TOOL FOR SOCIAL DISTANCING USING AI

Krishnakumar.S
Department of Electronics and Instrumentation Engineering, Bannari Amman Institute of Technology, Sathyamangalam, India.

Mohanaprasad. K
Department of Electronics and Instrumentation Engineering, Bannari Amman Institute of Technology, Sathyamangalam, India.

Monisha. K
Department of Electronics and Instrumentation Engineering, Bannari Amman Institute of Technology, Sathyamangalam, India.

Ramharnish. J
Department of Electronics and Instrumentation Engineering, Bannari Amman Institute of Technology, Sathyamangalam, India.

***Abstract*—In the war against the pandemic disease COVID -19, social distance forbid the risk of transmission of the disease. There are many guidelines announced by WHO to survive in this critical situation, among those guidelines the social distance reduced the peak of transmission. The WHO states that everyone should maintain at least 1 meter (3 feet) distance from another person. The reason behind this statement is, the droplets from disease affected person may contain virus which can easily transmitted to other persons. So the only better way to prevent our self from this pandemic disease is to maintain the social distance. So in this proposed system we designed a tool which will deduct the real time distance by using deep learning and some computer vision skills. The project runs on Google colab platform which has intelligent python notebook (IPNB) in it. We have included several open source libraries and packages and obtained certain functions out of them to run the system. This system will assist to monitor the people in a public places and to give alert to protect them from the pandemic disease.**

Keywords— Social distance detector, YOLO, Object detection, openCV, Google colab.

I. INTRODUCTION

The COVID-19 has taken the world by storm and it has transformed the way of living in every aspect. Several medical specialist, healthcare organization, researchers and technologist are working hard to produce vaccine but till now there is no proper vaccine is produced. Social distancing has become one the most important factor today and it is the best preventive measure to overcome this deadly disease. The biggest cause for the spread of the virus is due to person to person contact, without practicing social distancing. There are a lot of places where people don't follow social distancing and it has been very difficult to monitor proper social distancing. This project has a solution for the above said difficulty. This social distancing tool monitors and shows who are all following social distancing and who are not following. Through image processing the live video is taken, by using YOLO algorithm objects are detected and the distance is calculated by using Euclidean formula. We should keep at least 6feet distance from each other. A particular social distancing distance I is pre- set and if people are close to each other (maintain less distancing) they will be indicated by a red box surrounding them. If there is correct social distancing between people they will be indicated by a blue box surrounding them. By this way, in public places it is easy to monitor social distance among the people. This can be used by governments to analyze the movement of people and alert them if the situation turns serious.

II. LITERATURE SURVEY

In 2020 July, Dongfang Yang, et. al. [1] proposed "*A Vision-based Social Distancing and Critical Density Detection System for COVID-19*". There were two motives of the system; they were normal social distancing detection in less populated areas and Critical social density estimation in highly populated areas. Real time monitoring and not recording data is the main objective of the system. An audio alert occurs when a person violates social distancing. It uses MS COCO datasets for object detection. A fixed monocular camera is used here for detecting humans using area of interest (ROI) and to find the distance between individuals. A scatter plot between Social density (people/m2) and number of social distance violation is drawn and the detection is done.

In 2020 July, Marco Cristani, et. al. [2] proposed "*The Visual Social Distancing Problem*". The paper mainly focuses on how and what kind of crowd form during different occasions. It introduces Virtual Social Distancing (VSD) which the monitoring through cameras. It's mainly tells that not all violations are intended and some have their own excuses. For example, a mother with her new born baby, an

elder with a caretaker etc. In this scene geometry is used for ground plane detection. A geometrical disc of radius 1 meter is drawn around individuals and the circle is considered as safe area.

In 2020 August, Afiq Harith Ahamad, Et. Al. [3] Proposed "Person Detection For Social Distancing And Safety Violation Alert Based On Segmented Roi". This Paper Involves Mobilenet Single Shot Multibox Detector (Ssd) For Detecting Human Objects. The Distance Between Individual People Is Calculated Using The Pixel Values That The Individual Is Covering. Another Key Feature Of Detecting Individuals In Restricted Areas Is Also Added In This System. Accuracy Levels For Object Detection Have Been Done In Less Crowded Places And Also In Highly Crowded Places And It Has Given Good Accuracy Levels.

In 2020 August, Vinitha, et. al. [4] proposed "Social Distancing Detection System with Artificial Intelligence Using Computer Vision and Deep Learning". This paper deals with using Birds eye (top down) view of the image/frame for object detection. As the input is monocular (from a single camera lens). The whole frame is turned into a bird's eye view (image seen by a bird flying above).It is because top view or bird eye view has the property that points are distributed uniformly vertically and horizontally.

