Pandemic Surveillance Through Perspective Transformation Using YOLO and Mobile Net

Prachi Palsodkar, Prasanna Palsodkar, Yogita Dubey, and Roshan Umate

1 Introduction

In the year 2021, vaccination against coronavirus is completing fast and the world is trying to start with new normal. But the vaccination rate is not too high and it doesn't cover hundred percent assurances. To avoid the pandemic spread, social norms need to be followed. Many mutants have been observed which is another major concern that needs to be notified in public. The best preventive majors under such circumstances are to keep social distance and wear the mask in all crowded places like malls, markets, shops, schools, etc.

Considering the current scenario, tracking of social distancing is essential. A pandemic surveillance system is an approach that will be monitoring social distancing in specific areas like at the entrance of the building, mall, etc., and it will check whether the people present in the area are wearing masks or not. The system will be using CCTV feed to monitor the area. It uses object detection models like YOLO, Mobile-Net that help to detect the pedestrians on the frame of the image and then the system calculates the distance between them by performing the perspective transformation on the frame. The system also performs mask detection on the image to detect the mask on the pedestrians. The system considers these parameters to decide the output like the density of people, number of mask violations and number of social distancing violations [1]. The threshold limit on the density of people is set based on the government guidelines issued in the area. The system treats the social distancing

P. Palsodkar (✉) · P. Palsodkar · Y. Dubey
Yeshwantrao Chavan College of Engineering, Nagpur, Maharashtra, India
e-mail: prachi.palsodkar@gmail.com

R. Umate
Jawaharlal Nehru Medical College, Datta Meghe Institute of Medical Sciences (DU), Wardha, India

S. Mukherjee et al. (eds.), *Intelligent Systems for Social Good*,
Advanced Technologies and Societal Change,
https://doi.org/10.1007/978-981-19-0770-8_16

violation if the distance between two people is less than 6 feet. The alert signal by the system is given as one of the three, such as low risk, medium and high risk [2, 3].

The main objective in this chapter is to check the pedestrian flow and predict the distance with a warning alert system. This chapter will cover related work in the same area in Sect. 2, Preliminary Methodologies in Sect. 3, experimentation in Sect. 4 and result and conclusion in subsequent sections.

2 Related Work

COVID-19 spreads mainly among people who are in close contact (within about 6 feet) for a lengthened period. Spread happens when an infected person coughs, sneezes, or talks, and droplets from their mouth or nose float into the air and land in the mouth or nose of people nearby [2]. We need social distancing devices to ensure social distancing without any human interaction. Mostly there are wearable bands or watches that use proximity sensors that detect any nearby object, but it is not a protected way to maintain social distancing as it increases the risk of spreading COVID-19 through the band itself [4].

During the pandemic, significant work has been carried out on social distancing detection, Yang et al. [5] proposed a warning system using region-of-interest. Foster R-CNN model and YOLO4 was used to detect the real-time behaviour of the pedestrian. Zhou et al. [6] used MDA approach to check the crowd behaviour. They have used crowd collective behaviour classification for behaviour prediction. Punn et al. used the deep learning YOLO v3 technique applicable to various situations and used it to monitor real-time pedestrians. Real-time voice alert for workers' safety is discussed by Khandelwal et al. [3]. A similar kind of safety assist is discussed by Bochkovskiy et al. [7] using YOLO v4 model. Visual social distancing in surveillance also creates a privacy problem for individuals [8].

Many popular object detection models are available like Le-Net, Res-Net, Mobile Net, R-CNN, YOLO. The selection of an appropriate model was the challenge for the said experimentation. It was carried out with the following study.

Mobile Net V2 and SSD Lite show better accuracy with a lower number of parameters. Also, it is good for low computational devices with low Madd value. YOLO v3 [9] is comparatively very fast and accurate. It has faster FPS (frame Rate) with less background errors. With reference to Fig. 1 YOLO model is selected for the analysis.

3 Methodology

In "Pandemic surveillance system" the whole system is divided into three stages, namely the Input stage, Processing stage, and Alert stage as shown in Fig. 2.

