An Automated Regulatory Real Time System for Face Mask Detection and Social Distance Monitoring using Deep Learning

Divya Mohan*
Department of Computer Science and Engineering
Albertian Institute of Science and Technology,Kalamassery,Kerala.
Email: divyamohan@aisat.ac.in

Abdullah Shaheen
Department of Computer Science and Engineering
Albertian Institute of Science and Technology,Kalamassery,Kerala.
Email: cruflox@gmail.com

Harrith V Rajesh
Department of Computer Science and Engineering
Albertian Institute of Science and Technology,Kalamassery,Kerala.
Email: harrithvrajesh007@gmail.com

Abstract

The Covid-19 virus outbreak had a huge impact on different sectors across countries and such impact caused lifestyle, economic and other problems to millions of people around the world. The rising number of COVID-19 tests and vaccines helps control and provide more information about the epidemic's spread, potentially allowing it to be contained to prevent further infections. The small step of wearing a face mask as well as following social distancing would save lots of lives as the spread of the virus is being mitigated. This application uses YOLO object recognition on video footage and photos in real time, for calculating the social distance & identifying face masks on people's faces. There are different types of algorithms available, YOLO stands out from all other competitors present currently due to its higher resilience and faster detection speed in identifying masked faces and human subjects. The minimum distance issued by the World Health Organization to keep while adhering to social distance is 3 Feet or 1 meter, keeping this as the base for calculating distance, the model was trained and used for object detection.

Keywords: *Artificial neural network, object detection, YOLO*

1. Introduction

The past couple of years during the COVID 19 pandemic has brought us a lot of challenges, especially in the working sectors. To hamper the pandemic, numerous nations have imposed strict curfews and lockdowns where the public authority authorized that the residents stay safe in their home during this pandemic. Various healthcare organizations needed to clarify that the best method to hinder the spread of the virus is by distancing themselves from others and by reducing close contact. After the discovery and implementation of vaccines, most people are back to their everyday work life and continuing with the regular functioning of establishments like restaurants, schools, universities, construction sectors, offices, all ensuring workplace safety such as wearing masks and social distancing. Government authorities are working hard to ensure that these enterprises adhere to all safety regulations in order to flatten the curve and to help the healthcare system. They are conducting regular inspections and even shutting down these establishments if they keep violating the safety protocols. Many of these establishments are working towards the automation of detecting such violations, thereby reducing the time and labour spent for the same. The main aim of this work is to detect violations such as not wearing a mask or not following social distancing in a workplace, or a public event. This model has two applications. The first one being recognition of people's faces in order to evaluate whether or not they're wearing an appropriate mask. The second application is to recognize each personnel and assess whether or not the WHO recommended social distance is being maintained between two individuals. The goal is to significantly reduce the amount of effort from supervisory authorities in overseeing every event, by implementing this automated regulatory system. The application was implemented using darknet and OpenCV and YOLO was used for mask and people detection. A desktop app was built for the same purpose and it can successfully detect people who are not following covid protocols and show the total number of people in an interactive user interface.

2. Related Literature Review

Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques. In order to automate the task of monitoring social separation using surveillance footage, this study presents a deep learning-based system. The suggested framework by N. Singh Punn, S.K. Sonbhadra, S. Agarwal (2020) employs the YOLO v3 object detection model to distinguish persons from the background, as well as the Deepsort technique to track recognized people using bounding boxes and assigned IDs. In terms of mean average precision (mAP), frames per second (FPS), and loss values defined by object classification and localization, the results of the YOLO v3 model are compared to those of other popular state-of-the-art models, such as faster region-based CNN (convolution neural network) and single shot detector (SSD).

Social Distancing Tracker Using YOLO v5. This framework by Dongfang Yang, Ekim Yurtsever, Vishnu Renganathan, Keith A. Redmill (2020) is centered on a solution for continually determining social separation using YOLO object discovery on video film and photographs . To recognize persons in a video, the system employs the YOLOv5 object detection algorithm. Using recognized bounding box data, the detection model separates persons into categories. The Euclidean distance is used to resolve the pairwise distances

between the centroid of the distinct bounding boxes of the individuals. It uses an estimate of the real distance to pixel and setting and edge to detect social distance infringement between persons. To determine whether the distance esteem exceeds the base social distance limit, an infringement edge is set up. The system enables faster derivation times, making it suitable for delivering real-time results without sacrificing precision, even in complex situations. The framework ensures faster derivation times and is simple to use. As a result, even in highly complicated configurations, it is suitable for delivering real-time results without sacrificing precision.

