Web Application to Monitor and Alert Social Distancing Violations using Object Detection

Madineni Jitin Krishna[1] Ayyagari Aditya[2] Mummoju Aravindh Sai[3]
Gunthala Pradyumna Reddy[4] Tankala Praveen[5]
[1,2,3,4,5]Department of Computer Science and Engineering
[1,2,3,4,5]Vidya Jyothi Institute of Technology, Hyderabad, Telangana, India

Abstract — Web Application to Monitor and Alert Social Distancing Violations using Object Detection is an application which uses Machine Learning and Deep Learning concepts integrated with a web application. It is used to identify and monitor social distancing violations in inputs including but not limited to videos and webcam/CCTV footage. It also generates an alarm when the social distancing violations are detected along with statistics of said violations. It also sends an email notification to the administrative team when the violations exceed a threshold value. During pandemic like situations, to prevent the spread of disease in various places and alert people to maintain social distancing, this application could be indispensable in playing a vital role in reducing transmissions.

Keywords: Machine Learning, OpenCV, YOLOV3, SciPy, PyGame, Flask, Social Distancing Detection, Deep Learning, Computer Vision

I. INTRODUCTION

After the Covid-19 outbreak, it is imperative for us to wear a mask and follow social distancing to protect ourselves and others. Social distancing stops the spread of the virus as it puts distance between individuals. Even if an infected person just wears a mask in a relatively populated area, the spread of the virus significantly reduces and it also helps in staying away from infected people so as to limit the spread of virus. The main objective of this application is to detect the social distancing violations to prevent infections. So, our project detects the social distancing violations and generates an alarm to alert pedestrians who violate the guidelines. In this way, the person who is violating the social distancing protocols will be alerted and could take the necessary steps to prevent infecting others. This application provides a user-friendly interface while also tackling a few of the edge case scenarios of existing research work by either providing a solution or an extended analysis.

II. RELATED WORK

There are various articles, journals and other publications which already worked on a solution for this. But they lack in generating an alarm or notifying the administrative team by email when violations exceed threshold value to alert all the involved entities These systems are not integrated to alert users on field through a siren for example. The methods in various other publications does not have a user interface too and the output is viewed on terminal or console window.

III. PROPOSED SYSTEM

The proposed system is very much attainable and usable. The proposed system generates an alarm immediately whenever any violation is detected. Also, an email notification is sent to the administrative/security teams when violations continue for a certain period of time. On hearing the alarm, people could follow caution. Also, the model is integrated with a web application which serves also as the user interface. Through this, it would be easy for the user to run and manage the application effectively.

IV. SYSTEM DESIGN

Input device captures the video. YOLO object detection detects people. Then, the model calculates the social distancing between people and detects violations.