The overview of each stage is as follows:

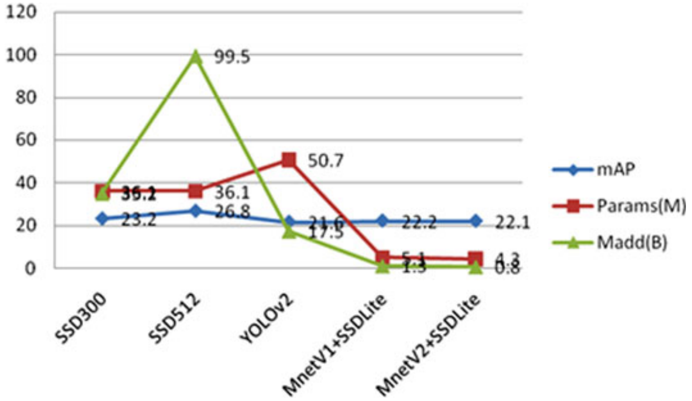


Fig. 1 Comparison of object detection models based on mAP, Params, and Madd

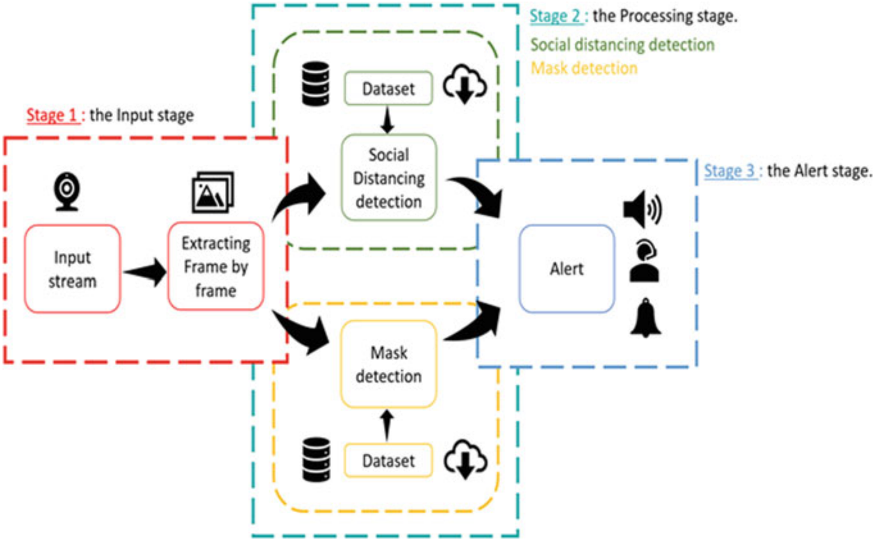

Stage 2 : the Processing stage.
Social distancing detection
Mask detection
Stage 1 : the Input stage
Stage 3 : the Alert stage.
Dataset
Social Distancing detection
Input stream
Extracting Frame by frame
Alert
Mask detection
Dataset

Fig. 2 System overview

Input stage

This stage is just a data transformation stage of the system which takes video from the CCTV continuously as input. This input video sequence will be converted into frames. These frames are later fed to the processing stage.

Processing stage: This stage contains two substages which will be executed simultaneously:

- **Social distancing detection**: This is one of the substages of the second phase of the project, which will run parallel with the Mask Detection stage. In this phase, the frame extracted from the input stream will be passed through the object detection model to detect the pedestrian in the frame and the coordinates of the detected pedestrian will be used to calculate the distance between them. The distance measurement is not simple as the camera is in a fixed position so it perceives the object's distance incorrectly; hence we use perspective transformation also called Bird's Eye View transformation. Now, after the transformation, the Euclidean distance will be calculated between the detected pedestrian and then it will be compared to the threshold (mostly 6 feet) and if the distance is less than that threshold, it will send a signal to the alert block of the project.

- **Mask detection**: This substage will receive input from the input stage. This stage will check whether a person is wearing a mask or not. Here we will have a face mask detection model based on computer vision and deep learning. If anybody is found not wearing a mask, this substage will provide this feedback to the output stage.