In 2020 December, Priya Pandiyan, et. al. [5] proposed "Social Distance Monitoring and Face Mask Detection Using Deep Neural Network". There were two motives discussed in the paper. One is face mask detection and the other is social distancing detector system. First face mask detection is done using raspberry pi. Face identification is done through open CV and the image is stored in Amazon AWS. Then the image is passed to CNN for face mask detection. If the person is found not wearing a mask then an alert by SMS is given to the monitoring team. Further a social distancing detector tool using Open CV is also done.

In 2021 January, Sergio Saponara, etc. al. [6] proposed "*Implementing real-time, AI-based, people detection and social distancing measuring system for Covid-1*". In this system a thermal image/video based object detection technique is used for faster and accurate object detection. The input is fed through a thermal camera and the image is then fed through the deep learning object detector. Thenumber of people is counted and centeroid is assigned to each individual. Distance between each centeroid is calculated and social distancing is monitored.

In 2021 February, Lakshmisri Surya, et. al. [7] proposed "AI Economical Smart Device to Identify Covid-19 Pandemic, and Alert on Social Distancing WHO Measures". This is a physical tool for monitoring social distancing which can be worn by individuals. PIR sensor is the core of the tools and it has a circuit comprising diode, resistors, transistors, buzzer and a battery. The tools actively can be worn like a wristband and it actively calculated social distancing distances between the individual and others. For people detection the author has proposed a thermal image based detection system.

In 2021 February, Adina Rahim, et. al. [8] presented "*Monitoring social distancing under various low light conditions with deep learning and a single motionless time of flight camera"*. The paper involves using You Only Look Once v4(YOLO v4) model for real time human object detection and it uses single motionless time of flight (ToF) camera. The ToF camera gives a real time distance image which simplifies the distance calculation between individuals in a photo frame. Tof camera uses light reflection from objects to calculate the distance.

The discussions including different methodology of the above papers provided many different ways in which the project can be taken forward. There are other features introduced in the system that acts as a solution for the problem.

III. PROBLEM STATEMENT

Some problems faced in social distancing detector tool are

- Covid virus spread is mainly because of not following social distancing in public places. There is no robust and fast monitoring tool.
- The speed for object detection (Human detection) is very slow as it needs to pass through huge datasets.
- No proper alert system for social distancing violations is being done.
- Computation speed is very less for performing Convolutional Neural Networks (CNN) for object detection.
- Calculating distance between individuals in camera frame/picture is difficult as it is monocular.
- Monitoring highly dense public places is difficult as there will be lot of objects within a frame.

IV. OBJECTIVES

- The primary objective is to design a fast object detection system for human detection.

- The system runs on Google Colab and Google Colab offers free Graphical Processing Unit (GPU) for very high computational speed.
- Google Colab also offers free Tensor Processing Unit (TPU) for performing huge machine learning tasks.
- Distance between individuals is done using Euclidean Algorithm and it is very useful even in highly populated regions.
- Number of social distancing violations is displayed in each frame.
- Every frame is put together as video and is saved in our desired folder.

V. SOFTWARE DESCRIPTION

Google Colab is an online based coding platform through which we can train machine learning models, analyze and visualize large datasets. The Google Colab platform is extremely powerful with which we can build complex ML models, work with large dataset, and even share it with others. The platform allows us to integrate executable code and text in a single document and also with images, HTML and so on. All the Colab notebooks that you create in stored in your Google Drive account. It has lot of features compared to the conventional Jupyter notebook. The Google colab platform is shown in figure 1.

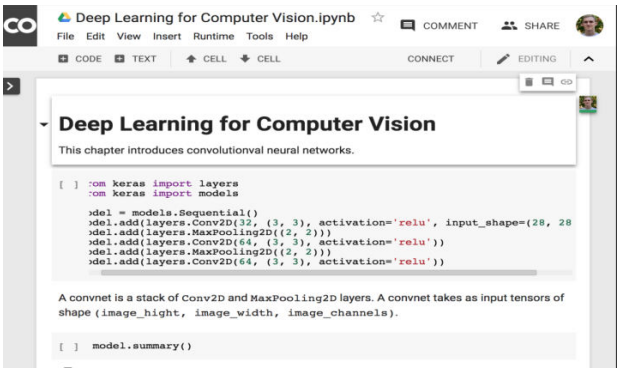

Figure-1: Google Colab Platform

VI. LIBRARIES USED

For training Deep Learning models we have to import and use lot of libraries. Libraries are used to perform various tasks in the project like plotting, generating numbers, performing numerical operation etc. Some of the common libraries used in the proposed project are **NumPy, cv2, random and matplot.lib**.

