FACE MASK, ROAD TRAFFIC AND LICENSE PLATE DETECTION USING YOLOv3

Shreya Garg
Maharaja Agrasen Institute of Technology
PSP Area, Sector-22, Rohini, Delhi, 110086
shreya00garg@gmail.com

Namita Goyal
Maharaja Agrasen Institute of Technology
PSP Area, Sector-22, Rohini, Delhi, 110086
namitagoyal@mait.ac.in

ABSTRACT

The growing cases of Covid-19 are a cause of concern all around the world. The major reason for this pandemic is the carelessness of people. Thus, it is important to look for solutions that help people in the long run. Wearing a mask is the most efficacious way to solve this problem. Roads being the most common way of transportation imply need of mask, road traffic and license plate detection systems and are something that are growing in demand. The detection system would be able scan people's faces and detect whether a person is wearing a mask correctly or not, detect current vehicles on road with their respective license plates. The system uses techniques that involve deep learning and a YOLOv3 detection model. Deep learning involves the use of neural networks while YOLOv3 involves detecting objects for a given image. This work involves inputting an image, video or live video which will lead to detection of mask on a person's face, detection of type of road traffic and detection of license plate of the vehicle by drawing a rectangular box. The project will ultimately make transportation management easier.

Keywords: Covid-19, Face Mask Detection, Road Traffic Detection, License Plate Detection, YOLOv3, Darknet, OpenCV

INTRODUCTION

The rapid increase of the Corona Virus has changed the life of people all over the world. The unexpected striking of the Covid-19 Pandemic has affected how people live their day-to-day lives. The affect was so great that it lead to over-filled hospitals, non-availability of oxygen cylinders, inaccessible blood donors to death in huge numbers. About 96.1 million cases were confirmed along with 2.06 million deaths globally as of January 20, 2021 [2]. The alarming rates of death were a cause of concern for people worldwide. People turned towards the most secure preventive measures which included wearing masks on a daily basis. Face masks helps the virus from spreading and reduce one's chances to catch it. However, it is important to note that

according to the World Health Organization (WHO), the right way to wear a mask is by adjusting the mask to cover the mouth, nose, and chin [3]. Thus, it necessary that a mask is worn properly by everyone as it acts a beneficial preventive measure.

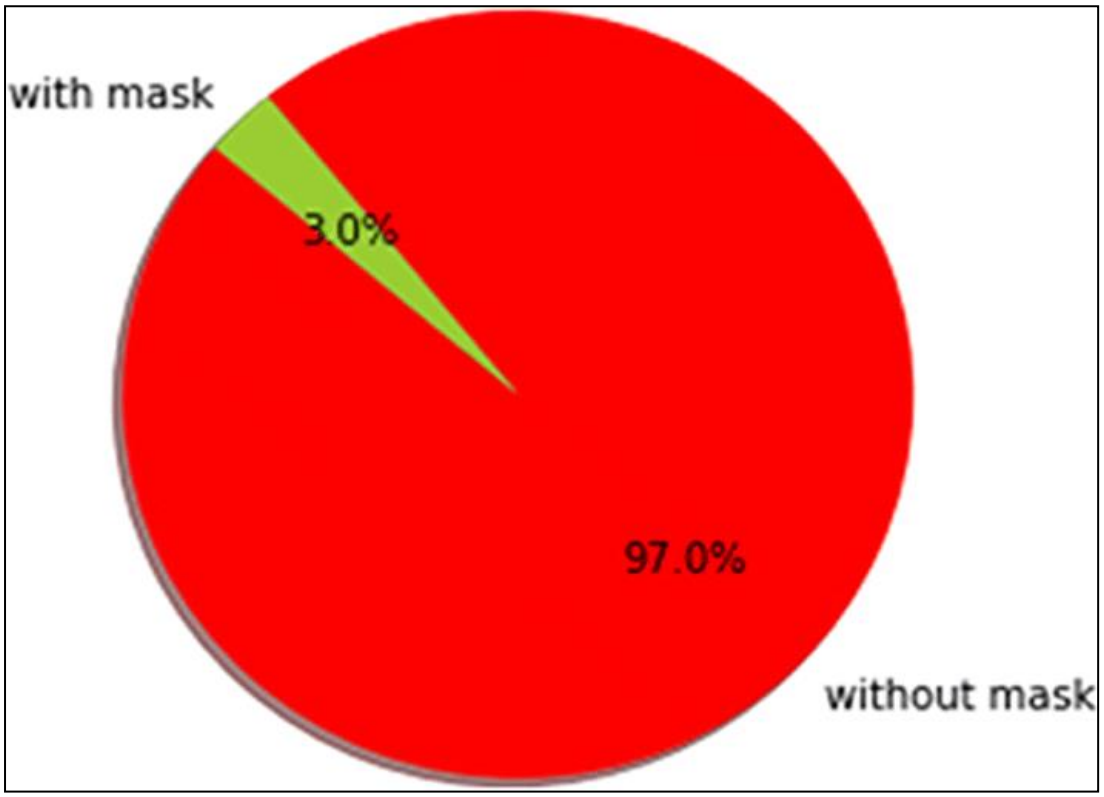

Fig 1. People not wearing mask has very high risk (97.0%) of getting infected [2].

Roads are the most common way of transportation for a common man. Most people use cars, buses, bicycles, footpaths and so on to commute from one place to another. Hence, it becomes necessary to monitor this road traffic and the people along with it to check whether people are following Covid-19 norms properly or not.