Alert stage: As we have seen in the processing stage, the social distancing violation data and mask violation data will be fed to this stage. This stage will then, according to the input from the processing stage, check whether there is any violation or not and also the level of severity of that violation. This will be followed by taking appropriate steps of alert. In case of high severity, it will also inform the security authority.

Working of the system consists of the input stage, as per the requirement, the area to be monitored is marked as the Region of Interest (ROI) by marking 4 points in a predefined order. Set the threshold distance for the minimum distance between two humans. After this, the working of the second stage starts, i.e. processing stage. The Processing stage has two subparts running parallel. One of them is Social Distancing Detection; in this stage, the model extracts the features from the frame of the image. The model uses CNN layers with YOLOv4 backbone which is already trained on the COCO dataset. The training gives the value of weights for the model to give accurate and efficient results. By using the YOLOv4 backbone, the model has extracted features with the probabilities of object (present in COCO dataset) in the given ROI. The model also localizes the feature by keeping a track of the coordinate of the feature on the frame of the image. These coordinates are used to perform Bird Eye View Transformation. The coordinates refer to the bounding box around the Pedestrian detected, calculate the Euclidean distance between the bounding boxes by first performing Birds Eye View Transformation to the frame of the image. The distance is compared to the threshold to calculate the social distancing violation whose count is passed to the alert stage.

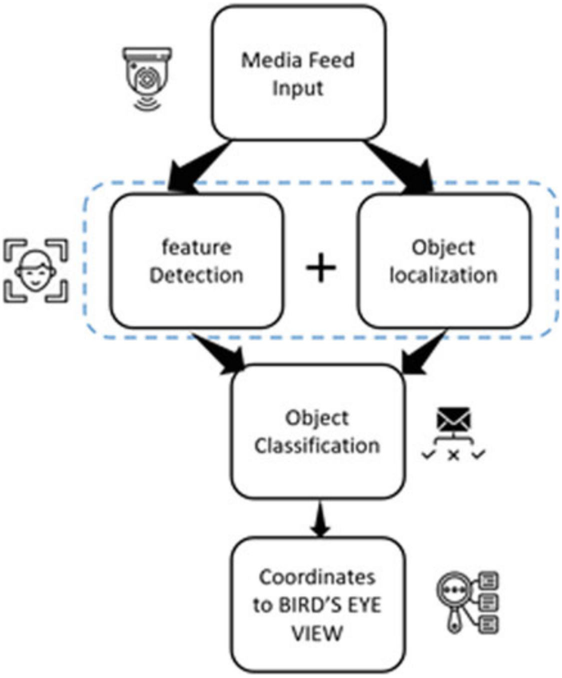
Step by step detection process is discussed in this section, which is as follows:

3.1 Pedestrian Detection

In this stage, examine pedestrian detection. The input feed from the input stage will be given to the pedestrian detection stage using the YOLOv4 framework which uses CNN to perform pedestrian detection. In the YOLOv4 framework after input, the backbone does the feature extraction process with the help of CNN. Here each layer produces a feature map with the help of a convolution layer and pooling layer. After this, the layers present at the neck section collect all the feature maps generated at different layers of the backbone. To detect pedestrians, weights are used which are obtained by training the model on the COCO dataset which has a person class. Figure 3 shows the process of Pedestrian detection.

- Object Localization: The backbone network has the location of different features extracted in the form of a feature map. This helps the system to know the location of a feature extracted. These are calculated with the help of kernels [10].
- Image Classification: After the features are extracted, the feature map is fed to the three different heads after passing through Feature Pyramid Network (FPN). The FPN is used because it enables the framework to learn objects of three different sizes. The three different heads use three different anchor boxes, i.e. (19 × 19), (38 × 38), (76 × 76). Each head predicts the following parameters: Box centre (X, Y), Box size, Object-ness score, and Class score. The total confidence score

Fig. 3 Workflow for object detection



for a class is a product of object-ness score and class score.

$$\text{Confidence Score} = \text{Object Score} * \text{Class Score} \tag{1}$$

The coordinates of the pedestrian, i.e. box centre are now sent to the next stage which is Bird's Eye Transformation.