Yonghui Lu et al. (2020) proposed an efficient YOLO Architecture, YOLO-compact for a real time single category detection.The number of categories in object detection is always one in most practical applications, and this paper aims to design a YOLO-compact network with high efficiency for single-category real-time object identification. YOLO-model compact is only 9MB in size, making it smaller than tiny-yolov3, tiny-yolov2, and much smaller than YOLOv3. YOLO-compact has an AP result of 86.85%, which is 37 percent higher than tiny-yolov3, 32 percent higher than tiny-yolov2, and even 2.7 percent higher than YOLOv3.

M. B. Ullah (2020), the author proposed a CPU-based YOLO object detection model that is intended to run on non-GPU computers The Proposed Model Optimize YOLO with OpenCV. Therefore, compared to other system, real time object detection can be much faster on CPU based computers. In CPU-based computers, their model detects objects from videos at a rate of 10.12 to 16.29 frames per second with an accuracy of 80-99 percent.

3. Proposed Method

In this work, we offer a method that detects individuals in real time to track social distance norms and real-time face detection to track face-mask usage. In this section, we'll go over how to put this solution together.

3.1 Dataset

Figure 1 below shows the custom dataset used for Face Mask Detection, this contains 2000 Images with annotations made for every object (i.e., person, mask and no mask) present in the frame. The need for creating a custom dataset was because the COCO dataset doesn’t contain classes for face mask detection. Based on the above figure, the labeling was done for each class.

Figure 1. Dataset Example

File Edit Format View Help

```
0 0.698113 0.518192 0.283019 0.905075
0 0.456958 0.488630 0.244104 0.736795
1 0.466981 0.568222 0.110849 0.291079
2 0.649764 0.596647 0.219340 0.393412
```

Figure 2. Annotations

It can be observed from Figure 2, it contains 3 different classes (0,1 & 2). The classes use for face mask detection are person, no_mask and mask.

3.2 Design

Figure 3 describes the design process. The video is first taken as input and then transformed into frames in the suggested system architecture. As a result, each frame is analyzed by the YOLO algorithm for object detection, and because YOLO can process roughly 45 frames per second, the analysis of object detection and object tracking is completed. The following stage is to determine the distance between each detected person, which should be at least 1 meter according to the World Health Organization COVID-19 Protocol. As a result, the Threshold value is set to 1 meter. We need to locate the centroid of each detected object and use the Euclidean distance to calculate the distance between each individual. Once the distance is determined, if the result is greater than the threshold value, social distance is maintained if the result is less than the threshold value, social distance is not maintained, and the authorities are notified.

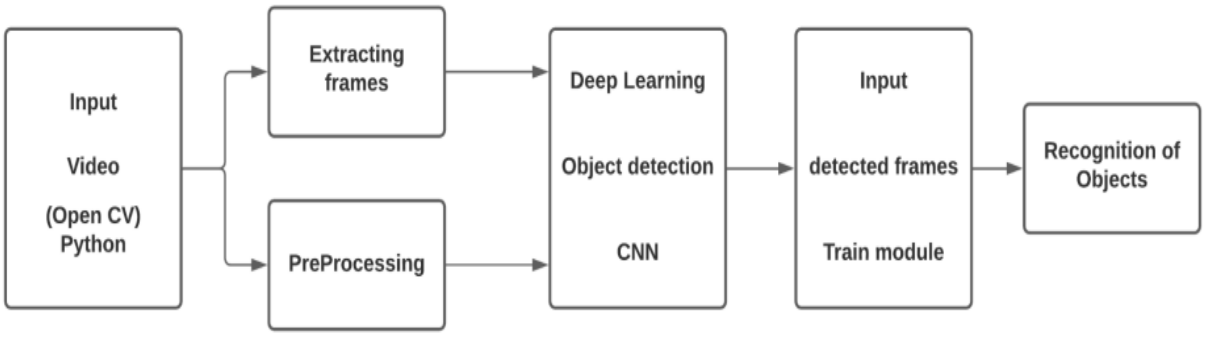

Figure 3. Design

3.3 YOLOv4

Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao (2020) proposed YOLOv4 with some major changes from its predecessor YOLOv3, resulting in significant improvements in both speed and accuracy. YOLOv4 is extremely fast, easy to train, robust, stable and gives promising results even for tiny objects, hence, we selected it as our object detector of choice. For an input image/frame, it detects objects belonging to three classes — unmasked faces, masked faces and people. This effectively means that the same model is used for both person detection to track social distancing and for masked-face detection for facemask monitoring. This significantly boosts overall efficiency and simplicity.