A. UML Diagrams

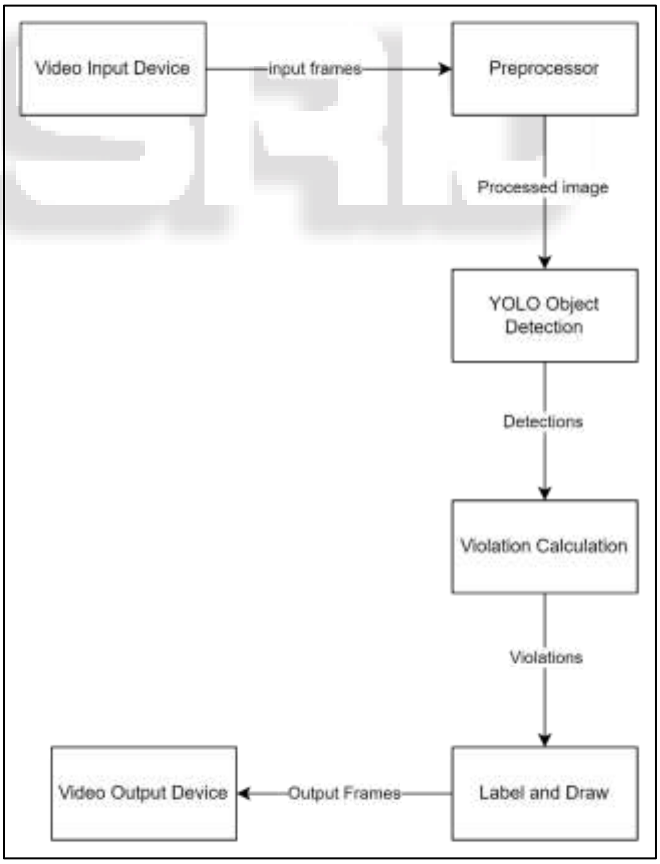

Fig. 1: Data Flow Diagram for Social Distancing Detection

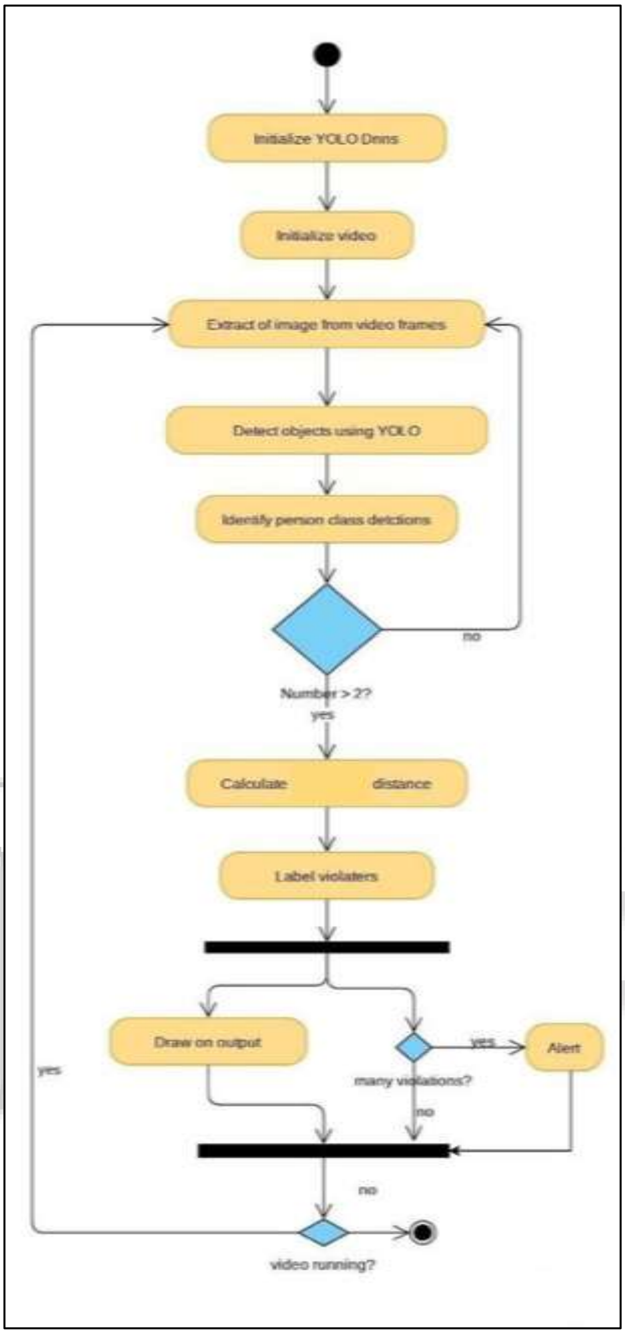

Fig. 2: Activity Diagram for Social Distancing Detection

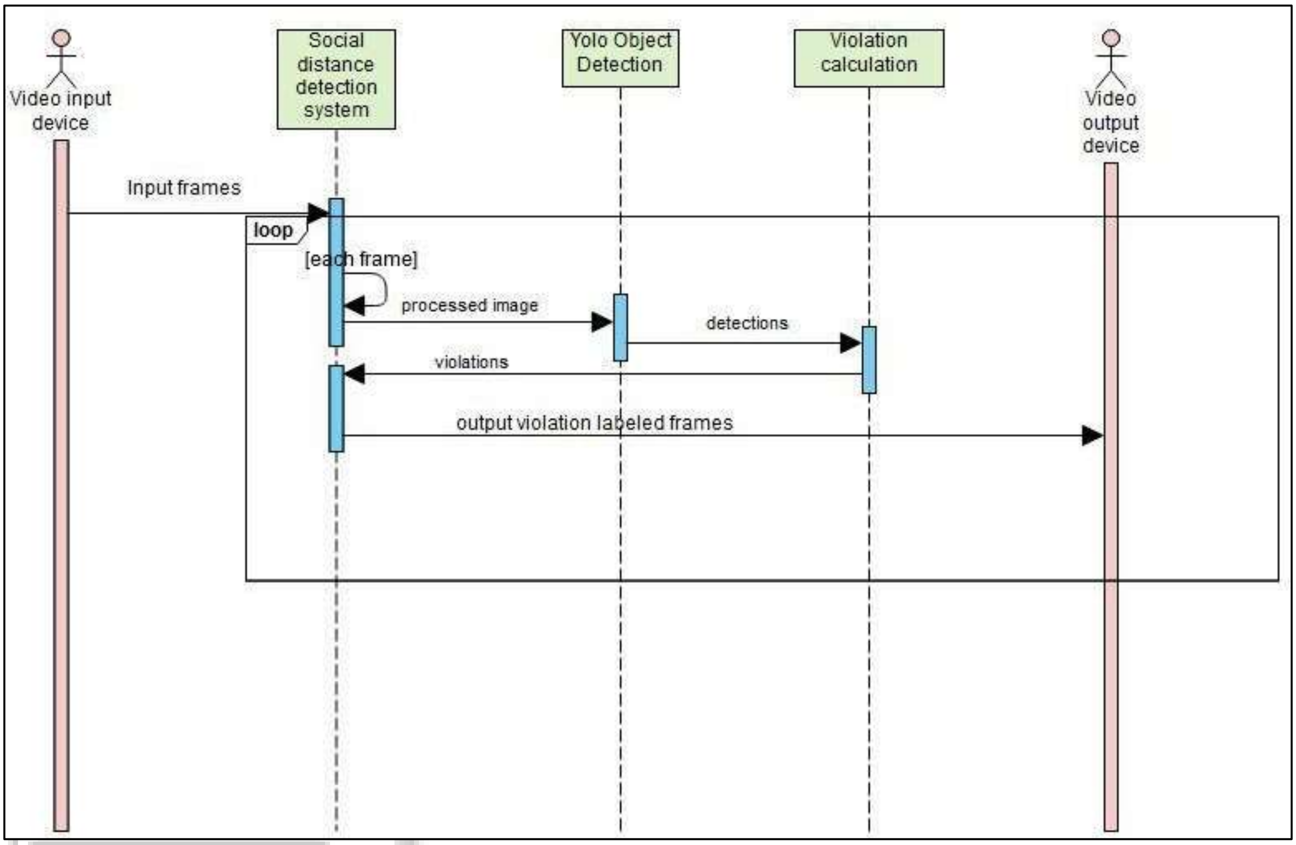

Fig. 3: Sequence Diagram for Social Distancing Detection

V. System Requirements

- Software: Anaconda: Jupyter Notebook, Python 3, Scikit-Learn, Numpy
- Editor: VS Code, PyCharm
- GPU Drivers: Any GPU Driver depending upon system configurations
- GPU: Any Graphics Processor with minimum 8GB dedicated Memory
- Camera: CCTV/ Webcam
- Storage Disk: 100GB HDD or more

VI. Model Description

A. Social Distance Detection

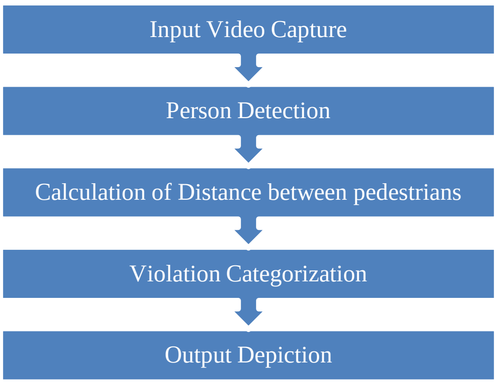

Fig. 4: Steps involved in Social Distance Detection

The system uses the YOLOv3 model to detect people from the frame of the input video. After detecting people, the distance is calculated between them. Then it identifies violations by comparing the distance between two individuals against a given threshold value.

This social distancing detection tool was developed to detect the safe distance between people in public spaces. The deep CNN method and computer vision techniques are employed in this work [3]. Initially, an open-source object detection network based on the YOLOv3 algorithm was used to detect the objects in the video frame. From the detection result, only the person class is used and other object classes are ignored in this application. Subsequently, the bounding box that fits best for each detected person can be drawn in the image, and this data of detected persons will be used for the distance measurement.