3.2 Mask Detection

Data set is a collection of millions of images used to train the computer. We need a very large data set of people wearing face masks and people not wearing face masks [11].

3.3 Bird's Eye View

Custom trained YOLOv4 model used to detect a person on each frame. The output from the model is a list of coordinates of bounding rectangles on detected persons; where a single rectangle is represented as: [x-min, y-min, width, height]. Social Distance is determined by calculating the Euclidean distance between the bounding rectangles. The imaging plane or the camera plane is a 2D projection of 3D world coordinates, therefore, the spatial relationship between the objects in this plane changes due to camera perspective. The objects near to the camera appear larger than those that are away from it. Calculating the distance between the rectangles in this perspective would give an imprecise estimate of the actual distance, we must correct it by transforming the image into top-down-view or bird's-eye-view. We have already discussed how this transformation works in the previous section. After the perspective transformation is done, we can now calculate the Euclidean distance between objects (people in our case) with more precision [12]. Figure 4 shows a flow for Bird's Eye View.

Then, Euclidean distance (ED) for real time distance is calculated as

$$\text{RTEC} = (6/\text{ED per feet}) * \text{Real time distance} \tag{2}$$

where RTEC stands for Real time Euclidean distance.

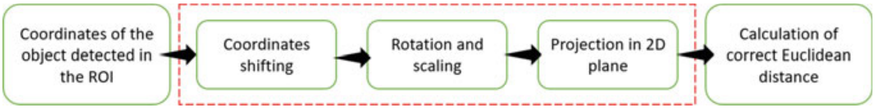

Coordinates of the object detected in the ROI
Coordinates shifting
Rotation and scaling
Projection in 2D plane
Calculation of correct Euclidean distance

Fig. 4 Bird's eye view

Table 1 Level of violation

Parameters level	Social distancing violation	Mask violation	Number of people
Safe	No	No	Less than threshold value
Low risk	Yes	No	A little above threshold
High risk	Yes	Yes	Above threshold

As shown in Eq. (2), if RTEC is below 6 feet, then an alert is created.

3.4 Alert

This is the last stage which receives input from the mask detection and bird's eye view stage. The bird's eye view stage calculates the distance which will be fed to the alert stage which decides whether the distance is safe or not. Also, the total number of people in the ROI will be fed within that particular time.

As per Table 1, the inputs received, the alert stage decides the level of violation. These levels of violation are based on the social distancing violation, number of people and mask violation. The threshold value will be determined according to the size of ROI. According to the level of violation, it will alert the authority.

4 Results and Discussion

The process begins with the CCTV feed fed into an object detection model after converting it into frames. We are using the YOLOv4 model for object detection. Along with the object detected in the bounding box, it also returns the coordinates of the objects detected which will be fed to the perspective transformation stage for further processing [13].

Figure 8 shows those coordinates that we get after the object detection as shown in Fig. 7 when Fig. 6 is provided as input. Those coordinates will be used for the calculation of Euclidean distance for the detection of social distancing violations (Fig. 8).

The mask detection model works parallel with the people detection part and takes the CCTV feed. This part returns whether the person in the image is wearing a mask or not as shown in Fig. 5. This information will be important for deciding the level of risk in the alert stage [15]. For experimentation purpose, 1315 data points were

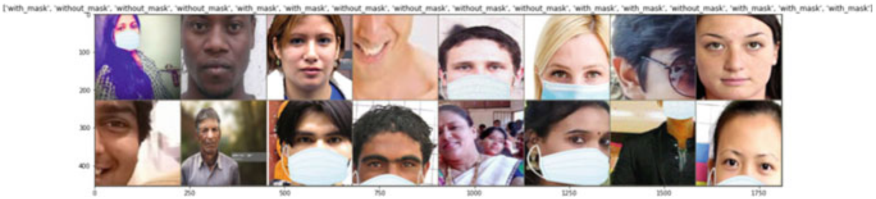

['with_mask', 'without_mask', 'without_mask', 'without_mask', 'with_mask', 'with_mask', 'without_mask', 'without_mask', 'without_mask', 'without_mask', 'with_mask', 'with_mask', 'without_mask', 'with_mask', 'with_mask', 'with_mask']
0
100
200
300
400
0
250
500
750
1000
1250
1500
1750