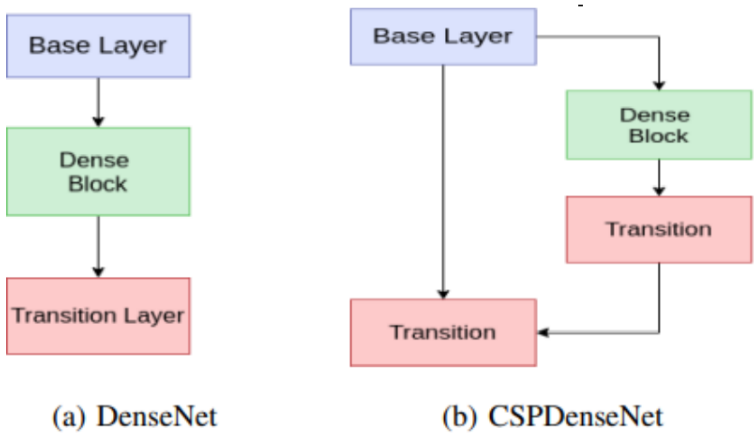

Figure 4. Structure comparison between Standard DenseNet and CSPDenseNet

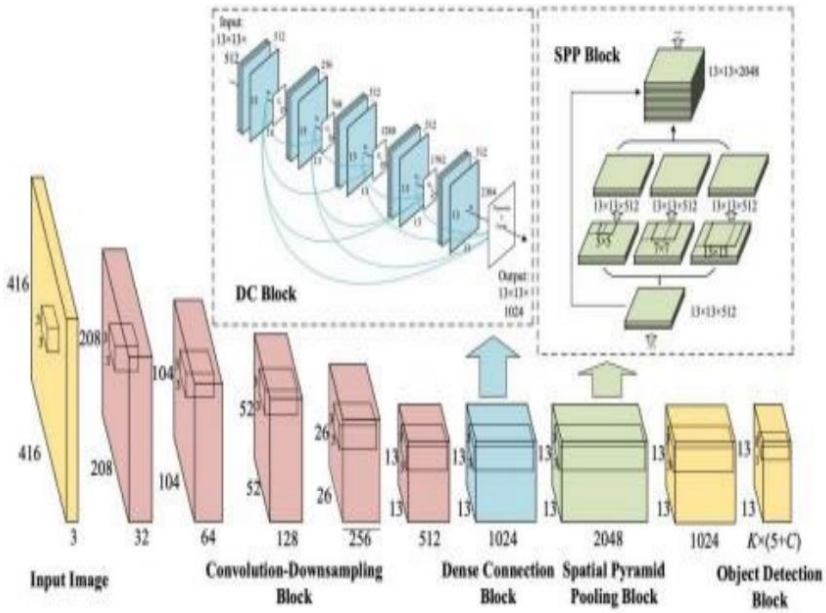

Figure 5. Demonstration of SSP Integration with YOLOv4 Architecture

The pipeline has 3 parts, the backbone, the neck and the head. The network takes as input an RGB image or frame. The backbone is responsible for extracting features from the image. For this Cross-Stage-Partial-connections Darknet (CSPDarknet53) proved to be the optimal choice. The output from the base layer is divided into two segments. One goes to the Dense Block whereas the other goes directly to the next transition layer as shown in Figure 5. Dense blocks contain layers and each layer contains Batch Normalization and ReLU followed by a convolutional layer. Each layer of the Dense Block takes the feature maps of all previous layers as input. This expands the receptive field of the backbone and helps in isolating complex features of an image. Spatial pyramid pooling (SPP) was used as the neck which contains blocks for increasing the receptive field and to aggregate parameters from different levels of the backbone. SPP's integration with the YOLO pipeline is described in Figure 5.

