

Ensuring social distancing using machine learning

Gaurav Mathurkar¹, Chinmay Parkhi¹, Manish Utekar¹ and Ms. Pallavi H. Chitte¹

¹Department of Computer Engineering, Ramrao Adik Institute of Technology, India

Abstract—COVID-19 (Coronavirus) has affected our daily life and is slackening the global economy. In this fight against the COVID-19, social-distancing is proving to be an effective measure to slow down the transmission of coronavirus. In order to control the spread of transmission of COVID-19, social distancing is the best measure which aims to avoid close contact of people. In order to control the spread of transmission of COVID-19, social distancing is the best measure which aims to avoid close contact of people. With the aim of ensuring social- distancing norms in workplace and public places, we can develop a social-distancing detection tool that monitors if people are staying at a safe distance from each other. This can be done by examining real-time video streams from the cameras. In order to provide an effective solution, we are developing a model which has mainly two phases - firstly object detection phase where people will be detected from the video streams. Secondly showing the statistical analysis in the form of dashboards and sending alerts to concerned authorities to take necessary actions.

Keywords— COVID-19, object detection, YOLO V3, alerts

I. INTRODUCTION

Social distancing is an important step to avoid spreading COVID-19 virus which involves increasing the physical space between people. Implementing social distancing during the ongoing global pandemic is an important step. Maintaining a particular distance from other people lessens your chances of contracting COVID-19. Businesses need to ensure that social distancing policies are followed to keep customers, workers safe. Determining the exact distance between people in any workplace or location can be challenging and also resource consuming. Social-distancing is an effective practice to help reduce the contacts with the people affected with COVID-19 outside the home. Anyone can get and spread COVID-19, the risk of serious illness may or may not be different for everyone. So it's a duty of each person in the society to follow social distancing norms as per mandated by Government officials which results in protecting the entire community. To support government efforts and the given public health guidelines, a Social distancing detector that helps the concerned institutions to determine whether people are following social distancing guidelines can be developed. The system provides a detailed analysis of the situation through the dashboard, thus helping the relevant authorities to obtain an accurate analysis of the situation in the Region of Interest (ROI). While work from home for office work is advisable practice to scale back the transmission of coronavirus, some people don't have that option and that they may need to run bent the grocery, pharmacies or the other public areas. In these situations, it'll be important to take care of social distancing norms so as to take care of a healthy society and perform the activities productively.

The main functions include:

- ❖ Generate statistical analysis for the required region
- ❖ Sending alerts (if social distancing norms are violated above a particular threshold) to the respective authorities to take actions as per requirement.

We can use OpenCV and YOLO to monitor whether people are maintaining social distancing or not.

II. BACKGROUND STUDY AND RELATED WORK

In order to provide a solution for effective social distancing management, several approaches have been proposed. Among noteworthy systems some of the systems are analyzed and considered for our system development. Using Faster R-CNN and yolo v4 a system was proposed to detect people in a region of interest and measure the interpersonal distances in real time without data recording to protect the privacy of individuals which can be misused intentionally or unintentionally [3]. Using Yolo v3 and deep sort techniques a system was developed using three object detection models and a performance comparison was done for the three object detection models like Faster R-CNN, SSD and Yolo v3 [2]. Refer the following figure 2.1 for performance comparison of different object detection models as discussed in the paper [3].

Model	TT(in sec)	mAP	NoI	TL	FPS
SSD	9651	0.969	12135	0.02	3
Faster RCNN	2124	0.691	1200	0.22	10
YOLO v3	5659	0.846	7560	0.87	23

Table 1: Comparison of different object detection models

From the Table.1 we can notice that YOLOv3 can process at higher number of frames per second (fps) as compared to SSD and Faster RCNN. Using architecture of 19 layers a model named SocialdistancingNet-19 was developed and provided the solution for maintenance of social distancing by showing the number of violations in the area. The calculated accuracy of the model was that the maximum accuracy was estimated as 93.3 and the minimum accuracy was estimated as 74.3% [1]. Using Passive Infrared Sensors (PIR) which are used to detect human movement when it comes in a particular range, actual design was implemented using raspberry pie microcontroller and the proposed device created a alert on the mobile device using ML algorithm to determine the possibility of covid patients around a person

[4]. In the paper [6] a solution is proposed for developing a mobile application for monitoring with the help of mobile camera with 85% accuracy using Single Shot Detection model.