Fig. 5 Mask detection (Dataset source [14])

Fig. 6 Input image for object detection

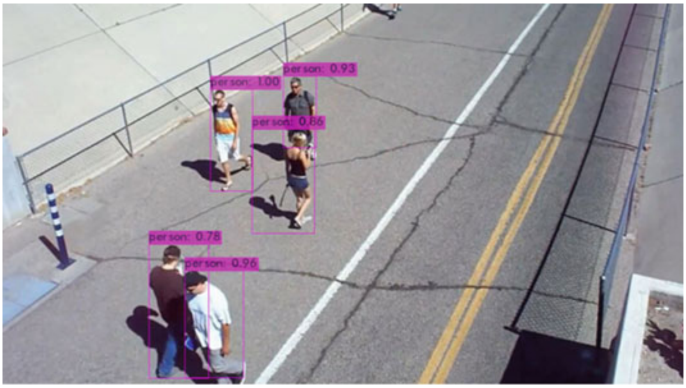
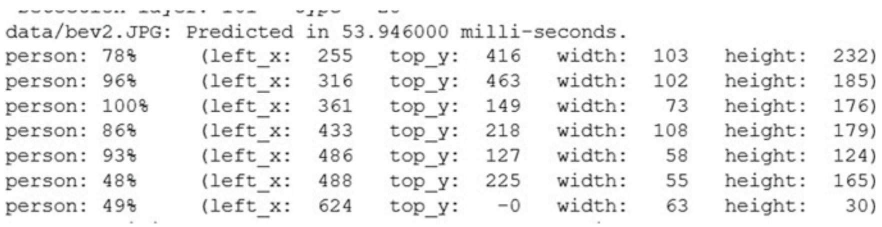


Fig. 7 Output image for object detection

considered for testing and 194 data points were considered for training (Figs. 9 and 10).

The coordinates and the image are fed to the perspective transformation stage for precise calculation of the Euclidean distance. The output will show the top view of



```
data/bev2.JPG: Predicted in 53.946000 milli-seconds.
person: 78%     (left_x:  255   top_y:  416   width:  103   height:  232)
person: 96%     (left_x:  316   top_y:  463   width:  102   height:  185)
person: 100%    (left_x:  361   top_y:  149   width:   73   height:  176)
person: 86%     (left_x:  433   top_y:  218   width:  108   height:  179)
person: 93%     (left_x:  486   top_y:  127   width:   58   height:  124)
person: 48%     (left_x:  488   top_y:  225   width:   55   height:  165)
person: 49%     (left_x:  624   top_y:   -0   width:   63   height:   30)
```

Fig. 8 Coordinates of detected people after detection

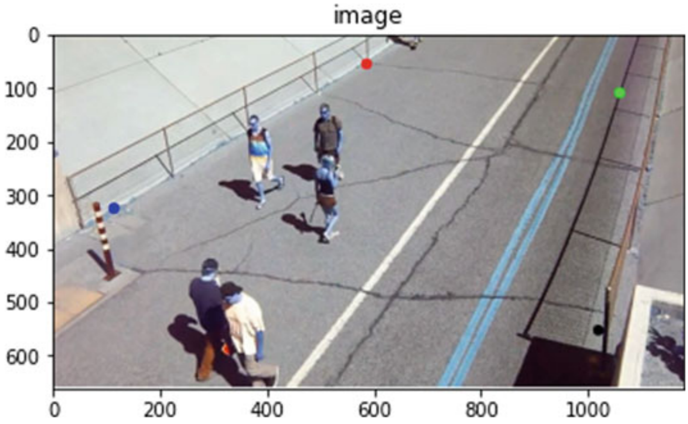
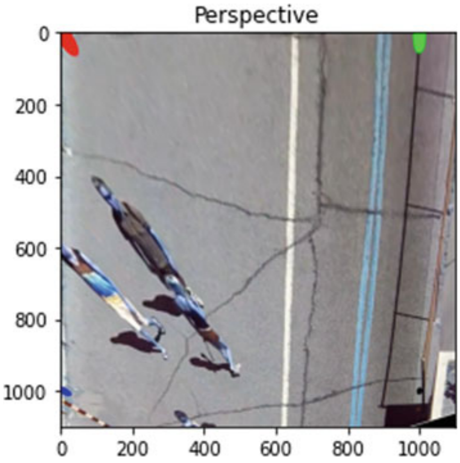


Fig. 9 Input images for perspective transformation

Fig. 10 Output perspective view of the image



the region of interest. This calculation will allow us to check for social distancing violations in the alert stage.