For this network the concept of Bag of Freebies (BoF) was adopted where freebies refer to training strategies which enhance the performance of the network without adding to its computational expense. There are multiple such freebies to choose from, out of which, Cutmix regularization, Mosaic data augmentation, DropBlock regularization and Class label smoothing were chosen for the backbone. Strategies like Self-Adversarial Training, CIoU-loss, Cross miniBatch Normalization (CmBN), mosaic data augmentation,elimination of grid sensitivity and DropBlock regularization were adopted as some of the freebies for the detector. Bag of Specials (BoS) technique was also adopted wherein 'specials' refer to strategies that enhance the network's performance while increasing the inference cost by a small amount. Mish activation, Cross-stage partial connections (CSP) and Multi-input weighted residual connections (MiWRC) were the specials chosen for the backbone, and Mish activation, SPP, SAM and PAN were chosen as specials for the detector.

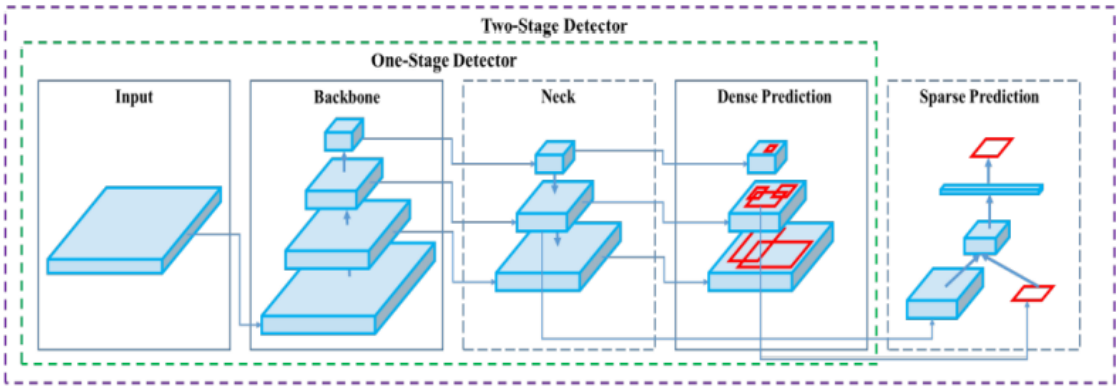

Figure 6. Two Stage Detector

Input: Image, Patches, Image Pyramid Backbones: VGG16 , ResNet-50 , SpineNet, EfficientNetB0/B7 , CSPResNeXt50, CSPDarknet53

Neck: Additional blocks: SPP , ASPP , RFB, SAM Path-aggregation blocks: FPN , PAN, NAS-FPN , Fully-connected FPN, BiFPN , ASFF , SFAM

Heads: Dense Prediction (one-stage):RPN , SSD, YOLO, RetinaNet (anchor based) CornerNet CenterNet , MatrixNet, FCOS (anchor free)

Sparse Prediction (two-stage): Faster R-CNN , R-FCN , Mask RCNN (anchor based) RepPoints (anchor free)

3.4 Training

Google Colab was used to train this model. For training a model to detect the custom objects modifications are done on the configuration files.

Changes:

- Changed subdivisions from 8 to 16
- Changed max_batches from 500500 to 6000 (i.e., Number of classes * 2000)
- Changed steps from 400000, 450000 to 4800, 5400 (i.e., 80% and 90% of current max_batches).
- Changed filters from 255 to 24 (i.e. [Number of classes + 5] * 3)

The model was trained using 1500 images. 500 images were used as the validation dataset. mAP and average loss is found out at every 100 iterations. The model was trained for 6000 iterations using Google Colab.

3.5 Detection

Opencv was used to capture the frame from a webcam or video.

```
cap = cv2.VideoCapture(0)
cap = cv2.VideoCapture("video.mp4")
```

The frame is used by the darknet to detect the classes from the frame.

```
detections = darknet.detect_image(netMain, namesList, darknet_image, thresh=0.25)
```

This frame is then passed onto a python function that uses the detections and its coordinates to draw bounding boxes on each frame using openCV methods.

```
cv2.rectangle(img, (box[2], box[3]), (box[4], box[5]), (255, 0, 0), 2)
```

3.6 Social Distancing

This work uses YOLO v4 to detect people. People are detected and bounding boxes are drawn using openCV methods. The centroid for each person was found and appended to a list. A red bounding box indicates that a person is not following social distancing protocol.
The distance between each person is found out using the Euclidean distance formula:
dst = math.sqrt(p1**2 + p2**2)
The threshold distance is set and is checked against the obtained distance to know whether the person is keeping a minimum distance or not. If they are not following the protocol the detected class (person) is appended to a red_list[].

