

Using Computer Vision to Detect Violation of Social Distancing in Queues



Muhammed Ismail, T. Najeeb, N. S. Anzar, A. Aditya, and B. R. Poorna

Abstract On March of 2020, World Health Organization (WHO) declared COVID-19 as a global pandemic after months of infecting and claiming many victims. There are some ways by which we can safeguard ourselves against the virus and thereby controlling the spread of the virus. They are following proper sanitization by washing hands with soap regularly, wearing masks and following social distancing, while being present in public places. Social distancing refers to maintaining at least 6 feet of distance between other people. But the main problem is that most of the people ignore these rules and hence the spread of the virus can not be controlled. The project uses computer vision in order to ensure that social distancing is being followed properly, thus helping to reduce the number of victims that the virus may claim. Computer vision is a field of computer science that deals with how computers can gather knowledge and learn from images and videos. It is a rapidly growing field of science thanks to the many advancements in technology over the past few years such as increase in processing power of computers and the exponential increase in data being available nowadays. The system works by taking input from CCTV or other similar image source and then processing the input to find out if any people violate the rules of social distancing and if any violations are detected, the system will consist of an alert module which will alert the respective authorities regarding the violation so that they can do the needful.

Keywords Social distancing · Computer vision · YOLOv4

Supported by Mar Baselios College of Engineering and Technology.

M. Ismail (✉) · T. Najeeb · N. S. Anzar · A. Aditya · B. R. Poorna
Mar Baselios College of Engineering and Technology, Mar Ivanios College Rd, Nalanchira,
Thiruvananthapuram, Kerala 695015, India
e-mail: mbcet@mbcet.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022
S. Smys et al. (eds.), *Inventive Computation and Information Technologies*, Lecture Notes
in Networks and Systems 336, https://doi.org/10.1007/978-981-16-6723-7_57

771

1 Introduction

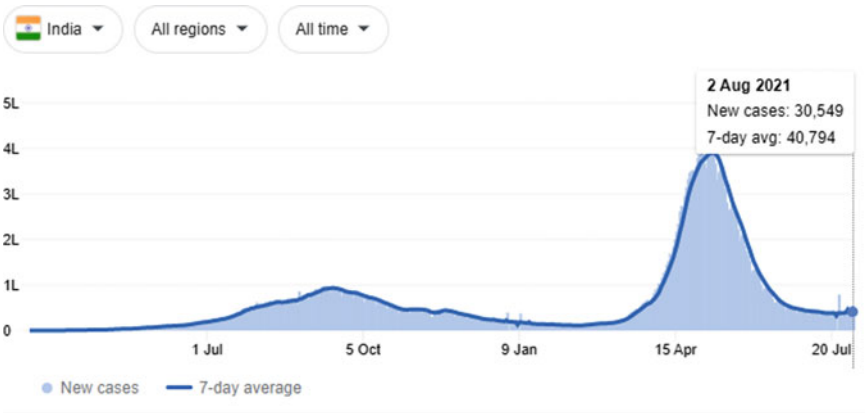
The first few cases of COVID-19 [1] started appearing in November 2019 in Wuhan province of China and by March 2020, the World Health Organization (WHO) declared COVID-19 as a pandemic, which refers to an outbreak or a disease that has crossed international borders and have affected many number of people. The virus spreads by means of contact which may be either direct or indirect, or by being in same breathing space as another infected individual.

Figure 1 shows the number of cases reported in India over the past month, and Fig. 2 shows the all time number of cases reported. From both the graphs, it is clearly visible that the daily cases being reported have still not gone down significantly after the second wave, while the death rate or the number of people who have died due to



Each day shows new cases reported since the previous day · [About this data](#)

Fig. 1 Graphical representation of reported cases of COVID-19 in India during past month



Each day shows new cases reported since the previous day · [About this data](#)

Fig. 2 Graphical representation of reported cases of COVID-19 in India of all time

COVID-19 has increased as the days passed by. There has been prediction of a Third Wave coming soon and if we take a closer look at the graph we can see that the peak has slowly started to rise again.

The Health department have issued many guidelines to follow so that the spread of virus can be controlled. Some of them are: Following proper sanitation by washing hands with soap regularly, only leave your homes in case of emergency and wear masks and follow social distancing, while being present in public places such as shops, railway stations, bus stands, banks, etc. But the main problem is that majority do not follow these guidelines properly, thus increasing the likelihood of spreading the virus.