Figure 11 shows the top view of the CCTV frame running in the background. Whenever there is any person detected in the frame, it is represented as a dot as shown in Fig. 10. The colour of the line between the dots is green whenever those people are at a safe distance from the other people (dot) in the surroundings. Otherwise, it is shown with a red line between the dots.

Figure 11 shows 4 points along the corner represents the ROI and the two points in the middle represents 6 feet distance required to maintain social distancing which is set by health organizations worldwide for safer interaction.

Figure 12 shows the ROI and object detection model working in unison. People in the ROI are being detected. The distance violation is evident in the next image which

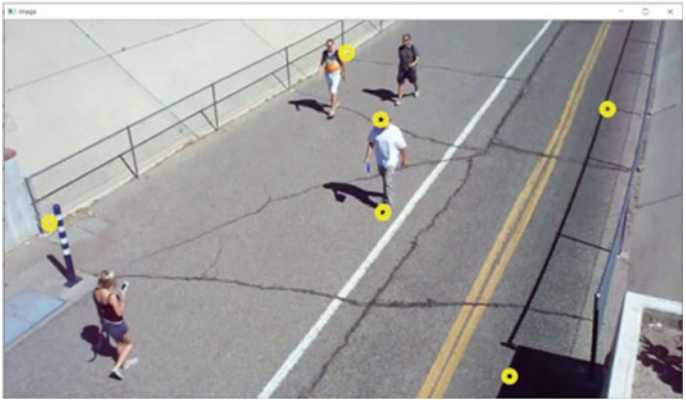


Fig. 11 Coordinates showing ROI and the middle coordinates to set the limit

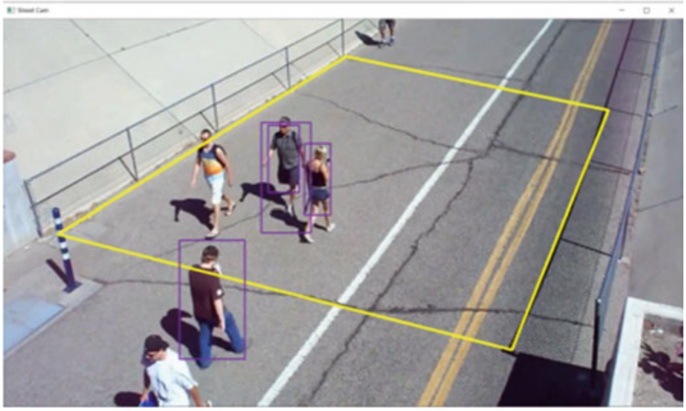
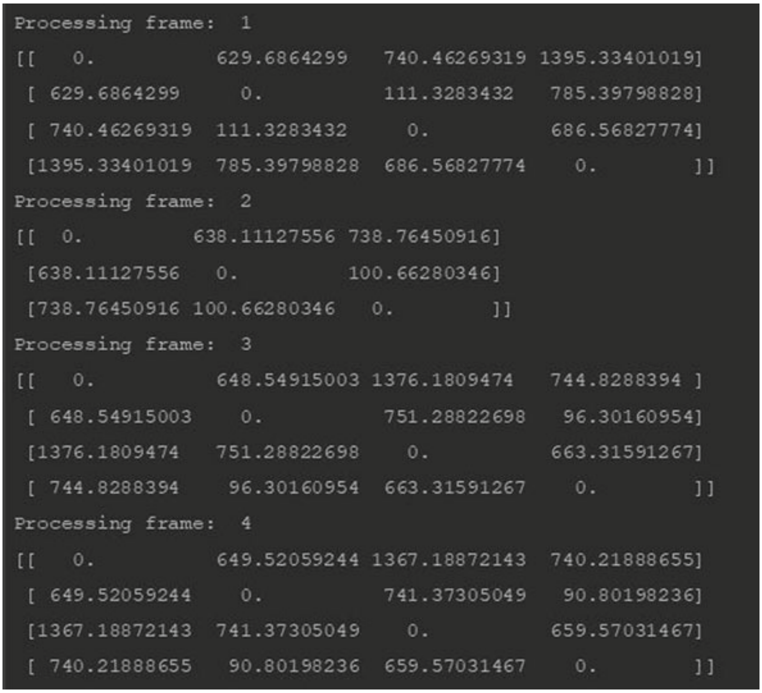