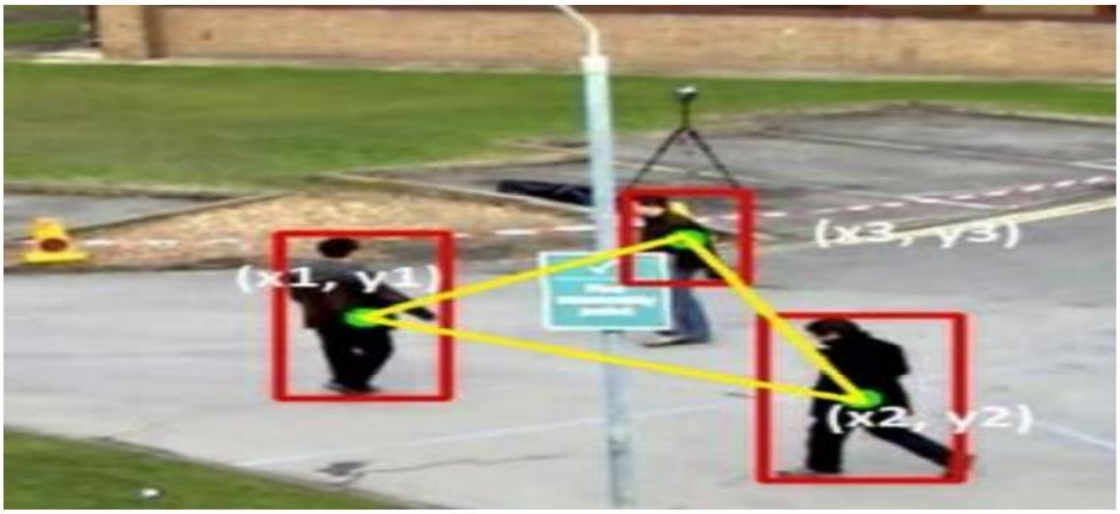

Figure 7. Calculation of Euclidean Distance

3.7 User Interface

The UI was built using html, CSS and JavaScript. The eel python library was used to integrate the backend and the frontend. The UI consists of three pages: a landing, a choosing page and the main detections/results page. The choosing page has two buttons: real time detection from which you can use the webcam and local video detection from which you can use an already existing video from the directory. Both these pages lead to a similar result page in which detections are shown. The right side contains all the values obtained from the detections i.e. the number of people wearing and not wearing masks, number of people not following the social distance protocol and also the total number of people in the crowd. The videos after detections are saved in the local directory which can be accessed later.