Numpy

NumPy stands for numerical python and it is a library used for performing mathematical operations in multi-dimensional arrays. By using NumPy we can perform arithmetic operation, statistical operation, Bitwise operator, Stacking, Matrix's operation, Linear algebra, copying and viewing array and broadcasting. Instead of using list we are using NumPy because python list requires 4MB memory whereas NumPy requires 2MB memory. Then its execution speed is high and it is well timed to work.

Matplot

Matplot library is used for data visualizing in python in a form of line plot, histogram, bar graph, pie chart and so on. One of the good sake of visualization is permitting us to access an N number of data in effortless manner. The different types of plots in matplot.lib are shown in Figure 2.

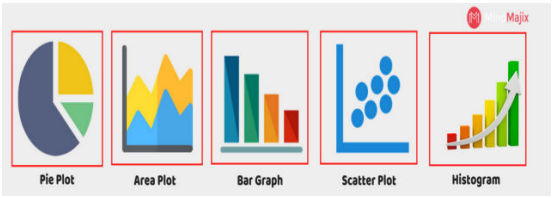

Figure-2: Types Of Plots

OpenCV

OpenCV is an open source library and CV stands for computer vision, which is preferred to code real-time computer vision applications. This library is used for image and video analysis. The processes that we can do in openCV library are face detection, License plate reading, Character recognition, Gesture recognition, Image restoration, Scene reconstruction, motion analysis.

Random

This random library is used to generate random numbers. The function which are used in to generate numbers are sample(), choice(),shuffle(),seed(), uniform(),random(), randint(), randrange().

VII. SEGMENTS OF SOCIAL DISTANCE DETECTOR

1.OBJECT DETECTION USING YOLO

Object detection is one of the major subset areas of computer vision or image processing. It is a computerized method for detecting/locating objects in a given frame or picture. Today object detection has become one of the hot

topics as its need is growing day by day. With the boom in Machine Learning techniques happening across the world, object detection is becoming a major topic and new technologies enhancing object detection is released every now and then. An object for object detection is shown in figure3.

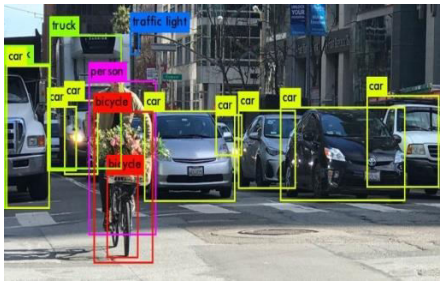

Figure-3: Object Detection

There are a lot of methods in object detection. Some of the techniques are R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN, YOLO, SSD, RetinaNet and much more. Each technique has its own unique characteristics. The above said techniques are differentiated mainly by their object detection method and speed of object detection. All the CNN based detection methods are traditional methods and have generally good accuracy but they are very slow in detecting objects. The detection method which is implemented in this project is (You Only Look Once) YOLO. YOLO is one of the newly introduced method for object detection which is fast and has good accuracy.

YOLO Object Detection

YOLO is an open source neural network framework. YOLO is a widely used single stage object detection oriented technique used for real time object detection. It processes real time input with very less delay having a good accuracy. As the name suggests, it only need time processing to locate and find all the objects within a frame/image. YOLO is being implemented by using K-means Clustering on the training dataset for object detection. The YOLO framework is incredibly fast as it can process 45 frames per second. The overall network architecture of YOLO is shown in Figure 4.

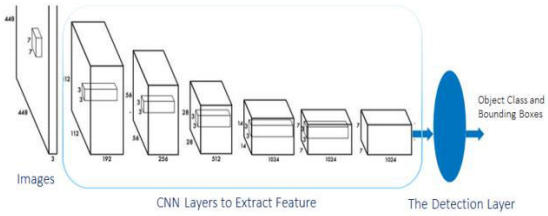

Figure-4: Yolo Network Architecture

The process by which YOLO detects objects in a given frame is discussed below. YOLO takes the input as an image or frame. The 1st process is to split the image into grids of equal size. The 2nd process after splitting the image into equal sized grids is to define or predict a number of bounding boxes in each grid with already drawn predefined shapes which centers the grid center. Each of these predictions is linked to a class probability and class confidence. It may contain either object class (human, car etc.) or just plain background of the image.In the final process image classification and regression is applied to each grid and the bounding boxes with high class probability of objects is finally selected. Now after this only the objects in the image are detected and the rest bounding boxes containing only background and no object is deleted. A visual representation block diagram of YOLO object detection shown in figure 5.