The project aims to help the authorities reduce the spread of the virus by helping them find out if people are following the protocols or not and if any violation is detected, the system will alert the concerned personnel so that they can do the needful. We focus mainly on detecting violation of social distancing. Social distancing is a practice that requires an individual to maintain a minimum safe distance of at least 6 feet from others while being present in public places so as to reduce the risk of transmitting virus from one person to another.

This is done by using Computer Vision [2], which is a field of Computer Science that deals with processing images and videos and extracting data from these input so that it can be applied for solving real world problems. There has been many advancements in the field of Computer Vision mainly due to two reasons. One is the advancements happening in computational hardware that led to increase in processing power which further led to increase in processing speed and decrease in processing time. Another reason is the availability of huge amount of data which is way more compared to data available when computer vision started out as a concept thanks to the wide accessibility of both hardware and a medium to share data such as the Internet.

The project works by taking input from a source such as CCTV and processing the input to see if any pedestrians present in the input stream have violated the rules of social distancing by not maintaining the minimum safe distance of 6 feet and if any such violation is detected, the alert module present in the system will alert those concerned of the violation so that they can do the needful.

2 Literature Review

From the time when COVID-19 was declared as a global pandemic, there has been lot of research and study regarding the ways by which we the spread of COVID-19 virus could be controlled. Some researchers focused on a way to help develop the vaccine to the virus sooner by using deep learning and computational power, while some others focused on the socio-economic impacts due to the virus and on finding a way to bring back the society to the order and prosperity it enjoyed before the virus. Out of the many curious individuals, some of them also focused on social distancing and its effectiveness on reducing the spread of COVID-19 virus and developed certain systems for the same.

Ghodgaonkar et al. [3] focus on analysing how social distancing was carried out in different parts of the world and studying about how the various restrictions made by different governments affected the people's behavior in following social distancing. Rezaei and Azarmi [4] created a system which used YOLOv4 to develop a monitoring tool for social distancing and achieved an average precision score or AP score of 99.8 processing input feed. Yang et al. [5] is by far the system that shares most similarity with current proposal in the manner that their project detects breach in social distancing by using different object detectors such as Faster R-CNN, YOLOv3 [6], etc., and provides an audio/visual based alert system which gives out a general alert when the number of violations becomes moderate and a more severe alert when the number exceeds a threshold value.

The system's difference when compared to the already existing techniques mainly lies in the fact that it is intended to be deployed in public places where people gather such as shops, schools, railway stations, etc., and the project focuses on a particular type of scenario. The situation which we are trying to focus is that of a queue system such as ticket queue where people need to stand in a line for some time such as when visiting banks, waiting for checkout at shops, etc., and during this time, special care needs to be taken so as to avoid coming in close contact with others. This scenario needs to be addressed especially since in many states movie theaters have started functioning again which is one of the most common places where people usually stand in a queue. Thus, the proposed system aims to add additional functionalities to the already existing systems so that it could perform more tasks when deployed and do so with maximum efficiency.

3 Implementation

The system works by first taking input from an image/video source such as CCTV system or other similar alternatives. The output coming from the source will be either in .mp4 or .avi format. The input will be fed into the program using opencv which provides built-in functions for doing the same. The only problem is that the processing will be done in BGR color space, which also can be remedied by converting the color space of the video at a later stage. The video stream obtained will be processed for removing any noise or unwanted artifacts if present. This can be done using filters such as median filters. This processed input will be fed as input to an object detector for detecting pedestrians present in the input. The detector we chose for the system is YOLOv4 [7] since it runs extremely fast compared to other detectors such as Faster R-CNN [8] and Histogram of Oriented Gradients (HOG) [9], while not compromising on accuracy, hence a perfect candidate for applying to real world scenarios. After detecting the number of people present in the given input frame, distance will be calculated between each pair of pedestrian using a distance measure, which for the system we have decided to use Euclidean Distance measure. Then, we will check if any pair violate the minimum safety distance of 6 feet and for every violation present a timer variable will be maintained unique to that particular violation. If any timer

variable exceeds a threshold value, then that violation will be alerted to the concerned authorities so that they can take necessary action like warn them and remind them to follow COVID-19 protocols. A threshold value is set so that those contacts which may happen only for few seconds can be avoided and the system can focus on more severe violations.

