people that regions at an instance and the other is line of interest which counts number of people who crosses a detecting line in an instance. The line of interest counting can be developed using feature tracking techniques where they are either calculated into trajectories and these trajectories are clubbed into object tracks or based on counting crowd densities from a temporal frame of the video.People in a crowded scene should be regarded as endangered in case in a certain local region of the scene a too large number of people is observed. However, on one hand absolute density is very difficult to extract from crowd videos, and on the other hand density alone is not a sufficient characteristic figure, since for example in a concert or public festival, a very high density is usual and not critical per se. We observed that even in scenarios with high people density, the situation can be regarded as non-critical, in case people can still move freely and smoothly through the crowd. As a consequence, we believe that information on the flow dynamics should be taken into account for risk assessment. Our approach assumes, that a local spot in the crowd might be critical, if the density is continuously increasing over time, and simultaneously a significant reduction of motion dynamics is observed.

Rosario Fedele and Massimo Merenda (2017), "An IoT System for Social Distancing and Emergency Management in Smart Cities Using Multi-Sensor Data", In 2017 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems.

Recommender systems technology aims at reducing the consumer over-choice due to the huge amount of information available on the web. These systems use information gathered through few interactions with its user, to create a customized selection of items that can interest the user and can facilitate a better user experience. Recommender systems could be further improved if Machine Learning and Deep Learning algorithms were used for information retrieval. Noteworthy applications of recommender systems in smart cities include, a mobile IoT recommender system for users that need to find Park-and-Ride infrastructures to switch from a private to a public transportation mode, an autonomous situation-aware evacuation route recommenderarchitecture, optimized in real time, to obtain smart buildings, reuse data generated by various IoT applications, and adapt the services provided by the single IoT system, to improve the user experience. Traditionally, the problems above can be solved using

Shortest Path Finding algorithms that belong to the mathematical field of Combinatorial Optimization. Noteworthy examples are, Prim's algorithm, which can be used to derive the Minimum Spanning Tree and the Dijkstra's and the A* algorithms, which allow one to derive the shortest path between two points. In the Matlab environment, it is possible to find different tools that allow one to solve this problem.

Marwa F. Mohamed, Abd El Rahman Shabayek, Mahmoud ElGayyar (2018), "IoT-based Framework for Crowd Management", Computer Science Department, Faculty of Computes and Informatics Suez Canal University, Ismailia, Egypt, IEEE International Conference on IoT.

The Internet of Things is a powerful industrial system of radio-frequency identificationand wireless devices that have the ability to transfer data over a network without needing human interaction. Mainly, IoT consists of three layers, the sensing layer to gather data from real world via existing hardware, the network layer to transfer the collected data over wired or wireless network, and the application layer which is responsible for two way communication between user and systems. IoT applications are rapidly evolving and growing in various fields. Sensing the crowd by normal sensors and managing it is a challenging problem. Crowd management requires several stages including crowd data acquisition via sensor layers, data transferring via network layers, data analysis for decision making and applying crowd control measures via application layer. Using different sensor devices, it is possible to gather information about visitor's crowds and determine which areas are overcrowded. This information is then transmitted to the management layer where the admin can decide to close some doors and roads. The management layer shall be equipped by a smart service that can recommend which doors to be opened or closed to direct the crowd flow. The admins then decide whether to publish this information to the public or not. In order to increase the user end application usability, the application provides different languages and auser friendly interface.

Chapter-3

1.2 Aim of the Project

The aim of this project is to detect and react to overcrowding issues at an early stage. This will help us to assess how effective our crowd safety precaution is and the considerations which are looked into are as follows, monitoring the overall number of people to ensure the safe venue capacity is not exceeded, monitoring the distribution of people to help prevent local overcrowding, identifying potential crowd problems to prevent the escalation of public disorder.

1.2 Project Domain

Our project come under the domain of Computer Vision where we have used both technologies to identify overcrowded areas. The Internet of things describes the network of physical objects that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet.