III. PROPOSED APPROACH

The proposed work includes usage of yolo v3 which is a fully convolutional neural network algorithm to detect the people in the video frames. OpenCV which is a real-time computer vision library is used to feed the input framed from the images or videos captured by the cameras to the yolo v3 neural network. The system will then provide the results for the particular region and provide a statistical analysis of the region captured by the cameras.

Yolo v3 which is capable of recognizing 80 different objects in given input images and videos is used for the object detection phase in the algorithm. Firstly, Yolo v3 is having 53 layer network trained on Imagenet. It uses a variant of darknet. But for the purpose of detecting object, more 53 layers are used on to the previous set of layers. Thus yolo v3 has total 106 layers in its network. The basic design of Yolo v3 is given in figure 3.1.

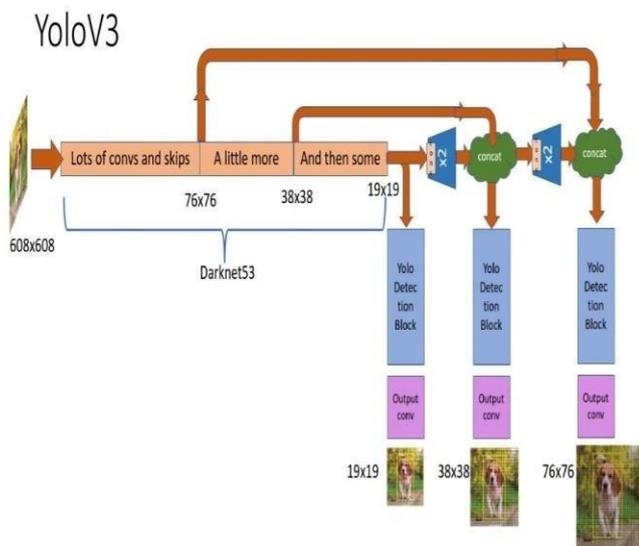


Fig 3.1:YOLO V3 architecture

The yolo v3 algorithm breaks the given input image into an $S \times S$ grid. A grid cell will be accountable for detection of object if the center of the object is present within the grid cell. To predict multiple labels for an object a threshold is used and for predicting each class score logistic regression is used by the algorithm. Classes that have scores higher than 50% which is considered as a confidence parameter to filter out the object that belongs to "people" class in the image. The interpersonal distance between the people is calculated using the distance between centroids i.e. the distance between the bounding boxes within the image. If the distance is found to be greater than a particular threshold then it is considered as a social distancing norms violation and displayed as unsafe with red bounding box in the image. If the social distancing norms are followed then the safe people are displayed with green bounding box.

Mathematical equations required for the system:
 The yolo v3 network predicts four coordinates for each bounding box.

$$z_x = \sigma(s_x) + m_x \dots \dots \dots (i)$$

$$z_y = \sigma(s_y) + m_y \dots \dots \dots (ii)$$

$$z_h = q_h e^{s_h} \dots \dots \dots (iii)$$

$$z_w = q_w e^{s_w} \dots \dots \dots (iv)$$

where z_x, z_y, z_h, z_w are the x and y center co-ordinates, height and width of our prediction and s_x, s_y, s_h, s_w , are the grid outputs. m_x and m_y are the upper left coordinates of the network. q_h and q_w are anchors dimensions of the box [14].

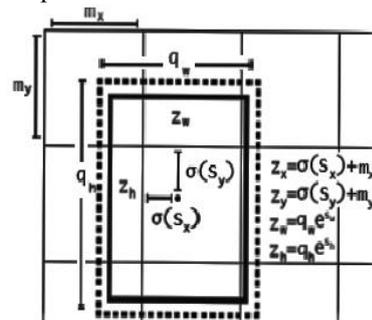


Fig 3.2:- Location prediction with coordinates in Bounding Boxes.

To find the distance between the centers of each bounding boxes which represent a person we use following formula,

$$\sqrt{(X_{\max} - X_{\min})^2 + (Y_{\max} - Y_{\min})^2}$$

The distance between the two people that is represented by the centers of the bounding boxes is given by the above equation.