For camera setup, the camera is captured at a fixed angle as the video frame is treated as a perspective view (visual output frame from the top) and transformed into a two-dimensional top-down view for more accurate estimation of distance measurement. In this methodology, it is assumed that the people in the video frame are walking on the same flat plane. The location for each person can be estimated based on the top-down view. The distance between persons can be measured and scaled. Depending on the preset minimum distance, any distance less than the acceptable distance between any two individuals will be indicated with red bounding boxes around them, else green bounding boxes can be seen.

Different types of input devices that can be used for parsing video:

1) IP address of CC TV camera can be used to fetch real time video.
2) 0 can be used in VideoCapture method argument (example, cv2.VideoCapture(0)) to fetch video from web camera.
3) Video file path can be used to parse existing video.

B. Person Detection

When it comes to deep learning-based object detection, the YOLO model is considered one of the state-of-the art object detectors which can be demonstrated to provide significant speed advantages suitable for real-time application. The YOLO algorithm was considered for object detection taking a given input image and simultaneously learning bounding box coordinates, object confidence and corresponding class label probabilities. It is trained on the COCO dataset which consists of 80 labels including a person class. In this work, box coordinates, object confidence and person object class from detection result in the YOLO model were used for person detection.

The region of interest (ROI) of an image focused on the person walking the street is transformed into a 2D view that contains 416×416 pixels. Then, camera view calibration is applied which works by computing the transformation of the perspective view into a top-down view. In OpenCV, the perspective transformation is a simple camera calibration method which involves selecting four points in the perspective view and mapping them to the corners of a rectangle in the 2D image view. Hence, every person is assumed to be standing on the same level flat plane. The actual distance between people corresponds to the number of pixels in the top-down view and can be estimated. A rectangular box is drawn around the person for all pedestrians in the image frame. When multiple boxes of pedestrians are over-lapped, we select a single box out of multiple overlapping boxes. For this, we can use the Non-Maximum Suppression (NMS) algorithm to locate the pedestrian correctly. NMS selects the bounding box which has highest objectiveness score. Using OpenCV, we can use: cv2.dnn.NMSBoxes().

C. Distance Calculation

In this step of the pipeline, the location of the bounding box for each person in the perspective view is detected. For each pedestrian, the position in the top-down view is estimated based on the bottom-center point of the bounding box. The distance between every pedestrian pair can be computed from the top-down view and the distance is scaled by the scaling factor estimated from camera view calibration. Then, the centroid of the bounding box (rectangle) is calculated. This centroid denotes position of a person on the image. We can calculate the distance between the camera and the object "d" as:

$$d = (W * F)/P$$

d - distance between camera and object
W - width of the object in pixels
F - focal length of the camera
P - width covered by object on average in photo frame (apparent width in pixels)

Now that we can calculate the distance between person and camera, we should now compare the distance between two people. There are various ways to find distance between 2 points or 2 vectors on the plane. Given the position of two persons in an image as (X1, Y1) and (X2, Y2) respectively, the distance between them can be calculated using multiple mathematical models. Two of which are demonstrated below:

$$\text{Euclidean Distance } (d1) = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$

$$\text{Manhattan Distance } (d2) = |x1 - x2| + |y1 - y2|$$

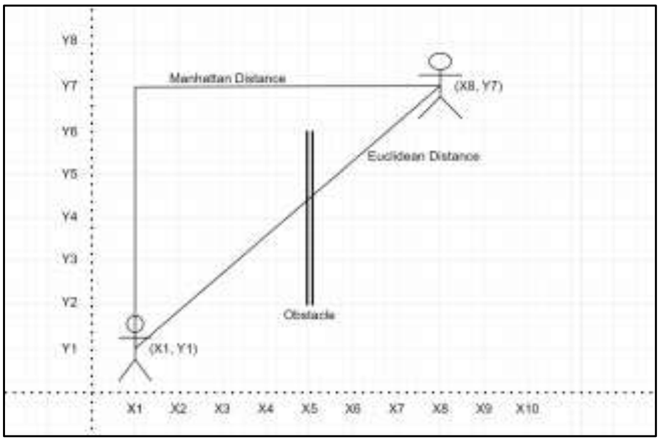

Fig. 5: Depiction of Euclidean and Manhattan Distance between people in a coordinate plane

For our use case, we are using the Euclidean distance as it gives the shortest distance between 2 points.