Computer vision is an interdisciplinary field that deals with how computers can be made to gain high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do. As a technological discipline, computer vision seeks to apply its theories and models for the construction of computer vision systems.

1.4 Scope of the Project

With booming in population and huge number of crowds converging at common places for meetings, gatherings, and reunions there is increased scope for these kind of projects. Our approach of identifying crowd in open space will be definitely useful in controlling the crowds in the public area, that too in this pandemic world crisis the importance of these kinds of projects have created necessary need among the governments of various countries to monitor the public crowd.

1.5 Methodology

Image Processing Method

This method is software oriented approach which is used in open space i.e. malls, public areas and others, we used the technology of computer vision to detect the people with the help of cameras in public area, the input to our model is directly taken from the real time monitoring cameras installed in public areas where the model will be installed in the system and the camera video is taken as input, rectangles are used to highlight the crowd violations where green rectangle insist no violations whereas red rectangle insist that there is a crowd violation in a particular area, the crowded area is identified with the help of pixel positions of the people in the video.

3.4 System Specification

3.4.1 Hardware Specification

- LCD Display Panel
- CCTV Camera
- Monitor
- Graphics card
- Peripheral Devices

3.4.2 Software Specification

- Python Programming Language
- ScipyLibrary
- NumpyLibrary
- Imutils Library
- Opencv-python Library
- Operating System - Windows

- ❖ YOLO Object Detection Library

3.4.3 Standards and Policies

The various standard versions of software which need to be downloaded and installed are as follows,

Python - 3.6.X

Scipy - 1.4.1

Numpy- 1.18.5

Imutils- 0.5.3

Opencv-python - 4.2.0.34

Windows Operating System - 8.1 or higher

Graphics Card - Intel Graphics 10.18.10.4252 or higher

Monitor - 1366 X 768 display or higher

Communication Device - Mobile or Waketalkie

Download the python from WWW and install it, to install python libraries use the command “pip install library-name version”, Configure the Adruino board with battery sensor and buzzer, connect the Adruino board to the computer, commit the necessary code to Adruino with the help of cable.

Chapter 4

MODULE DESCRIPTION

4.1 General Architecture

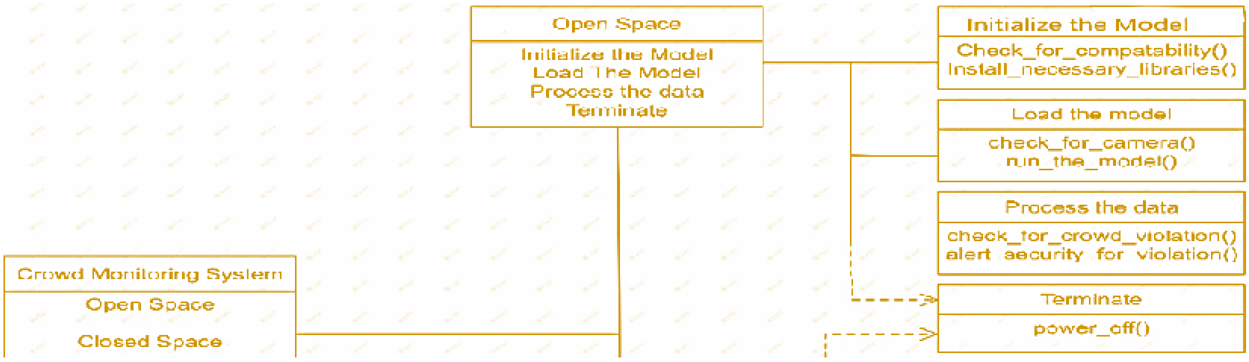

Figure 4.1: Architecture Diagram

The architecture of our project consist of a is closed place where the architecture explains about monitoring crowds in rooms and conference halls.

4.2 Design Phase

WORKFLOW DIAGRAM:

The above is the entity relationship diagram for our proposed model. Where the relation between crowd monitoring system and crowd is a n to n relationship, the relationship between open space and closed space is n to n relationship, the relationship between terminate on open space and closed space is a one to one relationship as the machine can be terminated only once that is it can be switched off only once to switch off it again we need to first power it on and then terminate so one to one relationship.