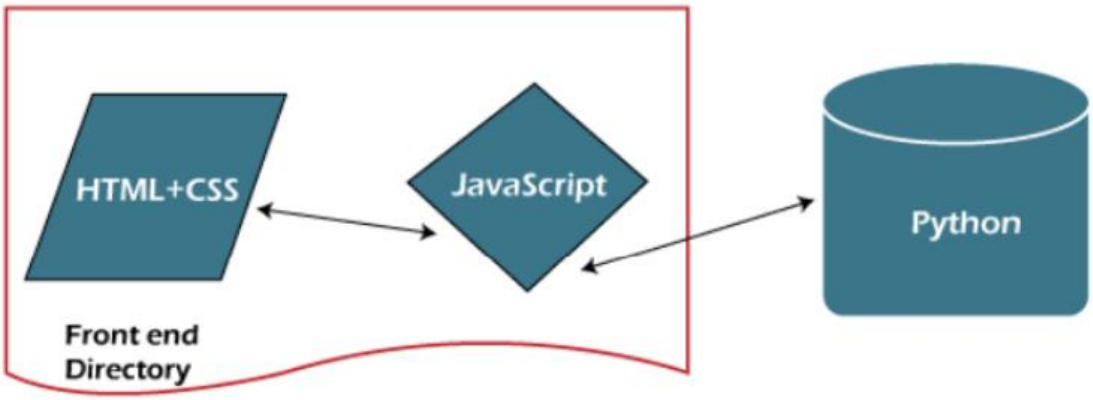

Figure 8. Eel Integration

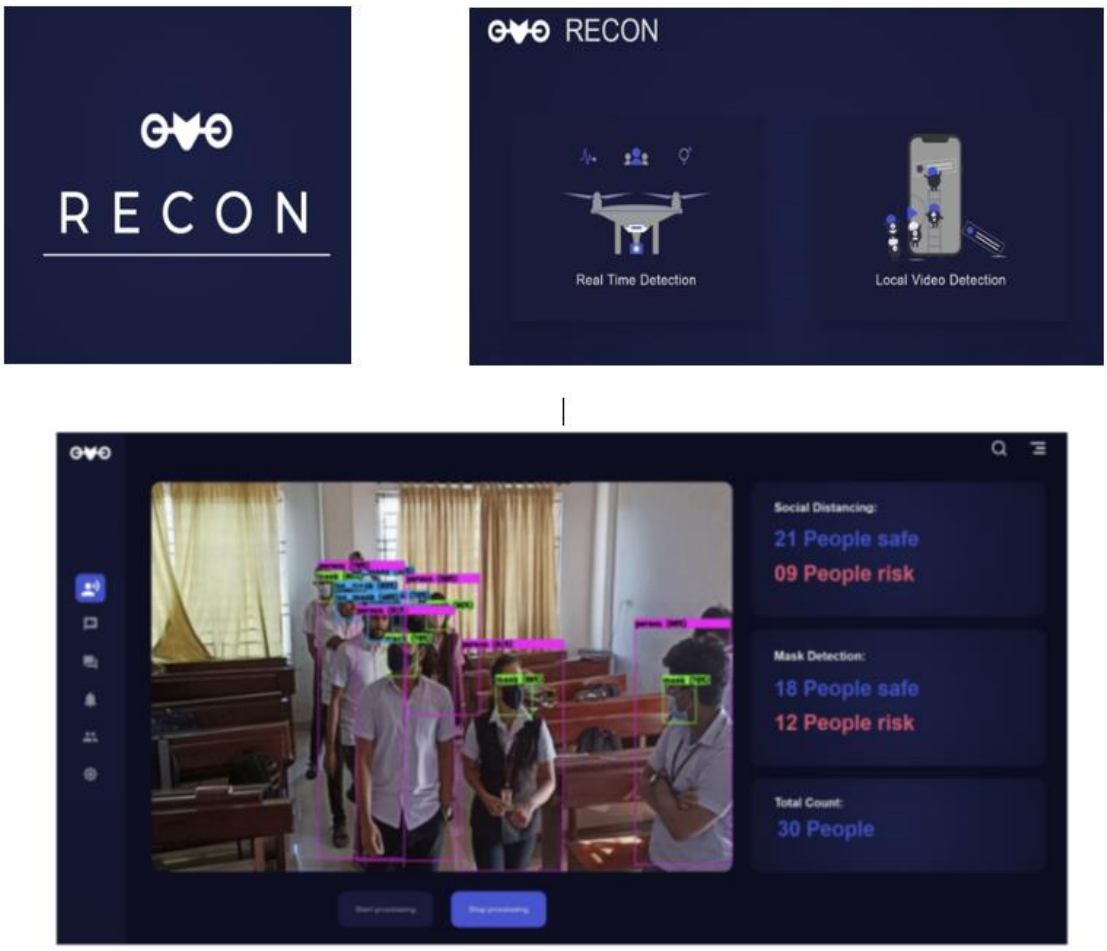

Figure 9. User Interface

4. Result and Analysis

The proposed system can effectively identify the people that are violating the COVID protocols and find the total number of people in the surrounding.