3.1 Violation Detection Module

This is the main crux of the system. Firstly, we load the input video stream onto the program and then run it through YOLOv4 object detector. YOLOv4 is pre-trained to detect almost 83 classes of objects, but we are only looking for the pedestrians detected within the given input frame at any given time. So in the next step we filter out the detection to only those which fall under the label "person." We proceed to the next stage only if the number of detections is not null. From the output of detection done by the system, we get the x and y coordinates for the midpoint of the bounding box drawn around the person detected as well its width and height. From these points, we obtain the coordinates of the corner points of the bounding box, since they are needed later for drawing bounding box around the detected people in color, which will be green for those who are safe and red for those who are not 2 feet apart. After finding out the desired coordinates, they are added onto a dictionary with key being an integer initialized with 1 and incremented on addition of a new person onto the dictionary.

In the next stage, we find out all possible combinations of pedestrians present in a given input frame, with the number of pedestrians considered at a time set as 2, so as to find the distance between them using the combinations function provided in `itertools` package. From the dictionary that we created earlier, we calculate the difference between the x and y coordinates of the midpoint of the bounding boxes of the pedestrians currently in consideration and using this compute the distance between the said pedestrians using Euclidean Distance measure which was discussed in the previous section. Following step deals with checking if the distance is less than the minimum safe distance to be followed. Here, the minimum safe distance threshold is not a universal value and will vary according to the situation where the system is to used. This variance occurs due to the fact that when we use Euclidean Distance to calculate the distance measure between the pedestrians, we are actually calculating the number of pixels between them, which will vary according to the resolution in which the video is displayed, the location of the camera, while taking the video, etc. Once the system has been setup and the distance threshold has been identified manually, further detection of violations will happen automatically. If the pedestrians are not maintaining the minimum safe distance, they will be added to a separate list which indicates that they are in danger.

After creating the list which contains the key value and other necessary details of pedestrians from the original dictionary, the next step is to draw bounding box around them and display them in an output window. For this, we have already found

out the necessary coordinates for drawing the rectangular box and we input these coordinates into a function provided by OpenCV library along with other parameters such as thickness of the line, color of the lines, whether it be green for those who are safe or red for those who are in danger, etc. The total number of violations detected at any given time will also be displayed in the output window.

3.2 Alert Module

For the alert module, we make use of an Arduino Uno Microcontroller Board because of its functionality, support network and easiness to work with. The board consists of both analog and digital pins, although for this system we only require the use of digital pins. Normally, the boards manufactured by Arduino are used in conjunction with the IDE that they provide. But the IDE supports only C and C++ and the project is written entirely in python. So in order to interface the Arduino Board with the program, we used the PySerial package which helps programs to access the serial port. By specifying the port to which the Board is connected in the system, we can create a channel for communicating with the Board. Due to the COVID-19 and subsequent Lockdown restrictions, the initial plan of using an LCD display along with a buzzer system failed because of the lack of resources available. Instead we repurposed the alert module with the help of basic LEDs. The module will now blink the LED's connected to it when the number of violations have exceeded a predefined threshold value.

4 Results and Discussion

Before discussing about the results, there are certain factors that should be considered. The measure that we used to evaluate the system is the Average Frames per Second or FPS, which refers to the number of frames being processed in one second. This is because in real world situations, We gave frame rate more importance compared to other measures because depending upon the frame rate by which the input video is being processed, the output can be either fast enough to keep up with actual use case situations or be slow enough that it is not possible to use it for getting real time output. Another factor to consider is the system which is being used for running the project. Depending upon the processor of the system, the availability of accelerators such as GPU's and the type of GPU's used, the rate of processing videos differs rapidly with more powerful ones being the better option. The configuration of system we used for testing is shown in Table 1.

Using a system having the above specifications, we were able to achieve an Average Frame Rate of 25 FPS, which is considered an above acceptable standard compared to the 30 Frames per Second normally used for real time processing.

Table 1 Specification of system used for testing and evaluation

Processor	AMD Ryzen 7 3700x
Number of Processing Cores	8
RAM	32 GB DDR4 3200MHz
GPU	NVIDIA RTX 2070 Super
Number of CUDA Cores	2560