4.3 Module Description

MODULE:

Developing an Image Processing model for monitoring crowd in open space. This is a software kind of approach where we have used computer vision to identify crowds, it requires a minimum specification computer to run the proposed model, the description and required specification for the computer are already discussed in the hardware component section.

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

The input design for the module is the video captured from the real time CCTV camera placed in any public place for which we need to monitor the crowd and make people to spread across the area instead of accumulating one area.

5.1.2 Output Design

The output design for this module is displayed in the same monitor in which the video from CCTV camera is used as input it displays the number of crowd violations in the monitor screen and also differentiates the people in the scene those who are not violating the crowd by enclosing them in a blue rectangular box, for those who violated the crowd it displays them in yellow color rectangular box.

5.2 Testing

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product isdefect free. It involves execution of software or system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

5.3 Types of Testing

5.3.1 Unit Testing

The unit testing is done on major important components of the proposed model in our IoT design, working of it is checked and verified then the sensors where its working is tested and verified then the battery to power up adruino and sensor is tested and verified, now the components related to software working of the model are checked and verified the camera used is tested for its working and verified then here to run the python code for crowd detection we have used python which is executed in the local machine to which the CCTV camera is connected so its setup is verified.

5.3.2 Functional Testing

The functioning for which the components are being used is tested in this phase whether the desired result are produced by the components. The sensor is tested along with the adruino whether it counts the people correctly and buzzers the alarm when the restricted count is reached. The python code is executed on the input video from the CCTV camera to check whether it identifies crowd in public places and distinguishes them with green and blue rectangular boxes.

5.3.3 Integration Testing

The combined working of the components is tested and verified in this testing. The adruino along with the sensor and battery is tested for its working collaboratively with each other getting attached. The camera and the code are tested along together whether the image from camera is used correctly for detecting crowd.

5.3.4 White Box Testing

For checking whether the data is available in the desired folder we need to specify the correct path for it to detect the crowd from the CCTV camera available in public places and malls for that we need to specify the inputs for it assume the expected results you need the program to produce should be done beforehand for better results.

5.3.5 Black Box Testing

This testing is done without the need of developer, the person who doesn’t know anything about developing he just checks whether the expected outputs are produced for the given inputs. For this test give a few minutes recorded video as input and test whether the program identifies the crowds in the video and alerts about it.

5.4 Testing Strategy

The strategy we used to test our proposed model was, first to test the components in closed space where we checked for adruino, sensor, buzzer integration working on producing the correct the result in the LCD screen and also the buzzer alarming at correct time, then for closed space we started with the specifications of the computer first then we installed our model in the system and also supporting libraries required for our model to monitor crowd, then we checked for the real time CCTV camera placed in public places, as we didn't have access to real time camera we checked our proposed model with already prerecorded video of format mp4 and it monitored and displayed the crowd violated area correctly.

Chapter 5

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

This project typically is a less cost effective and more economically benefitable to the organization which builds this project and also to the society considering the current undergoing pandemic situation. The analysis of this project ensures in helping government organizations and other NGO to determine their viability, cost, and benefits associated with this project before financial resources are allocated to help the needy. So from a technical point of view we can assure that the idea can be converted to working system with the already ready available technologies and hardware systems available to the organization. It also serves as an independent project assessment and enhances project credibility helping decision makers determine the positive economic benefits to the organization that the proposed project will provide.Social impact analysis of our project greatly reduces the overall risks of being inhuman to the surrounding and its people and also helps to reduce resistance, strengthens support of the people towards the government organization, and allows for a more comprehensive understanding of the costs and benefits of the project.