Fig. 12 Peripheral showing ROI



```
Processing frame:  1
[[   0.          629.6864299   740.46269319 1395.33401019]
 [ 629.6864299     0.          111.3283432   785.39798828]
 [ 740.46269319  111.3283432     0.          686.56827774]
 [1395.33401019  785.39798828  686.56827774    0.        ]]
Processing frame:  2
[[  0.         638.11127556 738.76450916]
 [638.11127556   0.         100.66280346]
 [738.76450916 100.66280346   0.        ]]
Processing frame:  3
[[   0.          648.54915003 1376.1809474   744.8288394 ]
 [ 648.54915003    0.          751.28822698   96.30160954]
 [1376.1809474   751.28822698    0.          663.31591267]
 [ 744.8288394    96.30160954  663.31591267    0.        ]]
Processing frame:  4
[[   0.          649.52059244 1367.18872143  740.21888655]
 [ 649.52059244    0.          741.37305049   90.80198236]
 [1367.18872143  741.37305049    0.          659.57031467]
 [ 740.21888655   90.80198236  659.57031467    0.        ]]
```

Fig. 13 Outputs coordinates values

shows a red line between the people when they are walking within 6 feet range and hence violating the norm.

Figure 13 shows output coordinate values for frames.

The distance matrix is calculated at each frame as shown in Fig. 13, these matrices show the distance of every person from every other person. Figure 14 shows a bird's eye view showing the distance between two persons.

For example, in the first frame, 4 pedestrians are identified and the distance matrix is a 4 × 4 matrix showing their distances from each other.

There are two objectives of object detection models like YOLO, R-CNN, etc. One is classification and the second is localization. In order to evaluate the performance of a model, we use the idea of IOU (Intersection over Union), which is the ratio of intersection of the region of ground truth and predicted bounding boxes to the union of the region of ground truth and predicted boxes. If the IOU is 1, it means that the ground truth and predicted boxes overlap and the detection is correct. Image size for YOLO v4 considered is Width = 512, Height = 512.

Table 2 shows the Precision, recall, and F1 score, appropriate selection of threshold plays a major role in distance estimation.

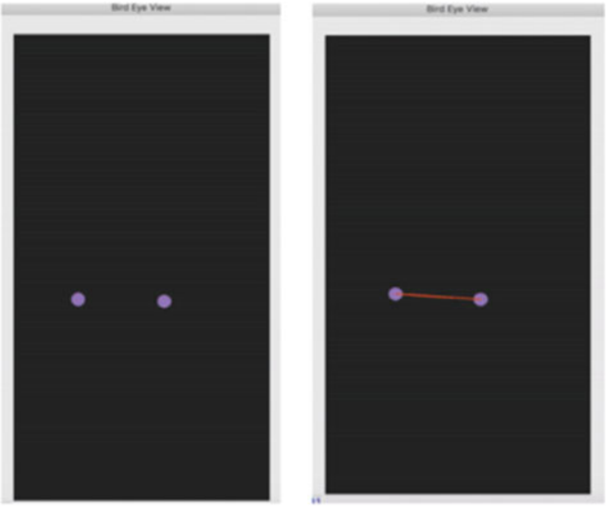


Fig. 14 The bird's eye view showing distance between two persons

Table 2 Precision, recall, and F1 score

No. of images (training)	Threshold (classes)	Precision	Recall	F1 score
608	25	0.66	0.73	0.68
608	80	0.90	0.42	0.54
320	25	0.64	0.60	0.61
320	80	0.89	0.60	0.61