This distance is calculated between all the centroids generated for the entire input video. Then, we can proceed with violation identification as follows. If distance <= threshold, then display red colored box around pedestrian and increment count value, else display green box around the pedestrian. If count is greater than 0, ring the alarm to alert the pedestrians to follow social distancing protocols. Multiple cases and levels of violations can be added.

1) Alerting on field pedestrians using alarm during violations:

When violations are detected, use mixer module from pygame to play the audio which consists of alert message.

```
pygame.mixer.init()
pygame.mixer.music.load('alert.wav')
pygame.mixer.music.play()
```

When number of violations become zero, stop playing the audio.

```
Pygame.mixer.music.stop()
```

Code Snippet

2) Integration with email:

If the number of social distancing violations doesn't become 0 after detecting violations within certain time (which the administrators can set), or if the violations exceed the threshold value, then an alert should be sent to the nearest security team or responsible team within the radius of camera location.

Steps to integrate:

1) Use smtplib module.
2) Connect to gmail using SSL. server = smtplib.SMTP_SSL('smtp.gmail.com', PORT_NUMBER)
3) Use server.login(sender_email, password) to login. Check for 2 factor authentication.

4) Create a message with alert text and subject. Use server.sendmail(sender_email, receiver_email, message) method to send the email.
5) Terminate the SMTP session and close the connection using sever.quit().

D. Home Page

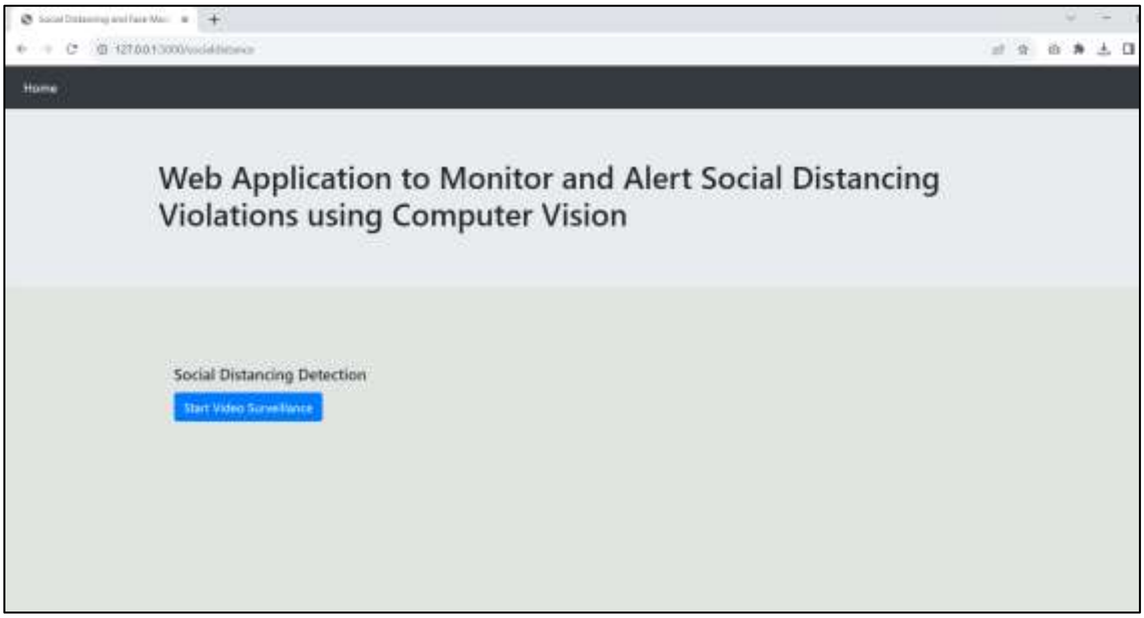

Fig. 6: User Interface of the Web Application

This is the user interface page where violations of social distancing can be monitored from. This is built using the Flask web framework where the look and feel of the UI is further improved with CSS and Bootstrap. Through this webpage, we can access two methods - home method and socialDistancing method. The "home_method" is used to display the content on the browser. The socialDistancing method, which is invoked on clicking the "Start Video Surveillance" button, is used to start the detection of social distancing violations.

