e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:03/Issue:09/September-2021 Impact Factor- 6.752 www.irjmets.com

SOCIAL DISTANCING DETECTION USING TENSORFLOW

Reddiar Karthikeyan[*1], **Ms. Jismy Joseph**[*2]

[*1]Student, Department Of Computer Applications, SCMS School Of Technology And Management, Muttom, Aluva, 683106.

[*2]Professor, Department Of Computer Applications, SCMS School Of Technology And Management, Muttom, Aluva, 683106.

ABSTRACT

The outbreak of COVID-19 has caused great frustration for everyone all around the world. To stop the disease from spreading people are told to follow proper social distancing from one another. But there are still many people who do not care about COVID and break the rules by not keeping the public distance. During the project we make a python-based program using TensorFlow to check whether people are keeping the specific distance between each other or not. This would help Government to monitor areas where proper maintenance of COVID protocols can take place and take appropriate action if not followed. To make this we will be using Video file as an input. When two people are very close, a red rectangle is shown around them indicating that they are not following a social distance.

Keywords: Tensorflow, Image Classification, Opencv, Social Distancing, Python Object Detection.

I. INTRODUCTION

COVID-19 outbreak can be reduced by following the protocols set by the government. One of the most important protocol is to follow Social distancing. The increase the affects of this disease can be decreased or spread only if the social distance is maintained which will stop the spread from one to another. But there are people who do not follow social distance. So, a system is needed to automatically identify or monitor this. This is done by using the Tensor flow object detection model and OpenCV.

II. METHODOLOGY

SELECTIONG A MODEL

For this project we will be using TensorFlow object detection model. It is a pre trained model, the dataset used for this model is COCO dataset which is Common objects in context. So, all models under the TensorFlow object detection are trained with this dataset. COCO contains around 120000 pictures and it contains 88000 objects that are labeled in those pictures. Models contained in the TensorFlow Object detection model zoo are trained to detect over 90 different kinds of objects labeled in this COCO dataset. This list of objects like a car, a toothbrush, a banana, a person etc. Efficiency of the system depends on the speed of the model. I did some tests to find out the quality of the model depending on the predictive speed. Since we do not use real-time video and using a pre-recorded video file as an input. I have selected the model faster_rcnn_inception_v2_coco with 28 mAP (detector performance in the verification set), with a maximum output speed of 58 ms.

PEOPLE DETECTION

To find people, there are some steps that need to be taken:

- First, we define the outputs that we required to induce from the model. For that we should the pass model file to the graph of TensorFlow
- Extract each frame from the video file and pass that image through the TensorFlow graph so as to induce the outputs that are required.
- Filter and remove all the predictions that are week and also remove the objects that are not requires for our system.

LOAD AND START THE MODEL

TensorFlow models are designed based on the graphs. So we should load the model into a TensorFlow graph first. This graph will contain various functions that help to determine the specified detection. The next step is to create a session that is responsible for performing the tasks described within the previous graph.

PASS EVERY FRAME THROUGH THE MODEL

A new session for each framework that needs to be considered is initiated. This is done by requesting a run () function. Some restrictions should be clarified when doing so. This includes the type of input required by the model and what results we wish to bring back. We need to get the following results:

- Coordinates of Bounding box for every Prediction.
- The Confidence rate of all objects that are detected (0 to 1)
- Class of all objects that are predicted (0 to 90)

FILTER OUT WEAK PREDICTIONS AND NON-RELEVANT OBJECTS

One of the many categories identified by the model is human, the human category is 1 in the TensorFlow object detection Model.

To remove both weak predictions (limit: 0.75) and other non-human categories of the item, We can use IF statement to combine both of these conditions to exclude another item from continuous integration.

if int (classes [i]) == 1 and confidnce [i]> 0.75

But since TensorFlow have already been trained, it is unlikely that they will ever find the Person class alone. So, TensorFlow will take too much time to load in the system because they strive to see all 90 different types of objects on the scene.

BIRD EYE VIEW TRANSFORMATION

Modifications of bird viewing will provide a superior view. Happily, OpenCV has great built-in functions for applying this technique to an image to alter the image taken from the viewing area to the top view of this image.

The first step is to select 4 points in the first picture as the points in the corner of the plan to be changed. This point should form a rectangle with at least 2 opposing sides. If this is not done often, the magnitude will not be the same when the transformation takes place. We are using the OpenCV setMouseCallback () function to create these links. Image size is also required by the function of compiling the matrix of change to be calculated using the image material. width, height, _ = image.shape This returns the width, height and other pixel values of the corresponding color. For each recipient, two points needed to make a binding box are returned.

Points that are in the left top and the point at the right bottom corner is used. From this, we can calulate the centroid point among these points by calculating the middle of these points.

Using this result, I calculated the links for the purpose found in the center of the box below. This point of the world, well represents the connection of the people in the picture. After that I used the conversion matrix to calculate the modified links of all the points found at the bottom. This will be done for each installation frame, using cv2.perspectiveTransform ().