Output

Figure 6.2: Output 1

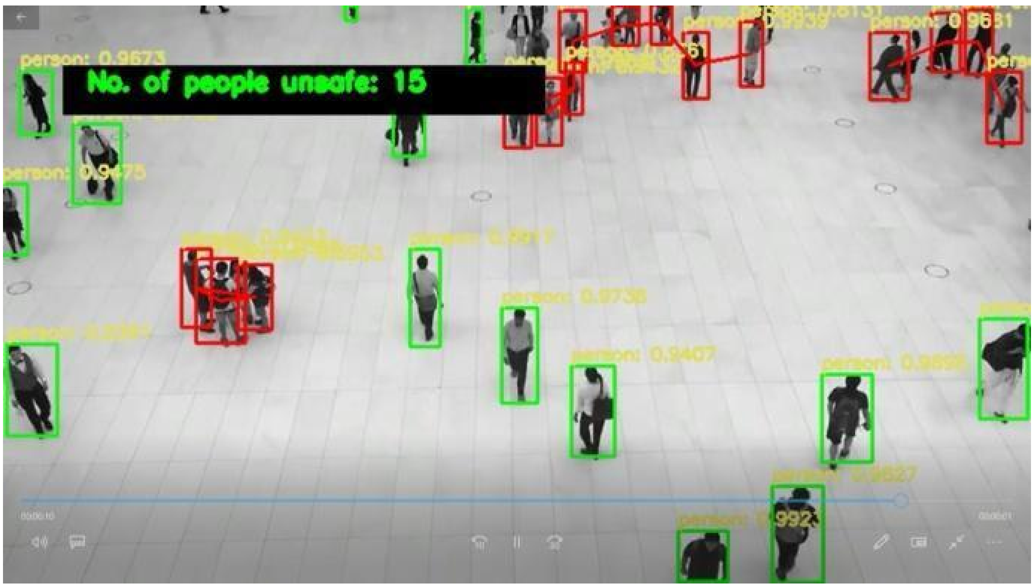

Figure 6.3: Output 2

Chapter 6

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

We have discussed the crowd management framework for spreading visitors over a crowded area. Sensors layer aims to track visitors and gather information about visitor's crowd. Management layer aims to analyze the collected data and extract the required information about the visitors upon administrative acceptance. The interface layer provides an application that aims to help administrators and visitors to avoid crowd disasters as it informs them about the count of people in a particular area and overcrowd area. Also, the proposed framework shall be usable irrespective of the age and gender if the person knows how to operate a computer in case of open space proposed model or a one know how to switch on or off electric circuit can operate our proposed model in closed space. In wide area, we implement computer vision using python to measure the distance between each individual and treat them as cause of crowd violation if they didn't follow the distance limit among each person in the crowd.The proposed framework will effectively save time and efforts to help administrators controlling and distributing visitors.

7.2 Future Enhancements

We have discussed two aspects of crowd monitoring in this report. The most notable point is the exceptional research on vision based crowd monitoring techniques.In order to address the privacy laws, focus needs to be shifted to less-intrusive and privacy preserving crowd monitoring techniques. Both, vision and non-vision techniques can compromise the privacy of an individual, either by facial recognition. While security oriented technologies might have permissions to identify and track individuals, general crowd monitoring techniques need to focus more on intrusion and privacy.Crowd monitoring is an important research area and in order to develop effective technologies, it is necessary to test the newly proposed techniques in different crowd scenarios. However, the possibility of getting real life scenarios for testing is either limited or biased with assumptions. Emulation software like Autodesk, can be really useful to evaluate the performance of developed techniques under such conditions.As part of our future work, we will deploy Crowd Track system for large scale environments, such as those occurring in disaster management scenarios. Also, we plan to perform user studies with safety and security personnel, to evaluate the applicability of the proposed measures in practice.