VII. EXPERIMENTAL RESULTS AND OUTPUT SCREENS

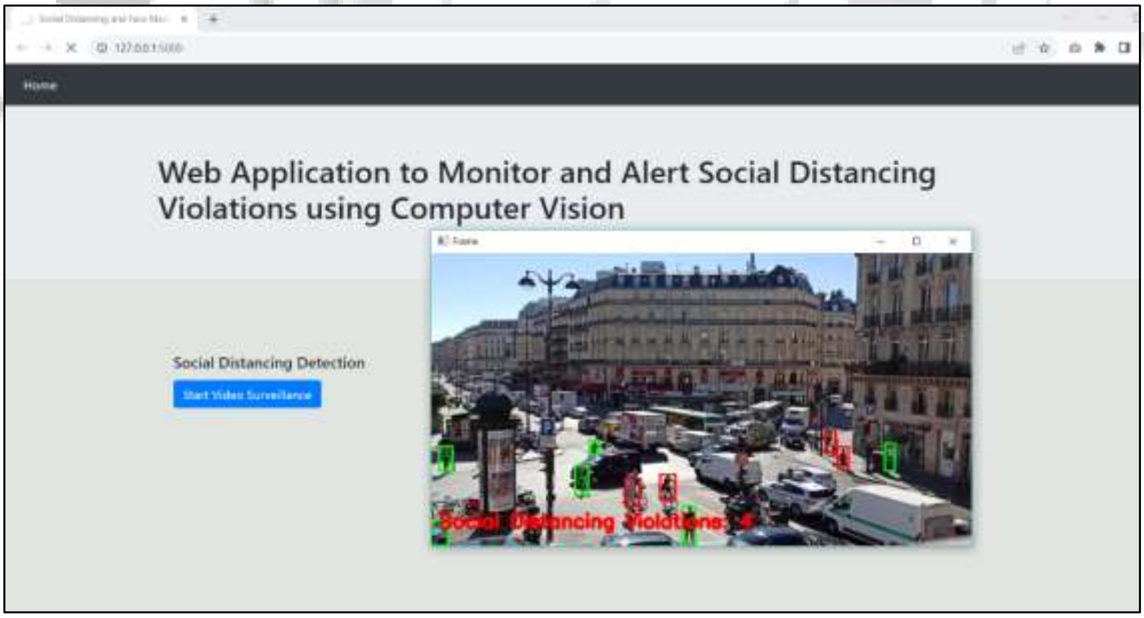

Fig. 7: Output result 1

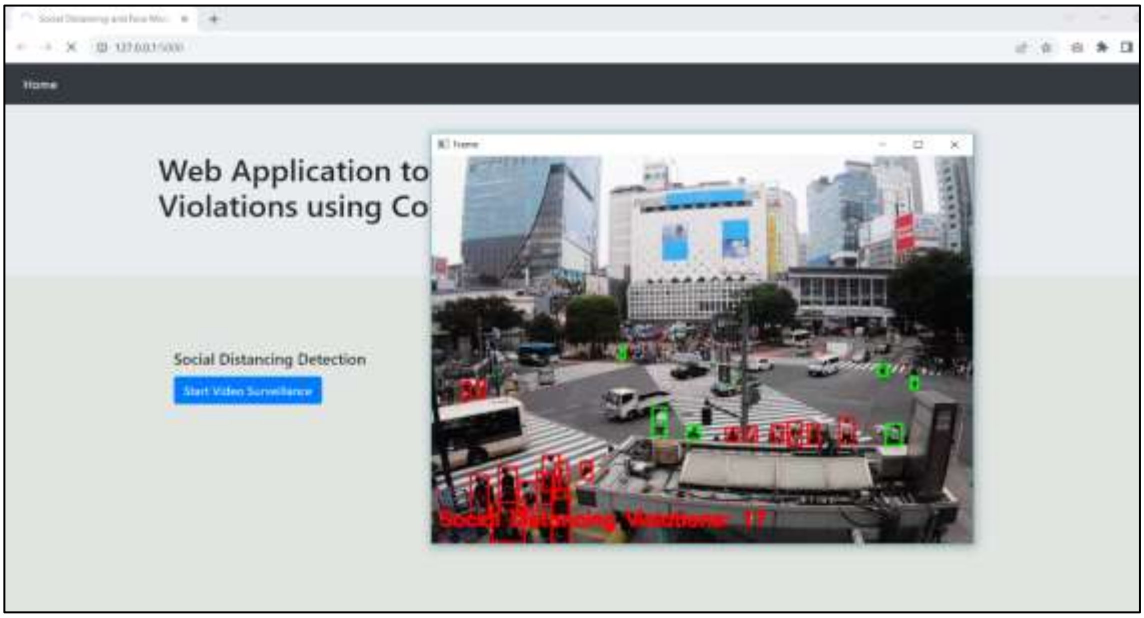

Fig. 8: Output result 2

VIII. Conclusion and Future Enhancement

There are similar solutions, but this solution is unique in its own way of implementation especially with the use of a better built user interface. In this paper, the way of alerting violations using an on field alarm and sending an email to the security team displays a better approach to minimize the spread of virus.

