Social Distance Detector Using Computer Vision, OpenCV and YOLO Deep Learning Algorithm

Suyash Paradkar[1], Shreya Tekade[2], Komal Ganatra[3], Saloni Sarwade[4], Prof. Rahul Bhole[5]

[1,2,3,4]B.E. Students, Department of IT, Zeal College of Engineering and Research, Maharashtra, India
[5]Professor, Department of IT, Zeal College of Engineering and Research, Maharashtra, India

---

Abstract – *This paper presents a method of detecting social distance using concepts like computer vision, OpenCV and YOLO deep learning algorithm among people to detect and control the spread of infectious disease like corona virus (covid-19). By using a pre-recorded video as input, the computer screen as output stream and OpenCV using YOLO algorithm we can tell if people are following social distancing or not. These types of implementations have been successful but due to speed issues of the computer on which these models are run, we can generate accurate results.*

Hence by using CUDA we can increase the performance of the model by utilizing the computing power of GPU which is available in most of the systems. It has great uses in applications like CCTV cameras, security cameras, web camera-based systems etc., and it can detect activity of people in real time. It has great use in automation side.

Our system is able to monitor people and tell us whether they are following the rules of safe distancing or not. It can detect whether distance between 2 people is at least 1 meter or not.

Keywords: Social Distance Detector, Computer Vision, OpenCV, YOLO, CUDA

1.INTRODUCTION

Covid-19 is caused by the SARS-CoV-2 virus. It can cause different ranges of problems ranging from mild fever to sever respiratory disease, which can even lead to death. The corona or covid-19 pandemic has created so much problem all across world and even after countries succeeding in vaccinating 50% of their population, still covid-19 spread is not stopping. People are being advised to wear masks and maintain social distance among them but they seem too careless for the same. Governments across the globe have made it mandatory to follow a minimum distance of 1 meter among them and to make sure that it is being followed this paper can provide a robust and fast solution. Especially we have to monitor those places in which there is predicted large number of people like hospitals, public gathering places, religious places, fairs etc. In these places there is less chances that people will follow social distancing rules hence we need to implement this model in these areas. With the help of CCTV cameras or files we can detect different people distances and henceforth can inform the designated authority about the same. This way people can be fined and punished accordingly.

Fig -1: Social distancing is important to stop covid-19

2. PROBLEM STATEMENT

To control the spread of disease which are infectious like covid-19 we use many measures but major one is social distancing which is a nonpharmaceutical way of avoiding this disease. In this paper we propose a model in which we perform real-time analysis and detection of people to see if social distancing is being followed or not? By using CUDA we also make use the graphical power of graphic cards which have dedicated memory to perform daunting tasks easily without putting much load on CPU. Below we present the solution we used for this particular model.

3. REQUIREMENT ENGINEERING / ANALYSIS

3.1 Python Programming Language

Python is high-level, general-purpose programming language. Python's code emphasizes on code readability, reusability with the use of very easy to use, maintainable indentation.

It is dynamically typed and it is garbage collected. It has very useful libraries for various stuff but it's mostly famous for it's useful and easy machine learning modules.

3.2 Computer Vision

Computer Vision is an interdisciplinary scientific field which is evolving with time that deals with how computers and other related machines can gain high-level understanding from digital images or videos. It includes different methods which includes concepts like acquiring, processing and analyzing digital images and videos and it is also used for extraction of high-dimensional data from real world scenarios in order to produce numerical or symbolic information which can be used by computers to process upon.

Fig -2: Green box shows distance > 1 m

Fig -3: Red box shows distance < 1 m

3.3 OpenCV

OpenCV is a special library mainly aimed at real-time computer vision. It is a cross-platform and free to use as it is open source. OpenCV also has special features like GPU acceleration for real-time operations in machines like computers. OpenCV CPU version is 9x faster. If we use DNN module implementation of OpenCV then it is faster. To judge speed of OpenCV, OpenMP takes 2 seconds when used with Dark net but OpenCV implementation takes mere 0.22 seconds! It has various use cases like object detection, landmark detection, face detection, tracking eye movements, extracting 3D models of objects, tracking modern camera movements etc.

3.4 YOLO for object detection

YOLO stands for You Only Look Once. It is extensively used in field of object detection computer technology which is related to computer vision and image signal processing. It is used to detect instances of class objects like humans, buildings, cats, dogs, etc. It is used in locations like face detection, pedestrian detection, video surveillance, image retrieval etc.