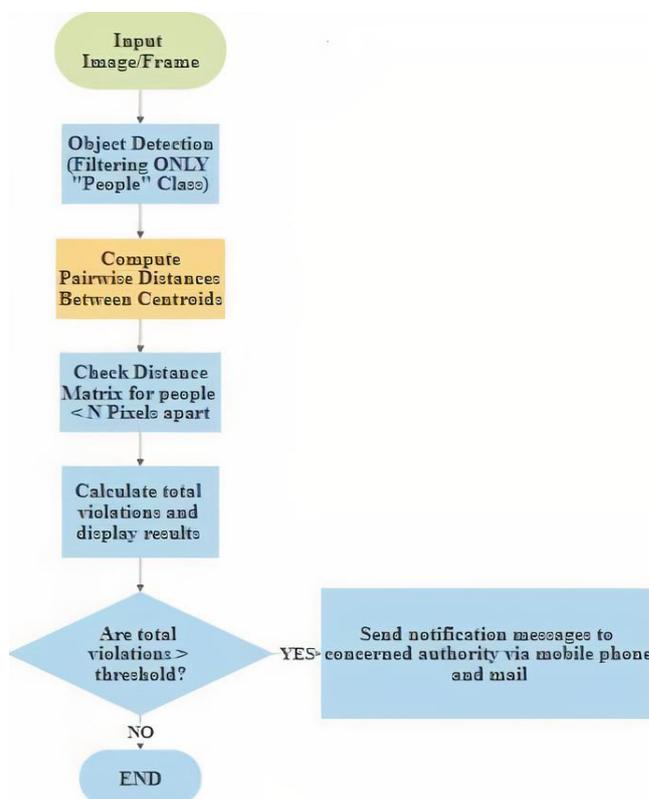


Fig 3.3 : System Design

IV. IMPLEMENTATION DETAILS

In order to process the frames and detect "people" class from the processed frames, some steps are to be followed that can be illustrated in the following flowchart. The flowchart gives a brief overview about each step in the system.

STEP 1:

The system takes an image or a frame from a real time video as an input using OpenCV. Since it wouldn't be very productive the system don't run the process for every frame. The video frame then will be processed to create a blob which is required for preprocessing our frame. The blob from Image can be used to process the frame which performs mean subtraction, scaling and swapping functions on the given image. After preprocessing the image will be provided to the Yolo neural network which then detects the objects within the image. For detecting the objects that belongs to the "People" class, a confidence of 50 % is set to filter out the objects with "People class" whose confidence is greater than 50 % as detected by our Yolo neural network.

STEP 2:

After detection of persons as an object, the system will treat this detections as a bird's eye view image as shown in figure. Further for transformation the system would select four points on the resultant image that are going to be the corner points of the image. With the two opposite sides being parallel, a rectangle will be formed with the help of those points.

STEP 3:

For each person detected, the system returns two points that are required to build a bounding box. These two points are nothing but the upper left corner and lower right corner of that box. Then from these the centroid of the box will be calculated by first calculating the midpoint of the line joining these two opposite sides. Then the distance between two centroids (i.e. Distance between two persons) is calculated with the threshold set as 2 feet or approximately 120 pixels.

STEP 4:

Using Matplotlib which is a plotting library we can show our results in the form of pie charts with detailed analysis of number of violations in the particular region as captured by the cameras on a dashboard adjacent to the video stream. We can provide total safe and unsafe number of violations which

can provide statistical analysis for the particular region.

STEP 5:

The system is designed in such a way that if total number of violations are exceeding a particular predefined threshold then an alert email will be sent to the respective authority with total active violations at a particular point of time, which allows the concerned authority to keep a track for the particular area easily. The smtplib module in python which defines the SMTP client object is used to send the email.

STEP 6:

At the same time of sending email, a bulk SMS message will be sent to the registered mobile numbers of the respective authorities when the total number of violations are exceeding a particular predefined threshold. The SMS Gateway API (Fast2SMS) is used to send messages to mobile phones using python.

V. ALGORITHM

1. Taking video file as input using cv2.VideoCapture()
2. Setting up the YOLO v3 network
3. Detecting the "person" class in the current video frame and using cv2.dnn.blobFromImage() for performing mean subtraction, scaling and swapping functions.
 - 3.1. If confidence is greater than 50% then proceed to step 4.
4. Calculating the co-ordinates of bounding boxes.
5. Using non maximal suppression for removing overlapping boxes
6. Calculating distance between two people using the formula
 - 6.1. If distance < 120 pixels:
Increase the unsafe people count
 - 6.2. If distance > 120 pixels:
Increase the safe people count
Show the results on the dashboard
7. If unsafe people count > predefined threshold:
Send a mail using smtplib module and text message using Fast2SMS API to authorities.