MEASURING SOCIAL DISTANCING

In each frame, a list containing all the new converted points is found from previous steps. In this list I had to measure the gap among each point combinations. Now I used the python library 'itertools' this library contains a function called 'combinations ()' that allows find all combinations within our list without having a double value. Then the distance is obtained by python using the maths.sqrt () function. The selected gap was 120 pixels, because it can measure 2 feet in our area. Once the 2 points that are closest to each other are identified, the color of the circle that marks the purpose is adjusted from green to red and is the same in the rectangle in the main frame.

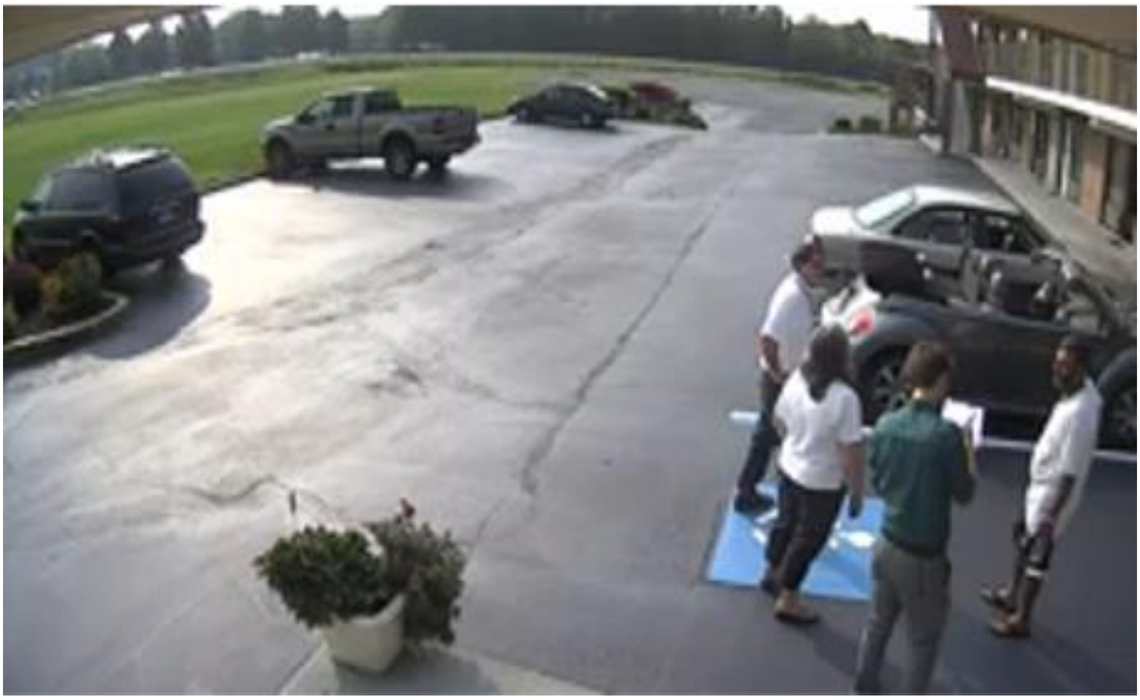
III. MODELING AND ANALYSIS

For this project we will be using TensorFlow object detection model. It is a pre trained model, the dataset used for this model is COCO dataset which is Common objects in context. So, all models under the TensorFlow object detection are trained with this dataset. COCO contains around 120000 pictures and it contains 88000 objects that are labeled in those pictures. Models contained in the TensorFlow Object detection model zoo are trained to detect over 90 different kinds of objects labeled in this COCO dataset. This list of objects like a car, a toothbrush, a banana, a person etc. Efficiency of the system depends on the speed of the model. I did some tests to find out the quality of the model depending on the predictive speed. Since we do not use real-time video and using a pre-recorded video file as an input. I have selected the model faster_rcnn_inception_v2_coco with 28 mAP (detector performance in the verification set), with a maximum output speed of 58.