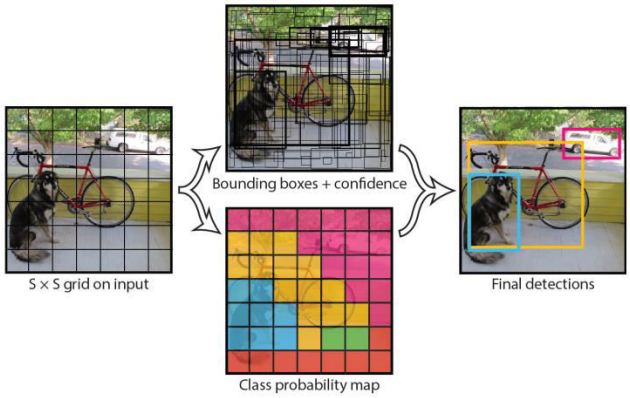

Figure-5: Visual Representation Of Object Detection

2.OBJECT TRACKING USING OPENCV

The next segment after object tracking object detection is object tracking. Object tracking is tracking detected objects and new objects in the input. It usually takes the bounding box coordinates for tracking objects. After taking the bounding box coordinates each unique ID is created for each object detections. After assigning ID's to objects detections object tracking maintains tracking of the detected objects as they move in consecutive frames of the input video maintaining the same unique ID. If new object is detected any of the frames then a new unique ID is assigned to the object. As we have unique ID's assigned we can easily get the count of individual objects in every moving frame (human objects in our case.

Figure-6: Example Of Object Tracking

Figure 6 is an example of object tracking in a traffic signal. We can notice lot of pedestrians crossing the road and some cars in each frame. As we can see all the human objects are tracked frame by frame as they cross the road. The bounding boxed does not change and the unique ID is maintained for every individual until they go out of the frame boundary.

A good object tracking algorithm has the following features,

- Needs only the object detection process done.
- Object tracking should happen faster than the object detection process.
- It should be able to track new objects introduced into the frame and also should be able to handle objects that goes away from the frame boundary.
- Be very fast and should not confuse between objects crossing or overlapping (occlusion).
- It should be able to track any lost objects which was not tracked in earlier frames.

The working of object tracking algorithm is discussed below. Object tracking is also known as centroid tracking since it depends on the Euclidean distance between existing objects centroids and newly detected objects centroids. The object tracking or centroid tracking algorithm has several steps. The steps are discussed below.

Step 1: Computing Centroids for Detected Bounding Boxes (Objects). A visual representation of step 1 is shown in figure 7

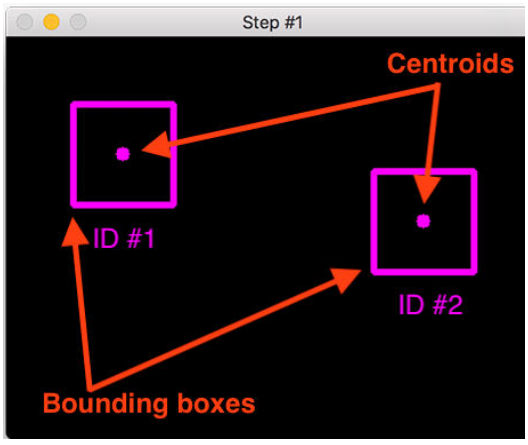

Figure-7: Assigning Bounding Boxes

The centroid algorithm accepts bounding box coordinates (x, y) for every single object detected in the frame. After getting the bounding box coordinates we have to compute the centroid for those boxed or simply we have to compute center for those bounding box coordinates. After computing centroids for each bounding boxes a unique ID is assigned to each bounding box detected in the frame.

Step 2: Calculating Euclidean distance between new objects and old bounding boxes. In this step we track existing objects within the frame and detect new objects and assign unique ID to newly entered object. A visual representation of step 2 is shown in figure 8. In the figure 8 three yellow dots represent present position of objects and 2 pink dots represent the position of the objects in the earlier frame. To not confuse between earlier position, present position and newly detected object position Euclidean distance is calculated between earlier and present points. After calculating the distance we know have to decide which pair is same object and which new object is.