VI. EXPERIMENTS AND RESULTS

The yolo v3 algorithm used in this system detects "people" class from its set of 80 different classes from the frames as seen in the figure 5.1 and figure 5.2. The system processes with frame rate of 1.39 fps. The system is tested using video streams and images. Each frame captured from the video are labelled as safe and unsafe according to the distance between the people. A dashboard is shown with total number of safe and unsafe people. A pie chart is shown on the dashboard which helps in getting a statistical view for a particular area. The pie chart shows safe percentage of people in an area using blue colour and unsafe percentage of people in an area using orange colour. Currently the threshold value (i.e. maximum number of violations allowed in captured area) is set to 7.

The application detects people who are close by and not adhering to the distancing norms and marks them in RED bounding boxes, signifying risk. Others are marked in GREEN.

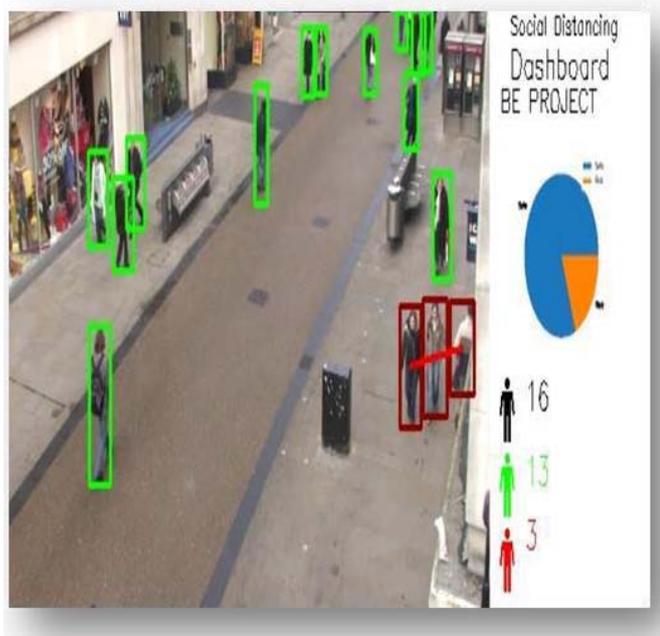


Fig 5.1: Social distancing violations < Threshold value



Fig 5.2: Social distancing violations > Threshold value

If total number of violations are exceeding a particular predefined threshold then an alert email is sent to the higher authority in-charge with total active violations at a particular point of time.

Also if the total number of violations are exceeding a particular predefined threshold, then an alert text message is sent to the concerned sub-authorities.

VII. CONCLUSION

The proposed system provides a systematic approach for tracking social distancing violations. It is possible to focus on areas where social distancing is not maintained and get statistical analysis for the region of interest (ROI) in the form of a visual representation through the dashboard. In order to alert the concerned authorities about the violations in a particular area, the system focuses on sending alert messages to the respective authorities after the particular number of violations(threshold) is exceeded in particular area. This will help the concerned authorities to track the social distancing norms violations easily and take actions as per the requirement. This proposed model is beneficial rather than physical monitoring which reduces the human efforts required and giving a vast coverage of areas for detection purposes. The system can be used in most of the public places such as railway stations, religious places, marketplaces and hospitals to minimize the spread of any contagious disease. Thus the proposed model is useful for giving alerts to maintain social distancing in this pandemic.