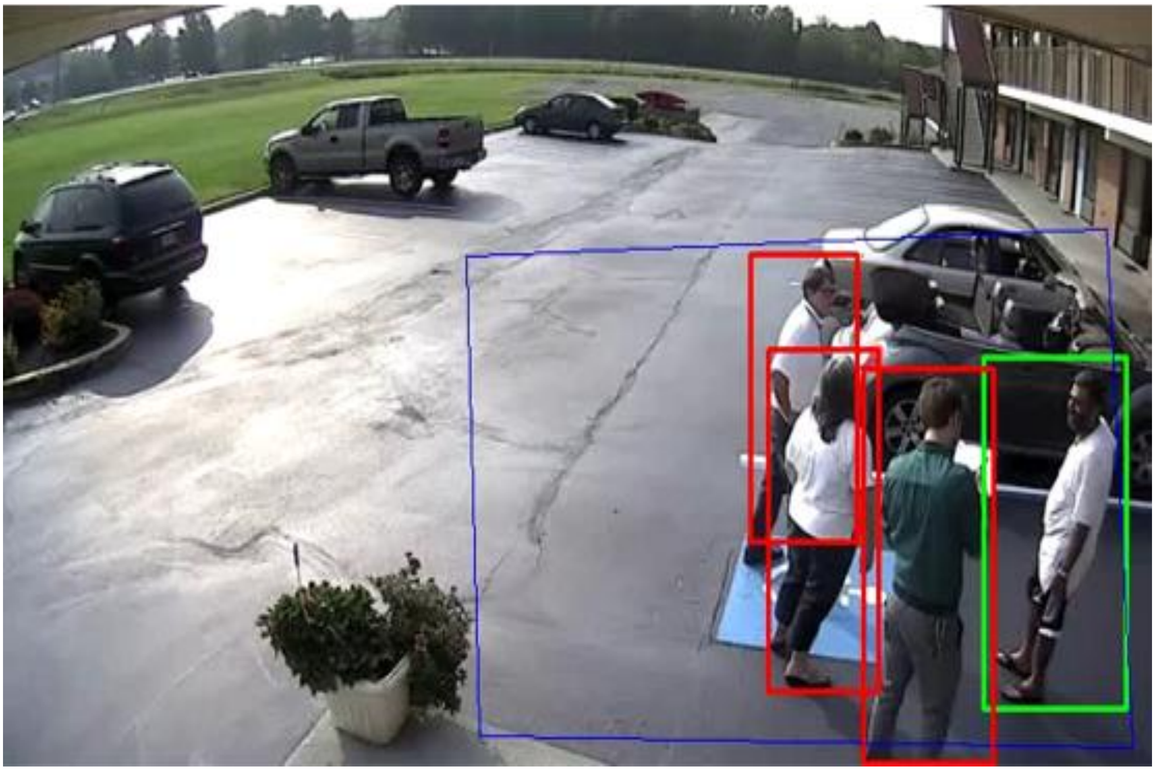
This Common objects in context dataset is can work with a large-scale object detection, captioning, and segmentation. The latest version of this Common objects in context dataset consists of images, bounding boxes, and their labels.

IV. RESULTS AND DISCUSSION

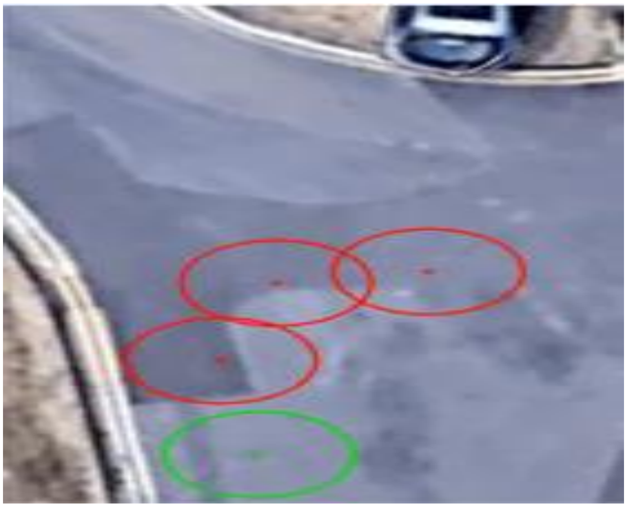
INPUT:



OUTPUT



BIRD EYE VIEW TRANSFORMATION



Results which are obtained by passing a video to the model as a part of this study is showcased. The Green and Red boxes are used to show if the social distance is followed or not. The result is correct based on the distance set to check the social distancing.

V. CONCLUSION

We have used TensorFlow and OpenCV to create a system will help in maintain an environment that is safe for the protection of public by automatically accessing when COVID protocol is maintained. This system will work best in this situation when lockdown is almost over and helps to monitor all public areas easily in an automated manner. The implementation of this program has been successfully tested and this social distance detector can reduce the violation of the COVID protocols and will help in controlling the new spread of the disease. This system can be used in various public places where monitoring of the social distancing becomes a very important need like Temples, Offices, Colleges, Railway Stations, etc.

VI. REFERENCES

[1] Vishnu Singh Sengar, Snigdha Das," APPLYING AI IN TIMES OF COVID-19".

[2] Savyasachi Gupta∗ , Rudraksh Kapil† , Goutham Kanahasabai‡ , Shreyas Srinivas Joshi§ , Aniruddha Srinivas Joshi, "SD-Measure: A Social Distancing Detector".

[3] Moein Razavi, Hamed Alikhani, Vahid Janfaza, Benyamin Sadeghi , Ehsan Alikhan, "An Automatic System to Monitor the Physical Distance and Face Mask Wearing of Construction Workers in COVID-19 Pandemic".

[4] Dhanush Reddy K N1 , Darshan B2 , Ganagalla Jeevan Kumar3 , Gopi Krishna R4 , Nitya S, "Social Distance Monitoring And Face Mask Detection System For Covid-19 Pandemic".

[5] S A Sanchez 1 , H J Romero 1 y A D Morales," A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework"

[6] Priya Pandiyan, "Social Distance Monitoring and Face Mask Detection Using Deep Neural Network"

[7] Shashi Yadav, "Deep Learning based Safe Social Distancing and Face Mask Detection in Public Areas for COVID19 Safety Guidelines Adherence"

[8] Aatiya Chaudhary," Real Time Social Distancing Violation Detection using SVM classifier and Deep Neural Network learning techniques".