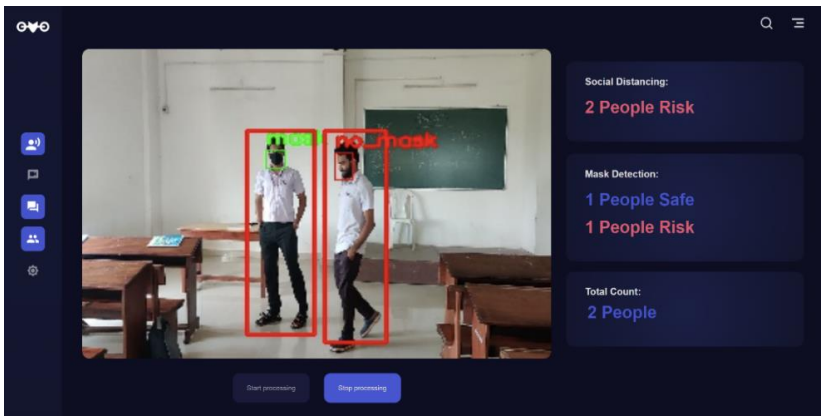

Figure 10. Social Distancing and Mask Detection

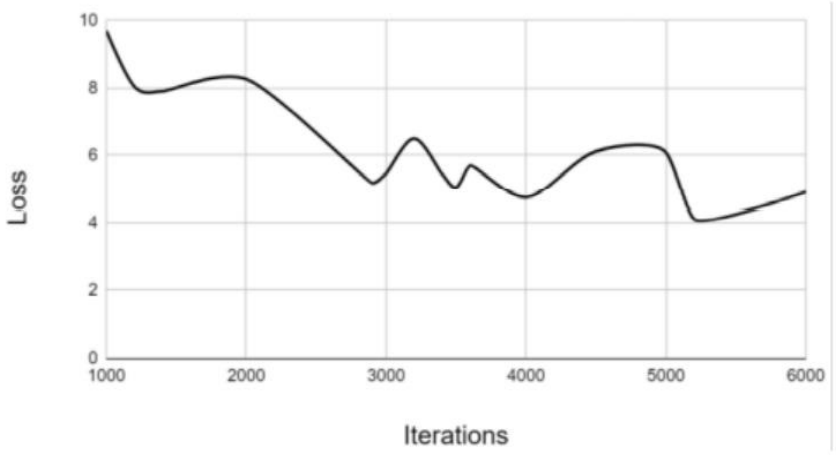

Figure 11. Loss Graph

The training loss is the error for the training data set (i.e., it is measured during each epoch). On the other hand, the validation loss is the error after passing the validation data through the network already trained (i.e., it is measured after each epoch). An epoch refers to the number of passes of the complete training dataset. The validation loss may be calculated from the training as:

validation loss = (training loss÷training images)*1000 (1)

Equation (1) is used to calculate validation loss. Training loss and validation loss are never measured in percentage. Rather, they are measured in terms of the mean square value of error. It may be observed that the validation loss remains constant or increases as the training loss increases from 4000 iterations. This condition is referred to as overfitting.

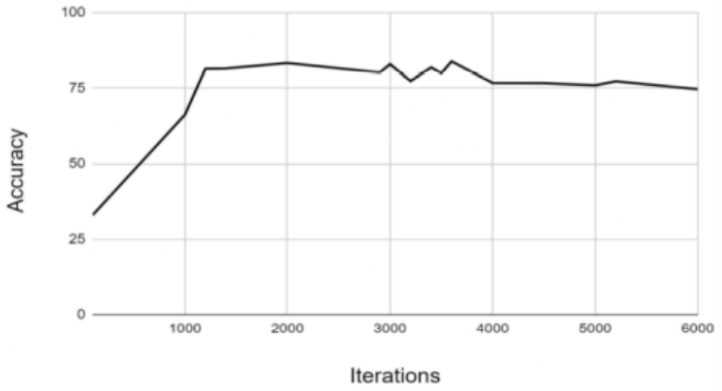

Figure 12. Accuracy Graph

The validation accuracy is an important metric used to predict the performance of an algorithm. The model is validated by calculating both metrics–loss and accuracy. It may be observed that the accuracy is increasing steeply till 1000 iterations, after that it obtains the maximum accuracy between 3000 and 4000 iterations.

Table 1. Loss, Accuracy,F1-Score,Precision,Recall

ITERATIONS	LOSS	ACCURACY	F1-SCORE	PRECISION	RECALL
1000	9.7	66.24	0.52	0.89	0.63
1200	8.05	81.49	0.69	0.89	0.78
1400	7.91	81.56	0.76	0.9	0.82
2000	8.27	83.35	0.73	0.92	0.81
2900	5.18	80.22	0.74	0.89	0.81
3000	5.47	83	0.75	0.92	0.82
3200	6.5	77.34	0.75	0.89	0.81
3400	5.45	81.9	0.79	0.92	0.85
3500	5.05	79.97	0.77	0.89	0.82
3600	5.7	83.92	0.77	0.89	0.83
4000	4.74	76.69	0.77	0.88	0.82
4500	6.12	76.64	0.76	0.89	0.82
5000	6.09	75.94	0.75	0.88	0.8
5200	4.12	77.25	0.76	0.89	0.82
6000	4.9	74.66	0.75	0.88	0.81

5. Application

Halls: During pandemic there was special instruction to follow the covid protocols on the halls that conducted programs like weddings or conferences. To limit the covid spread this system can be used on the halls using CCTV cameras and alert to the concerned authority.