REFERENCES

- [1] Ninad Mehendale, Rinkal Keniya, Real-time social distancing detector using Social-distancing-Net-19 deep learning network, Scribd, (2020), <https://www.scribd.com/document/481289110/social-distance-detector>.
- [2] N. Singh Pun, S. Agarwal, S.K. Sonbhadra, Monitoring Covid-19 social-distancing with person-detection and tracking via fine-tuned YOLOv3 and Deep-sort techniques, arXiv, (2020), <https://arxiv.org/abs/2005.01385>
- [3] V. Renganathan, D. Yang, E. Yurtsever, U. Ozguner, K.A. Redmi11, A vision based social-distancing and critical density detection-system for COVID19, Researchgate, (2020), https://www.researchgate.net/publication/342763051_A_Vision_based_Social_Distancing_and_Critical_Density_Detection_System_for_COVID-19
- [4] P. Whig, S.M. Mohammad, R.R. Nadikattu, Novel economical social-distancing smart device for COVID19, International Journal of Electrical Engineering and Technology(IJEET,) (2020), https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3640230
- [5] S. Gawde, D. Kalbande, A. Ghorai, Digital solution for enforcing social-distancing, SSRN 3614898, International Conference on Innovative Computing & Communications (ICICC), (2020), https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3614898
- [6] Dr. Neelavathy Pari S , Geetha A V , Jeevitha V K, Balaji Vasu, Monitoring Social-distancing by Smartphone App in the Effect of COVID19, IJERTV9IS090469, (2020), https://www.researchgate.net/publication/344536998_Monitoring_Social_Distancing_by_Smart_Phone_App_in_the_Effect_of_COVID-19
- [7] R. Zust, M. Bielecki, Z.G. Stanga, D. Siegrist, D. Meyerhofer, A. Stettbacher, T.W. Buehrer, G.A.G. Cramer, J.W. Deuel, Social-distancing alters the clinical course of COVID19 in young adults: A comparative cohort study, Clinical Infectious Diseases, PubMed, (2020), <https://pubmed.ncbi.nlm.nih.gov/32594121/>
- [8] Pratibha Rathi, Pranav Adarsh, Manoj Kumar, YOLO v3-Tiny: Object Detection and Recognition using one stage improved model, IEEE, (2020), <https://ieeexplore.ieee.org/document/9074315>
- [9] S Yana, Dedi Satria, S. Syahreza, Rizal Munadi, Implementati on of SMS-Gateway in the Flood Notification System, Researc hgate, (2020), https://www.researchgate.net/publication/340490994_Implementatin_of_SMS_Gateway_in_the_Flood_Notificatin_System_using_Raspberry_Pi
- [10] Shuaiyang Li, Liquan Zhao, Object Detection Algorithm Based on Improved YOLOv3, Researchgate, (2020), https://www.researchgate.net/publication/340145010_Object_Detecti on_Algorith m_Based_on_Improved_YOLOv3
- [11] Dr. T. Uma Devi, Sakshi Gupta, YOLOv2 based Real Time Object Detection, IJCST, (2020), <http://www.ijcstjournal.org/volume-8/issue-3/IJCST-V8I3P4.pdf>
- [12] Shuyuan Yang, Fan Zhang, Licheng Jiao, Fang Liu, Rong Qu, Lingling Li, Zhixi Feng, A Survey of Deep Learning-based Object Detection, arxiv, (2019), <https://arxiv.org/pdf/1907.09408.pdf>
- [13] Chulhee Lee, Youhak Lee, Jin-Sung Kim, Hyuk-Jae Lee; Fast Detection of Objects Using a YOLOv3 Network for a Vending Machine, IEEE, (2019), <https://ieeexplore.ieee.org/document/8771517>

- [14] Farhadi, Ali., Redmon Joseph, YOLOv3: An Incremental Improvement, arXiv:1804.02767,(2018), <https://arxiv.org/abs/1804.02767>
- [15] Tareq Abed Mohammed; Saad Albawi; Saad Al-Zawi, Understanding of a convolutional neural network, IEEE,(2018), <https://ieeexplore.ieee.org/document/8308186>
- [16] Varsha Pathak, Manish Joshi, A survey of SMS based Information Systems, Researchgate,(2015), https://www.researchgate.net/publication/277334889_A_survey_of_SMS_based_Information_Systems.
- [17] Deva Ramanan, Zitnick Piotr Dollar, Lin Michael, Tsung-Yi, Belongie Lubomir, Maire Serge, Bourdev Ross, James Hays, Girshick, Pietro Perona, C. Lawrence, Microsoft COCO (Common Objects in Context),arxiv,(2015),<https://arxiv.org/pdf/1405.0312.pdf>
- [18] M. Gordon, "An introduction to network-programming the Pythonway", IEEE, (2005), <https://ieeexplore.ieee.org/document/1541900/authors#authors>