Technology is enabling new innovations every day and we have to build tools for everyone to access this technology to make a positive and verifiable impact on the world. Social distancing is helpful in reducing the transmission of viruses and diseases in general. This application calculates the distance between people in a frame and identifies whether social distance is maintained or not to effectively ensure safety of everyone. It can be further enhanced by optimizing the person detection algorithm and calculating distance between the people more inclusively and accurately.

When there are obstacles in between the people, then the process to find out shortest distance can be enhanced by using graph search algorithms such as A* algorithm when coordinates of obstacles are known.

The output data such as number of violations at a specific place and time period can be logged in a file or database or on cloud and can be used to draw interpretations and perform analysis. The accuracy percentage can be calculated by,

Accuracy = ((number of successful detections) / (total number of observations)) * 100

The system can be improved by adding the following features

1) Edge Case 1 – If the pathway is covered with aerial obstructions, the pedestrians in the input videos may not be visible. In such a case, we could use IR sensors to detect people. Also, a new model using Convolutional Neural Network (CNN) can be developed which could be trained on IR images as well.
2) Edge Case 2 – When the input video frame from the camera device is blurred, the accuracy of the model decreases. In such a case, Generative Adversarial Network model can be used to remove the noise from the input.

References

[1] Iram Javed, Muhammad Atif Butt, Samina Khalid, Tehmina Shehryar, Rashid Amin, Adeel Muzaffar Syed & Marium Sadiq, “Face mask detection and social distance monitoring system for COVID-19 pandemic”, Springer Link, 30/09/2022.
[2] B. Sahaja, R. Krupa Rani, G. Arun Kumar, A. Jagan, “SOCIAL DISTANCING VIOLATION DETECTION SYSTEM”, International Research Journal of Modernization in Engineering Technology and Science, June 2022.
[3] Sergio Saponara, Abdussalam Elhanashi, Qinghe Zheng, “Developing a real-time social distancing detection system based on YOLOv4-tiny and bird-eye view for COVID-19”, Springer Link, 22/02/2022
[4] Saharsh Arya , Leena Patil , Ayushi Wadegaonkar , Nishad Shinde, Palash Gorsia, 2021, Study of Various Measure to Monitor Social Distancing using Computer Vision: A Review, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 10, Issue 05 (May 2021),
[5] Krisha Bhambani, Tanmay Jain, Kavita A. Sultanpure, “Real-time Face Mask and Social Distancing Violation Detection System using YOLO”, IEEE Explore, 31/12/2020.
[6] Hrishikesh Kurhade, Akshata Magar, Monika Maurya, Prof. Dhanashri Bhopatrao, “Social Distancing Violation Alert System”, International Research Journal of Engineering and Technology (IRJET), April 2021.
[7] Gündüz, M.Ş., Işık, G. A new YOLO-based method for social distancing from real-time videos, Springer Link, 7 April 2023, 15261–15271 (2023). https://doi.org/10.1007/s00521-023-08556-3.
[8] Durgesh Aswar, Abhijeet Adsul, Rushikesh Gadwe, Komal Wanzare, Chandrashekhar Patil, “Social Distance

Violation Alarming System: A Deep Learning Approach", Volume 10, Issue 5, JETIR May 2023.
[9] Krishna Maid, Suraj Gotarne, Aarti Gaikwad, Vandana Inamdar, "Real-Time Social Distancing Detection", IJSRD - International Journal for Scientific Research & Development, Vol. 11, Issue 3, 2023, 01/06/2023.
[10] Keniya, Rinkal and Mehendale, Ninad, Real-Time Social Distancing Detector Using Socialdistancingnet-19 Deep Learning Network (August 7, 2020). Available at SSRN: https://ssrn.com/abstract=3669311 or http://dx.doi.org/10.2139/ssrn.3669311