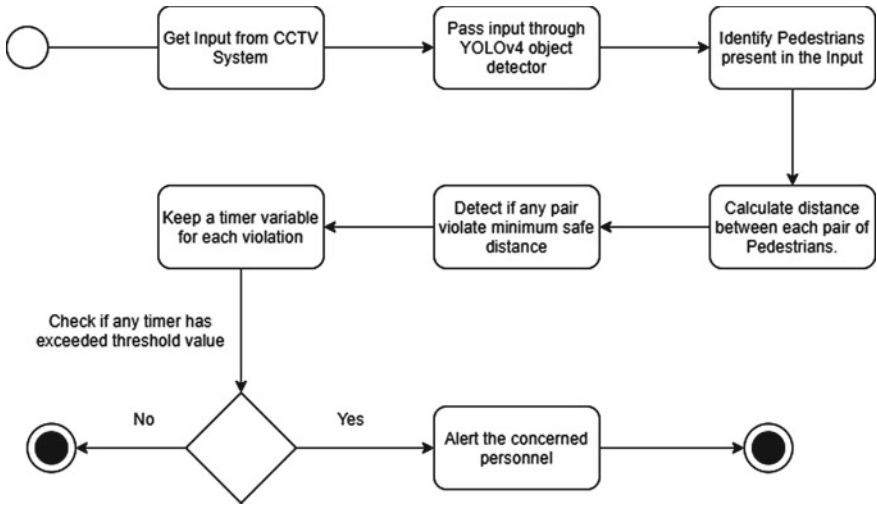


Fig. 3 Activity Diagram



Fig. 4 Output when minimum safe distance is maintained by the people



Fig. 5 Output when a violation is detected by the system

Figure 2 shows an example of a situation where everyone present in the given frame is maintaining the minimum safe distance to be kept. Hence, they are covered by green colored bounding boxes. In Fig. 3, 2 out of the 3 people present in the given frame are not maintaining the safety threshold distance between them and are thus in violation of social distancing. Hence they will be covered in red bounding box to indicate that they are in violation and danger (Figs. 4 and 5).

5 Conclusion

The expected output of the system is that it will be able to identify most if not all the people present in the given input and it is able to do so with a minimum required accuracy. It will also be able to calculate the distance between each pair of pedestrians detected, while accounting for real world parameters such as the angle and position at which the CCTV or input source is situated and the calculated distance value must be approximately equal to the real-world distance between them, provided within an error range of 0.5–1.5 feet. The system would also keep track of each violation and alert the authorities if any violation has exceeded the pre-set threshold value which normally would be set to 10s.

The expectation of developing the system is that the authorities such as health officers, police officers and the concerned people such as shop owners can use the system in order to control the virus from spreading and claiming a greater number of people than it already has. If we take the case of China where the virus first originated, they were able to contain the virus within 9–10 months because the government

enforced COVID-19 protocols strictly and the citizens also abided by the same. The expected outcome from using the system is that it helps the society to also do the same and break out from the grasp of COVID-19 virus or at the least reduce the rate at which the virus spreads until a proper vaccine has been developed, tested and made available to the public.

References

1. T.P. Velavan, C.G. Meyer, The COVID 19 epidemic. *Trop. Med. Int. Health* **25**(3), 278 (2020)
2. D. Barik, M. Mondal, Object identification for computer vision using image segmentation, in *2010 2nd International Conference on Education Technology and Computer*, vol. 2 (IEEE, 2010), pp. V2–170
3. I. Ghodgaonkar, S. Chakraborty, V. Banna, S. Allcroft, M. Metwaly, F. Bordwell, K. Kimura et al., Analyzing Worldwide Social Distancing through Large-Scale Computer Vision (2020). arXiv preprint [arXiv:2008.12363](https://arxiv.org/abs/2008.12363)
4. M. Rezaei, M. Azarmi, Deepsocial: social distancing monitoring and infection risk assessment in covid-19 pandemic. *Appl. Sci.* **10**(21), 7514 (2020)
5. D. Yang, E. Yurtsever, V. Renganathan, K.A. Redmill, Ü. Özgüner, A Vision-Based Social Distancing and Critical Density Detection System for Covid-19 (2020), pp. 24–25. arXiv preprint [arXiv:2007.03578](https://arxiv.org/abs/2007.03578)
6. J. Redmon, A. Farhadi, Yolov3: An Incremental Improvement (2018). arXiv preprint [arXiv:1804.02767](https://arxiv.org/abs/1804.02767)
7. A. Bochkovskiy, C.-Y. Wang, H.-Y.M. Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection (2020). arXiv preprint [arXiv:2004.10934](https://arxiv.org/abs/2004.10934)
8. W. Wu, Y. Yin, X. Wang, X. De, Face detection with different scales based on faster R-CNN. *IEEE Trans. Cybern.* **49**(11), 4017–4028 (2018)
9. N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1 (IEEE, 2005), pp. 886–893