4. PROPOSED SYSTEM

Our proposed system is a four-level module that focuses on things like human identification, their tracking and then after their distance detection (mainly humans using YOLO object detection algorithm) and calculation using computer vision and deep learning by using OpenCV.

We will follow the following criteria given below: -

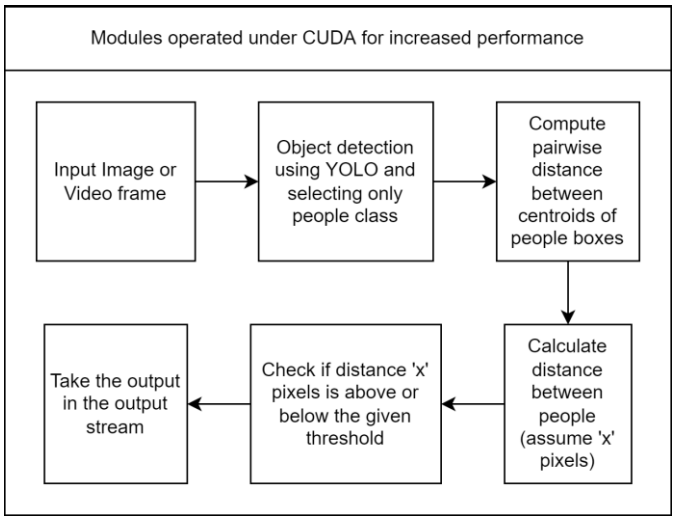
4.1 Pass the input which will be image or video and we will pass it frame by frame.

4.2 Detect the human object in the image or video frame using YOLO object detection.

4.3 Compute the pairwise distances between centroids of boxes drawn on the people.

4.4 Check if the distance between any two persons is 'x' pixels and check whether it is above or below the given threshold.

4.5 Generate the output in the output stream.



Flowchart -1: Workflow of our proposed system

5. IMPLEMENTATION

5.1 Image / Video Input

First, we provide image / video input to our model which will extract it frame by frame. When we will use videos then no. of frames per seconds will be much more hence it will be difficult to process hence, we use CUDA powered system so that processing is done at faster rates. It will be beneficial to use high performance GPUs for better results.

5.2 Object detection

Now we are going to detect people object using YOLO. We will only use people class filter to filter only people out of many classes available. This implementation is done using Darknet which is a neural net implementation and open

source. YOLO looks at the entire image only once and then goes through the network once and detects objects.

5.3 Distance measurement between person

After detecting people available in the frame provided as input, the model is going to detect the person and draw different bounding boxes on the person. So, this can create many boxes around the person and this can be avoided by using an algorithm called as Non-maximum suppression (NMS). NMS is going to take into consideration the box which will have highest probability of covering that person and hence only one box will be drawn around the person. After this step we are going to use this box for further calculations

(I) After the bounding box is drawn, we are going to find centroid of bounding box.

(II) After finding centroid, we are going to find distance between two these two centroids which will find the distance between the bounding boxes.