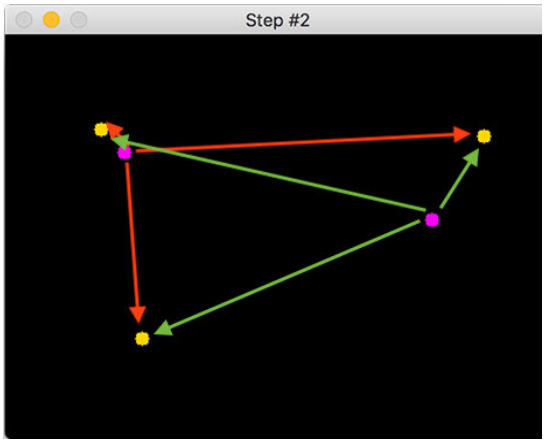

Figure-8: Representation Of Step 2

Step 3: Updating of coordinate points for existing objects and assigning ID for new objects. After calculation Euclidean distance between new and old points**,** pairs which are close to each other are concluded as objects that have moved. So the ID's of the pairs which has less Euclidean distance will be updated (ID will be carried over to yellow points which are above). Now only the yellow point (Object) is left alone and the ID assigning for this will be solved in step 4.

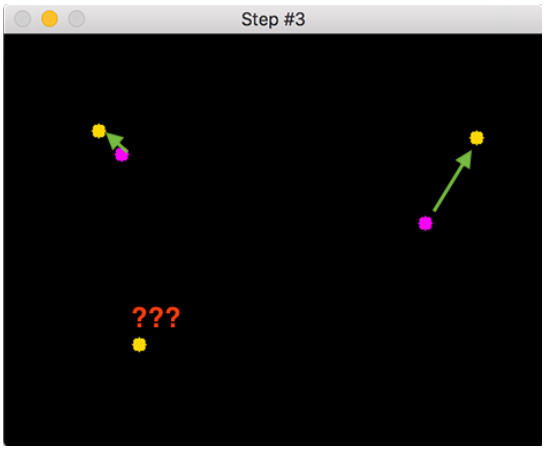

Figure-9: Representation Of Step 3

Step 4: Assigning ID's to track existing objects and new objects. Now we are left with 3 objects in present position (Some frames moved). The IDs of objects which were represented in purple will be assigned to their respective bounding box (new position of the purple objects). The yellow object which is left alone below will be treated as newly entered object and a new ID will be assigned too is. A visual representation of step 4 is shown in figure 10.

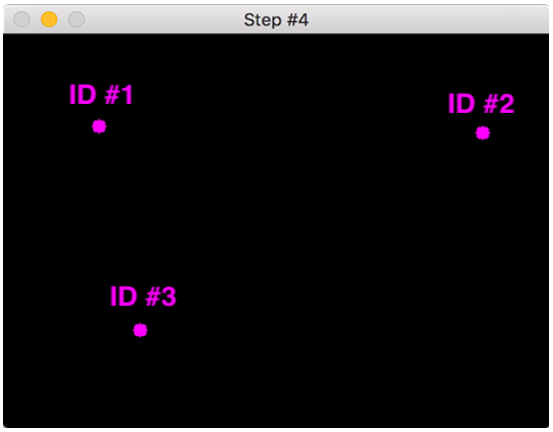

Figure-10: Representation Of Step 4

Step 5: Deregistering old objects. In this step the algorithm tries to find any lost objects and tries to regain its ID. The ID's of objects which have left the frame boundary will be deleted. Step 1 to step 6 shows the implementation and working of object tracking.

3. DISTANCE CALCULATION BETWEEN OBJECTS IN FRAME PLANE.

Since we are processing the data in a video there comes a question for calculation distance between objects that are in the video. Also naked eye vision is completely different from camera lens vision. The last and final segment of social distance detector is converting camera lens vision distance into real world distance. Our main objective is to measure the distance between the camera and the object. Then we have to get the pixel wise area occupancy of objects in the frame for calculation of distances in frame level. Figure 11 shows the principle of distance calculation.

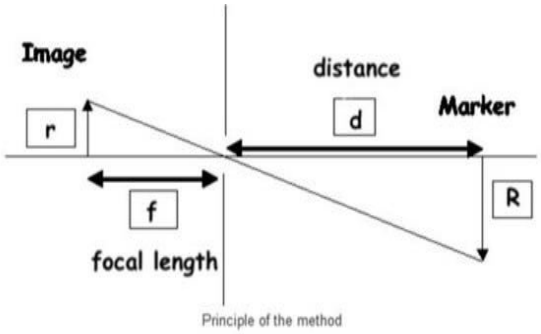

Figure-11: Principal of Distance Calculation.