5 Conclusion

This chapter gives a view of computer vision-based surveillance during pandemic using a perspective transformation. Perspective transformation gives more accurate distance calculations, thus the chances of the accuracy of the system in the detection increases. The methodology also suggests the use of cloud technology not only for the places with no computational power, but also for all places owing to the advantage of low cost and pay-as-you-go model in the use of cloud technology making the implementation cost-effective. Object detection and deep learning models uses birds eye view and YOLO v4 technique. Bounding boxes determines the actual distance and trigger the alert system.

Future work requires more generalisation of the algorithm where all possible angles of face will be covered for mask detection and the density of the crowd will be same as Indian market population density.

References

1. Amin, P.N., Moghe, S.S., Prabhakar, S.N., Nehete, C.M.: Deep learning based face mask detection and crowd counting. In: 2021 6th International Conference for Convergence in Technology (I2CT), 2021, pp 1–5 (2021). https://doi.org/10.1109/I2CT51068.2021.9417826
2. Punn, N.S., Sonbhadra, S.K., Agarwal, S.: Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLOv3 and Deepsort techniques. arXiv:2005.01385 (2020)
3. Khandelwal, P., Khandelwal, A., Agarwal, S., Thomas, D., Xavier, N., Raghuraman, A.: Using computer vision to enhance safety of workforce in manufacturing in a post COVID world arXiv:2005.05287 (2020)
4. Nguyen, C.T., Saputra, Y.M., Van Huynh, N., Nguyen, N.T., Khoa, T.V., Tuan, B.M., Nguyen, D.N., Hoang, D.T., Vu, T.X., Dutkiewicz, E., et al.: Enabling and Emerging Technologies for Social Distancing: A Comprehensive Survey. arXiv 2020. arXiv:2005.02816 (2005)
5. Yang, D., Yurtsever, E., Renganathan, V., Redmill, K.A., Özgüner, Ü.: A vision-based social distancing and critical density detection system for COVID-19. Sensors **21**, 4608 (2021) (2020). https://doi.org/10.3390/s21134608
6. Zhou, B., Wang, X., Tang, X.: Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012, pp. 2871–2878 (2012)
7. Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M.: YOLOv4: optimal speed and accuracy of object detection. arXiv:2004.10934 (2020)
8. Cristani, M., Del Bue, A., Murino, V., Setti, F., Vinciarelli, A.: The visual social distancing problem. IEEE Access **8**, 126876–126886 (2020)
9. Punn, N.S., Sonbhadra, S.K., Agarwal, S.: Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques. arXiv 2020. arXiv:2005.01385 (2020)
10. Zhao, Z.Q., Zheng, P., Xu, S.t., Wu, X.: Object detection with deep learning: a review. IEEE Trans. Neural Netw. Learn. Syst. **30**, 3212–3232 (2019)
11. Kumar, A., Kalia, A., Verma, K., Sharma, A., Kaushal, M.: Scaling up face masks detection with YOLO on a novel dataset. Optik **239**, 166744 (2021). ISSN 0030-4026. https://doi.org/10.1016/j.ijleo.2021.166744
12. Karaman, O., Alhudhaif, A., Polat, K.: Development of smart camera systems based on artificial intelligence network for social distance detection to fight against COVID-19. Appl. Soft Comput. **110**, 107610 (2021). ISSN 1568-4946. https://doi.org/10.1016/j.asoc.2021.107610
13. Zhao, Z.Q., Zheng, P., Xu, S.t., Wu, X.: Object detection with deep learning: a review. IEEE Trans. Neural Netw. Learn. Syst. **30**(11), 3212–3232 (2019)
14. Dataset. https://github.com/prajnasb/observations
15. Chowdary, G., Punn, N., Sonbhadra, S., Agarwal, S.: Face mask detection using transfer learning of inception V3 (2020). https://arxiv.org/2009.08369