SOURCE CODE

9.1 Source Code

```
from scipy.spatialimport distance as dist
import numpyas np
import imutils
import cv2

# import the necessary packages

# initialize minimum probability to filter weak detections along with
# the threshold when applying non-maxima suppression
MIN_CONF = 0.3
NMS_THRESH = 0.3

# define the minimum safe distance (in pixels) that two people can be
# from each other
MIN_DISTANCE = 40


# Detect people function
def detect_people(frame, net, ln, personIdx=0):
# grab the dimensions of the frame and  initialize the list of
    # results
(H, W) = frame.shape[:2]
    results = []
```

```
# construct a blob from the input frame and then perform a forward
    # pass of the YOLO object detector, giving us our bounding boxes
    # and associated probabilities
blob = cv2.dnn.blobFromImage(frame, 1 / 255.0, (416, 416), swapRB=True,
crop=False)
net.setInput(blob)
layerOutputs = net.forward(ln)

# initialize our lists of detected bounding boxes, centroids, and
    # confidences, respectively
boxes = []
    centroids = []
    confidences = []

# loop over each of the layer outputs
for output in layerOutputs:
# loop over each of the detections
for detection in output:
# extract the class ID and confidence (i.e., probability)
            # of the current object detection
scores = detection[5:]
classID = np.argmax(scores)
            confidence = scores[classID]

# filter detections by (1) ensuring that the object
            # detected was a person and (2) that the minimum
            # confidence is met
if classID == personIdxand confidence > MIN_CONF:
# scale the bounding box coordinates back relative to
                # the size of the image, keeping in mind that YOLO
                # actually returns the center (x, y)-coordinates of
                # the bounding box followed by the boxes' width and
                # height
box = detection[0:4] * np.array([W, H, W, H])
                (centerX, centerY, width, height) = box.astype("int")

# use the center (x, y)-coordinates to derive the top
                # and and left corner of the bounding box
x = int(centerX - (width / 2))
                y = int(centerY - (height / 2))

# update our list of bounding box coordinates,
                # centroids, and confidences
boxes.append([x, y, int(width), int(height)])
centroids.append((centerX, centerY))
confidences.append(float(confidence))

# apply non-maxima suppression to suppress weak, overlapping
    # bounding boxes
idxs = cv2.dnn.NMSBoxes(boxes, confidences, MIN_CONF, NMS_THRESH)

# ensure at least one detection exists
if len(idxs) >0:
# loop over the indexes we are keeping
for iin idxs.flatten():
# extract the bounding box coordinates
(x, y) = (boxes[i][0], boxes[i][1])
            (w, h) = (boxes[i][2], boxes[i][3])

# update our results list to consist of the person
            # prediction probability, bounding box coordinates,
            # and the centroid
```

```
r = (confidences[i], (x, y, x + w, y + h), centroids[i])
results.append(r)

# return the list of results
return results


# load the COCO class labels our YOLO model was trained on
labelsPath = "D:\\final yr
project\\Single-Multiple-Custom-Object-Detection-and-Tracking\data\\labels\\coco.n
ames"
LABELS = open(labelsPath).read().strip().split("\n")

# derive the paths to the YOLO weights and model configuration
weightsPath = "D:\\final yr
project\\Single-Multiple-Custom-Object-Detection-and-Tracking\\yolov3_tf2\\yolov3.
weights"
configPath = "D:\\final yr
project\\Single-Multiple-Custom-Object-Detection-and-Tracking\\yolov3.cfg"

# load our YOLO object detector trained on COCO dataset (80 classes)
print("[INFO] loading YOLO from disk...")
net = cv2.dnn.readNetFromDarknet(configPath, weightsPath)

# determine only the output layer names that we need from YOLO
ln = net.getLayerNames()
ln = [ln[i - 1] for iin net.getUnconnectedOutLayers()]

# initialize the video stream and pointer to output video file
print("[INFO] accessing video stream...")
# vs = cv2.VideoCapture(args["input"] if args["input"] else 0)
vs = cv2.VideoCapture("crowd_monitoring_sample.mp4")

# loop over the frames from the video stream
while True:
# read the next frame from the file
(grabbed, frame) = vs.read()

# if the frame was not grabbed, then we have reached the end
    # of the stream
if not grabbed:
break

# resize the frame and then detect people (and only people) in it
frame = imutils.resize(frame, width=900)
    results = detect_people(frame, net, ln, personIdx=LABELS.index("person"))

# initialize the set of indexes that violate the minimum social
    # distance
violate = set()

# ensure there are at least two people detections (required in
    # order to compute our pairwise distance maps)
if len(results) >= 2:
# extract all centroids from the results and compute the
        # Euclidean distances between all pairs of the centroids
centroids = np.array([r[2] for r in results])
        D = dist.cdist(centroids, centroids, metric="euclidean")

# loop over the upper triangular of the distance matrix
for iin range(0, D.shape[0]):
for j in range(i + 1, D.shape[1]):
```

```
# check to see if the distance between any two
                # centroid pairs is less than the configured number
                # of pixels
if D[i, j] < MIN_DISTANCE:
# update our violation set with the indexes of
                    # the centroid pairs
violate.add(i)
violate.add(j)

# loop over the results
for (i, (prob, bbox, centroid)) in enumerate(results):
# extract the bounding box and centroid coordinates, then
        # initialize the color of the annotation
(startX, startY, endX, endY) = bbox
        (cX, cY) = centroid
        color = (0, 255, 0)

# if the index pair exists within the violation set, then
        # update the color
if iin violate:
            color = (0, 0, 255)

# draw (1) a bounding box around the person and (2) the
        # centroid coordinates of the person,
cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)
        cv2.circle(frame, (cX, cY), 5, color, 1)

# draw the total number of social distancing violations on the
    # output frame
text = "Crowded areas identified till now: {}".format(int(len(violate) / 2))
    cv2.putText(frame, text, (10, frame.shape[0] - 25), cv2.FONT_HERSHEY_SIMPLEX,
0.85, (0, 0, 255), 3)

# output frame should be displayed to our screen
cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) &0xFF
# if the `q` key is pressed, break from the loop
if key == ord("q"):
break
```