Using similar triangles we can obtain the formula as,

f/d = r/R ---------- (1)

f = d*r/R pixels ------ (2)

d = f*R/r cm --------- (3)

The parameters are

- f – Focal length.
- r – Radius of the marker (object) in the frame plane.
- R – Radius of the marker (object) in object plane.
- d – distance between camera and the object

With all these parameters known we use the principle of Similar Triangles and get the formula for distance d. After applying this formula we know will be able to convert the object plane distance into frame plane distance. Now it will be much easier to calculate distance between objects within the frame. For measuring distance between objects in the frame Euclidean distance formula is used.

EUCLIDEAN DISTANCE

Measuring distance between two attributes or objects is an important task in deep learning. The usual algorithms used in deep learning are K-nearest neighbors, K-means clustering and so on. They play vital role in relative distance between two objects or attributes in a domain. In this project the Euclidean distance algorithm is implemented to find distance between two objects. In this case the humans are the mentioned objects. The Euclidean distance calculates the distance between two real valued vectors. The most likely reason to use this particular algorithm is to calculate distance between two rows of data that have numerical values like floating value or integer values.

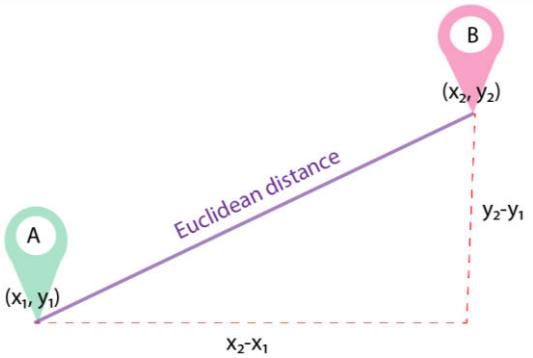

Figure-12: Euclidean Distance Visualization.

By using Pythagoras theorem distance formula is derived,

Euclidean Distance = √[((x2-x1)^2) - ((y2-y1)^2))]

Now for n dimensional place

$$\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Where,
n is no of dimensions.
xi , yi are the data points of individual objects.

VIII. BLOCK DIAGRAM OF PROPOSED PROJECT

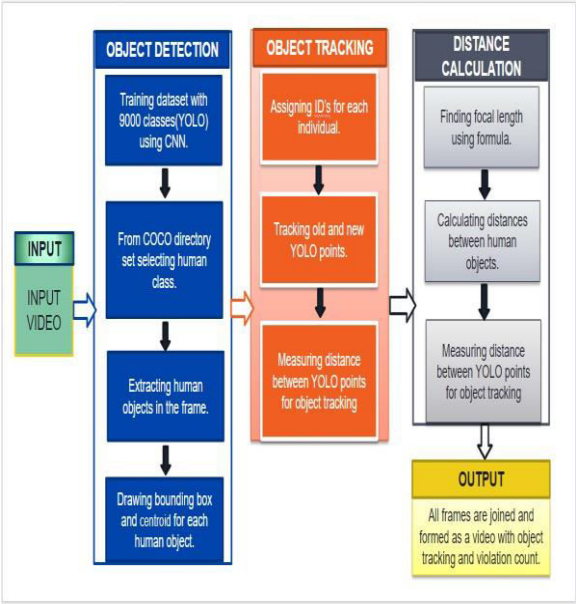

Figure-13: Block Diagram of Proposed Project

From the above block diagram we infer that, the process of social distance detection takes place in 3 consecutive steps. Once the video is given as input and program is run, the video is split into individual frames. Each frame is taken as jpeg image format. The image frame is run through the object detection phase using the convolution neural network where it identifies humans from the COCO directory. The coco directory consists of various classes like humans, cats, dogand other such animals and objects. When a frame is run through image processing, individual pixels are identified. Then the pixels group that varies from one another are identified which results in the identification of individual objects. Then the objects are run through the coco directory, where different datasets and pre trained models are already available. The gathered pixel group and available datasets are run through a matching program. When it is run through the human objects are identified and their pixels are marked, if the gathered pixel group doesn't match the human dataset, it proves that the object is not human but something else.

Once the humans are identified from the frame, bounding boxes are drawn with the help of blob from image method which is a prebuilt library ofopenCV library, which is an image processing library. Once the bounding boxes are drawn for the identified human object, the next process is to assign the individual centroid to the object. From every corner of the bounding box the centroid position is obtained and the circle is drawn. It is to be noted that the pixel position of the centroid is stored in the centroids variable which is already created earlier. Once the object is identified, the frame enters the second phase of object tracking.