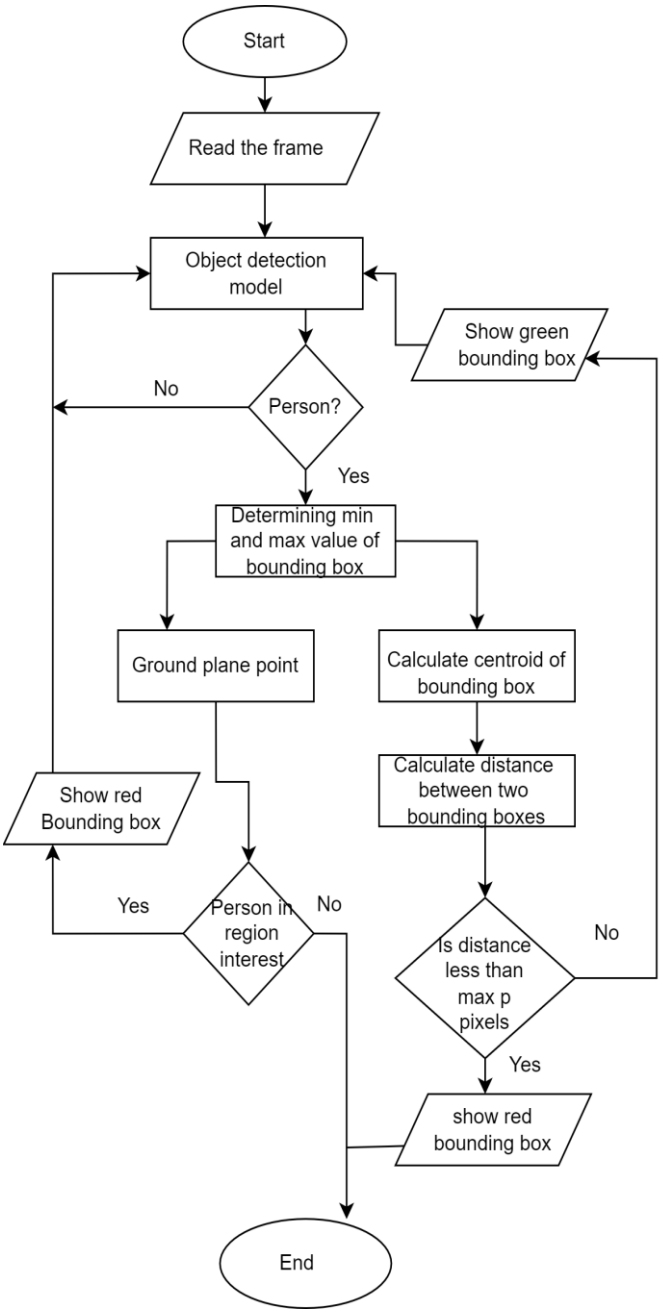
(III) After finding distance we are going to compare this distance with 'x' pixels distance already predefined and check whether it is above or below the mentioned limit.

(IV) Add the total combinations of boxes which are not following social distance norms and put it in some storage class. We will use this afterwards to display real time number of people disobeying social distancing rules.

5.4 Output the results

After storing the number of comparisons falling below threshold value, we are going to add these up and display it under violation label. We are going to combine the frame in case of video stream and because we are using CUDA, we expect to get the results faster than conventional methods. The output stream can be anything like computer screen, monitors, web displays, TVs, etc.

6. FLOWCHART



7. FUTURE SCOPE

We can make a dashboard to display various different parameters along with number of people disobeying social distancing norms. We can display data like frequency of having maximum population in specific time intervals and getting the time in which there is maximum violation of social distancing. Also, we can improve the camera calibration or take that images or videos as input in which people are clearly visible in frame to get more accurate results. We used boxes to calculate centroids and then calculated distances between them but if we want more accuracy, we can take top-down approach instead of boxy approach. This way we can calculate circle centers and find distances between them so

that we can obtain better distance approximations among different combinations of people.

8. CONCLUSIONS

This paper used computer vision, deep learning algorithms like YOLO and OpenCV to reduce the impact of covid-19 by accessing distance between people and if any pair fails to follow the social distance norm then we will indicate it with red line and this all will be done in real-time and because we are using CUDA, we will get faster results than traditional methods. Our proposed method demonstrated this by using a pre-recorded video of people walking on a road. This model can be used in other places like hospitals, offices, public areas etc. Also, some improvements can be done in the model to get more functionalities and improve performance.

9. REFERENCES

[1] Sadettin Melenli and Aylin Topkaya, "Real-Time Maintaining of Social Distance in Covid-19 Environment using Image Processing and Big Data", IEEE

[2] Payal Panchal, Gaurav Prajapati, Savan Patel, Hinal Shah and Jitendra Nasriwala, "A Review on Object Detection and Tracking methods", International Journal for Research in Emerging Science And Technology, Volume-2, Issue-I, January 2015

[3] Yew Cheong Hou, Mohd Zafri Baharuddin, Salman Yussof and Sumayyah Dzulkifly, "Social Distancing Detection with Deep Learning Model".

[4] Shashi Yadav, "Deep Learning based Social distancing detection", International Journal for Research in Applied Science & Engineering Technology (IJRAET), vol. 08, no. 07, pp. 2321-9653, 2020.

[5] M. Andriluka, S. Roth and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking", 2008 IEEE Conference on computer vision and pattern recognition, pp. 1-8, 2008.

[6] B. Benfold and I. Reid, "Stable multi-target tracking in real-time surveillance video", IEEE Conference on Computer Vision and Pattern Recognition, pp. 3457-3464, 2011.