References

[1] S. Hashish, and M. Ahmed (2015). "Efficient wireless sensor network rings overlay for crowd management in Arafat area of Makkah." In 2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems, pp. 1-6.

[2] D. C. Duives, T. van Oijen, and S. P. Hoogendoorn (2020), "Enhancing crowd monitoring system functionality through data fusion: Estimating flow rate from Wi-Fi traces and automated counting system data," IEEE Sensors, vol. 20, no. 21, p. 6032.

[3] Sebastian Bek and Eduardo Monari (2015), “The Crowd Congestion Level A New Measure For Risk Assessment In Video Based Crowd Monitoring”, International Journal of Information Technology and Decision Making 3(4):2353- 2361.

[4] Rosario Fedele and Massimo Merenda (2017), “An IoT System for Social Distancing and Emergency Management in Smart Cities Using Multi-Sensor Data”, In 2017 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems.

[5] Marwa F. Mohamed, Abd El Rahman Shabayek, Mahmoud ElGayyar (2018), “IoT-based Framework for Crowd Management”, Computer Science Department, Faculty of Computes and Informatics Suez Canal University, Ismailia, Egypt, IEEE International Conference on IoT. **[15] Jianjie Yang, Jin Li, Ye He,**“Crowd Density and Counting Estimation Based on Image Textural Feature”, JOURNAL OF MULTIMEDIA, VOL. 9, NO. 10, OCTOBER 2014.

[16] Aditya vora, Vinay chilaka,“fchd: fast and accurate head detection in crowded scenes”, arxiv:1809.08766v3 [cs.cv] 5 may 2019

[17] Sheng-Fuu Lin, Member, IEEE, Jaw-Yeh Chen, and Hung-Xin Chao, “Estimation of Number of People in Crowded Scenes Using Perspective Transformation” ieee transactions on systems, man, and cybernetics—part a: systems and humans, vol. 31, no. 6, november 2001.

[18] Mikel Rodriguez, Ivan Laptev, Josef Sivic, Jean-Yves Audibert, “Density-aware person detection and tracking in crowds. Proceedings of the IEEE International Conference on Computer Vision (2011)”.