With the help of yolo points and individual ids the object is tracked from frame to frame. The frame to frame tracking helps with the identification of the count of violations made. In the distance calculation phase, the yolo points andindividual ids play a major role. It is used to separate one object from another and also helps in calculating the closest distance from each other. When the distance calculated is less than the previously set threshold value it is identified as violated objects. The bounding boxes of the violated object turns from green to red. Thenumber of violations per frame is mentioned at the left corner of the frame for easy user interface. All these processes are taking place for one single frame. These processes are repeated sequentially for every frame of the input video. These output frames are printed in the output screen.

As the final step the individually stripped frames are put in a for loop to be merged to convert it into a video. This output video is then stored to the mounted drive. This pretext is observed from the block diagram as shown figure 13.

IX.PROPOSED METHODOLOGY

The first part is the processing part. The processing segment of the social distancing project is divided into three parts, same as which the code is also designed. The three parts of the processing are,

1. Initialization.
2. Creating function
3. Prediction system.

1. INITIALIZATION

The initialization consists of – lines of code, which deals with the declaration of minimum, object detection confidence variable and non-maxima suppression threshold variable. The minimum distance variable is declared and 50 is given as input value. The minimum distance variable indicates the minimum distance that should be maintained between two human objects in terms of pixels in the frame image. Hence in the initialization section the user defined variables are declared.

2.CREATING FUNCTION

The function creation section of the code implies the methods that are applied for object detection. There are several types of object detection like single shot detectors, faster R-CNN etc. but in this project we are using YOLO object detection with openCV which generally requires a bit more output processing. The detect people function is called in the code, which generally encapsulates any object detection logic. The next step of the code consists of the calling of user declared variables from the initialization section, like the minimum threshold and minimum confidence. This is achieved using the help of from function of python which calls variables and objects from various sections of the code to one and other.

The NumPy and openCV is also imported in the script. The program consists of a single function for detecting the people object. The detect people function accepts four parameters such as the frame of the video file inputted, the net which has pre trained YOLO object detection model, the ln which has the output layer names of the YOLO CNN, and the person ID which specifically identifies the human object and all other objects are neglected. The frame object is used for scaling purposes, and result variable is a list function which ultimately returns results such as the prediction probability, bounding box coordinates for detection and the centroid of the object in the end when called upon. The layer outputs and detections function are put in a loop in order to extract every frame of the video and process it through object detection, and every time the class ID and confidence of the current detected frame is passed to the results function. In order to perform object detection in our frame, the pre-processing is a blob to be constructed.

Then we compute the bounding box coordinates and find the center of the bounding box. With help of bounding box coordinates we find the top-left coordinates for the object. The initialized list will soon hold our object detection confidences, bounding boxes and object centroids. Then the program loops over each layer output and detections out of which we first extract class ID and confidence of the current detected object. From there we verify that the identified object is human alone and the minimum confidence value is met or exceeded. Then the non-maxima suppression is used to suppress weak, overlay bounding boxes. Assuming the result of NMS yields at least one detection, it is looped over them to extract bounding box coordinates and our result list is updated. The result function is returned to the calling function.

3.PREDICTION SYSTEM

The prediction system section of the program begins with importing libraries like numpy, argparse, imutils, cv2 and os. Most important functions like cv2_imshow from colab patches and distance function from scipy.spatial are imported from their respective libraries. The script is then added with various arguments to be passed via the terminal. The arguments are the input video file location or live stream address, the location of the output file where it needs to be stored and the display parameter where each processed frame is displayed on the output screen.

DEFINING YOLO POINTS

The next process is to load COCO labels and we define our YOLO paths. Only with the help of YOLO paths we can load the model into the memory. Through openCV library's DNN module the YOLO net module is fed into the memory. Here an optional setting is made, if the user has inbuilt GPU they can activate it here in the config. Once the output layer names from YOLO is gathered they are kept in order to process our result. When the segment enter the loop function the frames are dissected from the video and if the dimensions are largewe can resize each frame while maintain aspect ratio. Then the detect_people function implemented in the previous section is called and so we grab results of YOLO object detection.

The violate function is implemented to maintain the list of people who violate the social distancing regulations. Then the program proceeds to compute Euclidean distance

between all pairs of centroids, and loop over upper triangular of distance matrix. Then if function is implemented to see if anyone violates mentioned safety distance. Now we enter the visualization segment. When human objects are identified their bounding boxes and centroid coordinates are extracted and initially green colour is given by giving appropriate rgb value. When violated object is identified that bounding box is turned to red colour by changing its rgb value. Violate set is used to display the information about number of violations on the frame image.