Workplaces: This system can be implemented on the workplace to make the personnel follow the covid protocols.

Hospitals and Schools: These places are where the number of people are high and it is extremely difficult for a person to enforce the protocols so this system can be used to automate the covid protocol enforcement.

6. Conclusion

This application was created in order to produce an effective method for identifying and notifying officials when someone fails to follow COVID 19 safety regulations in a workplace, business facility, or other location The model will detect people who are not following covid 19 protocols i.e., not wearing mask and not following social distance protocols. Euclidean Distance is found out between the centroids of the bounding boxes to detect violations. OpenCV was used to capture each frame and draw the bounding boxes. A desktop app called ‘Recon’ was created to implement the same.

References

Amit Chavda, Jason Dsouza, Sumeet Badgujar, Ankit Damani (2020). "Multi-Stage CNN Architecture for Face Mask Detection ".

B. M.,et al.(2020). "Enabling and emerging technologies for social distancing: a comprehensive survey and open problems.", arXiv:2005.02816.

B. Qin and D. Li(2020). "Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19," Sensors, vol. 20, no. 18.

Dongfang Yang, Ekim Yurtsever, Vishnu Renganathan, Keith A. Redmill, U¨mit O¨zgu¨ner (2020). "A Vision-based Social Distancing and Critical Density Detection System forCOVID-19", arXiv-2007.

Guangxin Lou, Hongzhen Shi (2020)." Face image recognition based on convolutional neural network".

Hui (2020). "YOLOv4", Medium, [Online], https://medium.com/@jonathan hui/yolov4-c9901eaa8e61.

Alexey Bochkovskiy , Chien-Yao Wang, Hong-Yuan Mark Liao (2020). "YOLOv4: Optimal Speed and Accuracy of Object Detection", arXiv:2004.10934v1 [cs.CV] 23 .

Imran Ahmed , Misbah Ahmad , Joel J.P.C. Rodrigues, Gwanggil Jeon, Sadia Din (2020). "A deep learning-based social distance monitoring framework for COVID-19", ELSEVIER2020 .

Joseph R. et al (2018)."YOLOv3: An Incremental Improvement", arXiv:1804.02767.

J. Zhang, F. Han, Y. Chun, and W. Chen(2021). "A novel detection framework about conditions of wearing face mask for helping control the spread of COVID-19," IEEE Access, vol. 9, Article ID 42975, 2021.

Mahdi Rezaei and Mohsen Azarmi (2020). "Deep SOCIAL: Social Distancing Monitoring and Infection Risk Assessment in COVID-19 Pandemic", MDPI/Journals/Applied Sciences.

N. Singh Punn, S.K. Sonbhadra, S. Agarwal (2020). "Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deep sort techniques", arXiv:2005.01385 .

Sergio Saponara, Abdussalam Elhanasi and Alessio Gagliardi (2020). "Implementing a real-time, AI-based, people detection and social distancing measuring system for Covid-19", Springer Link.

Shashi Yadav, (2020). "Deep Learning based Safe Social Distancing and Face Mask Detection in Public Areas for COVID-19 Safety Guidelines Adherence", iJRASET.

S. Yadav(2020), "Deep learning based safe social distancing and face mask detection in public areas for COVID-19 safety guidelines adherence," International Journal for Research in Applied Science and Engineering Technology, vol. 8, no. 7, pp. 1368–1375.

Rinkal Keniyam, Ninad Mehendale (2020). "Real-Time Social Distancing Detector Using Socialdistancingnet-19 Deep Learning Network", SSRN .

Tingxiang Fan et al. (2020). "Autonomous social distancing in urban environments using a quadruped robot", arXiv: /2008.08889 .

WHO,(2020)."Coronavirus disease (COVID-19) advice for the public", https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advicefor-public.