OUTPUT EXTRACTION

The final step of the program with all the important processes being completed is to grab all the frames and convert it into an output video. For that the writer function is declared. The processed fame is annotated to the disk. Then each processed frame with the human objects being detected, distance calculated, with identified violated objects and count printed frame is put in through a loop. The loop verifies it as processed frame and then writes it one by another. The write function merges all the frames to make it as a video. Once the video is processed and completed it is saved to the allotted memory drive in this case the Google drive in mp4 format. The name of the output file is already declared as output.mp4. This video file can be found in the output folder defined in the code.

X.RESULT AND DISCUSSION

In this project, artificial intelligence based social distance detector tool is presented in an intelligent python notebook framework with various concepts of deep learning being used. Highly accurate human detection is achieved through YOLOv3 paradigm. The detection model uses YOLO object detection for detecting human objects out of the coco directory, creates id's for object identification, creates centroids and bounding boxes for object tracking, uses Euclidean distance and proximity distance to calculate distance between various objects and uses python writer function with other such python libraries installed in various part of the code to deliver the output video. The result video output's screenshot is shown in figure 14.

Figure-14: Screenshot of Result.

The pre trained transfer learning model is trained in a

manner to adopt to various appearance, scale and size, varied poses of a person and visibility from a significant point of view. The model has an overhead dataset which is used for the initial process and when used the newly trained model is also appended to existing dataset. When we trained the model without transfer learning the model delivered an accuracy of 92% and when trained with transfer learning model it delivered 95% accuracy. Comparing the model we delivered here with YOLOv3 with other RCNN models it has 95% accuracy calculated from the true positive, true negative, false positive and false negative data as shown in table 1.

S.no	Model	True Detection Rate	False Detection Rate
1	Fast-RCNN(pre-trained)	90%	0.7%
2	Faster-RCNN(pre-trained)	92%	0.6%
3	Mask-RCNN(pre-trained)	92%	0.5%
4	YOLOv3(pre-trained)	92%	0.4%
5	**YOLOv3(trained overhead dataset)**	**95%**	**0.3%**

Table -1: Performance Table Of Proposed Model.

XI.CONCLUSION

The corona virus (covid-19) is a highly contagious disease that attacks the breathing functions of human body. The disease originated in china in late 2019. The disease then spread like a wildfire spreading across the entire globe with the death toll of 3.25 million till may of 2020. The main cause for the spread is lack of social distancing which causes the disease spontaneously in the first place. Governments worldwide imposed strict curfews and rule of maintaining a 6 feet distance from one and another. But in countries like India monitoring social distancing is hard hence the social distance detector tool proves to be a highly innovative tool to monitor high human mass. If the tool we designed is implemented in every city in every state it will be easy for the officials to monitor people at all times. One important advantage is that it does not need any extra hardware to work, it works with a simple computer and surveillance system with the right AI tool. This project will help us save millions of lives worldwide.

XII.REFERENCE

[1] Vanitha V, valentine V "Social distancing detection system with artificial intelligence using computer vision and deep learning" in IRJET volume number 8, issue 8 on August 2020.

[2] Narinder Singh Punn, Sanjay Kumar Sonbhadra, Sonali Agarwal and Gaurav Rai "Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deep sort techniques" in arXiv volume 7, on 27 April,2005.

[3] Mahdi Rezaei , and Mohsen Azarmi " Social Distancing Monitoring and Infection Risk Assessment in COVID-19 Pandemic" in appli sci volume 7 on 26 October 2020.

[4] Dongfang Yang, Ekim Yurtsever, Vishnu Renganathan "A Vision-based Social Distancing and Critical Density Detection System for COVID-19" in arXiv on 8July 2020.

[5] Priya pandiyan, "Social Distance Monitoring and Face Mask Detection Using Deep Neural Network" in IEEE volume 7 issue 8 on December 2020.

[6] Krish chaudray "Maintaining Social Distancing Using Artificial Intelligence" university of science and technology TJMCS Vol .5 No.3, pp. 197-204, 2020.

[7] Sergio saponara, Abdussalam, Allesio gagliardi "Implementing a real-time, AI- based, people detection and social distancing measuring system for Covid-19" in journal of real time image processing Vol. 11, Issue. 5, pp. 65-72 Sep-2020.

[8] Zhao, Z.-Q., Zheng, P., Xu, S.-T., Wu, "X Object detection with deep learning: a review" IEEE Trans. Neural Networks Learn. Syst. 30(11), 3212–3232 (2019)