In today's era, humans and computers go hand-in-hand. Due to the Covid-19 pandemic, wearing masks became of importance, leading to the making of a mask detection system. Object detection and recognition helps many areas including live-video feeds to record and identify objects. Object detection can be used in fields including traffic-control, crime-related, mask detections, image annotations, face recognitions and many more.

YOLO is a Convolutional Neural Network (CNN) for performing object detection in real-time that identifies specific objects in videos, live feeds, or images [11]. CNNs are classifier-based systems that can process input images as structured arrays of data and identify patterns between them [11]. YOLO has the advantage of being much faster than other networks and still maintains accuracy [11].

In this work our objective is aimed at making a mask detection system using YOLOv3. The mask detection system aims at detecting whether a person is wearing a mask or not is a real-time image. The algorithm applies a single neural network to the full image, and then divides the image into regions and predicts bounding boxes and probabilities for each region [2]. YOLOv3 works faster and is very effective.

LITERATURE REVIEW

Literature Review of Real-Time Single and Multiple Object Detection has been a widely studied area of research since the rise of Machine Learning an Artificial Intelligence (AI).

YOLO is a state-of-art algorithm for real-time object detection by Joseph Redmon and Ali Farhadi. The authors released three versions: YOLOvl in 2016, YOLOv2 in 2017, and YOLOv3 in 2018 [4].

YOLO algorithm has been proved to be effective in many object detection applications such as vehicle detection, aerial target detection, pedestrian detection, etc [4]. The feature extractor of the YOLOv3 contains 53 convolutional layers, and thus it is named Darknet-53. This is a hybrid approach between the previous network v2 and v1. The detail description of Darknet-53 is shown in Table 1 [4].

Repeat	Type	Filters	Size	Output
	Convolutional	32	3×3	256×256
	Convolutional	64	3×3/2	128×128
×1	Convolutional	32	1×1	
	Convolutional	64	3×3	
	Residual			128×128
	Convolutional	128	3×3/2	64×64
×2	Convolutional	64	1×1	
	Convolutional	128	3×3	
	Residual			64×64
	Convolutional	256	3×3/2	32×32
×8	Convolutional	128	1×1	
	Convolutional	256	3×3	
	Residual			32×32
	Convolutional	512	3×3/2	16×16
×8	Convolutional	256	1×1	
	Convolutional	512	3×3	
	Residual			16×16
	Convolutional	1024		
×4	Convolutional	512		
	Convolutional	1024		
	Residual			16×16
	Convolutional	1024	3×3/2	8x8
	Avgpool		Global	
	Connected		1000	
	Softmax			
53 convolutional layers				

Table 1. The darknet-53 model [4].

YOLOv3 uses the darknet-53 model to train weights and get them ready for testing and validation. The YOLOv3 model has high accuracy and is beneficial for real-time detection.

DATASET

KAGGLE

Face Mask Dataset (YOLO Format) - a Kaggle dataset was downloaded to use for training [7]. The dataset was made by collecting images from Google, Bing and Kaggle. The dataset contained images of people with mask and no masks. The Face Mask Detector was split into Train, Testing and Validating Sets. The images were annotated in YOLO format. Images (.jpg) and their respective annotations YOLO format (.txt) were saved in a same folder. The Training Dataset contained 700 images, the Testing Dataset contained 120 images and the Validation Dataset contained 100 images.

The following are the details of the different classes present in the dataset:

1. no_mask

A “no_mask” typically refers to an image in which a person doesn’t have his face covered with a mask. There can multiple people in a single image who can be marked with the class “no_mask” during the annotation of the image while training phase.

2. mask

A “mask” typically refers to an image in which a person has his face covered with a mask. There can multiple people in a single image who can be marked with the class “mask” during the annotation of the image while training phase.

COCO Dataset (YOLO Format)- The MS COCO (Microsoft Common Objects in Context) dataset is a large-scale object detection, segmentation, key-point detection, and captioning dataset. The dataset consists of 328K images and 80 Object Categories which include: 'person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus', 'train', 'truck', 'boat', 'traffic light', 'fire hydrant', 'stop sign', 'parking meter‘ etc.

Indian License Plate Dataset (YOLO Format) - A Kaggle dataset were downloaded to use for training. The dataset contained 1,183 images out of which 700 were used [18].

The following are the details of the different classes present in the dataset:

1. LP

A “LP” typically refers to an image in which an automobile has a License Plate.

LABELIMG SOFTWARE

The images were annotated using LabelImg Software.

The LabelImg Software was downloaded using [10]. LabelImg is a software that helps us categorize images in our desired classes and save them in YOLO format.

There are mainly two classes that were involved for Mask Annotation: **“no_mask”** and **“mask”.**

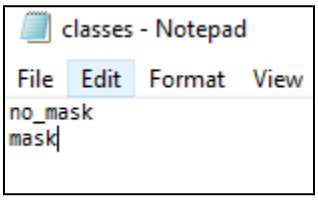

Fig 2. Classes of Face Mask Dataset.

First the Anaconda Prompt was run and the path was changed using “cd” to where the LabelImg software was saved. The LabelImg software was opened using “python labelimg.py” [8].

An example of image and annotation are as follows:

The Image:

Fig 3. A real-time image of people with and without masks [7].

The Annotation:

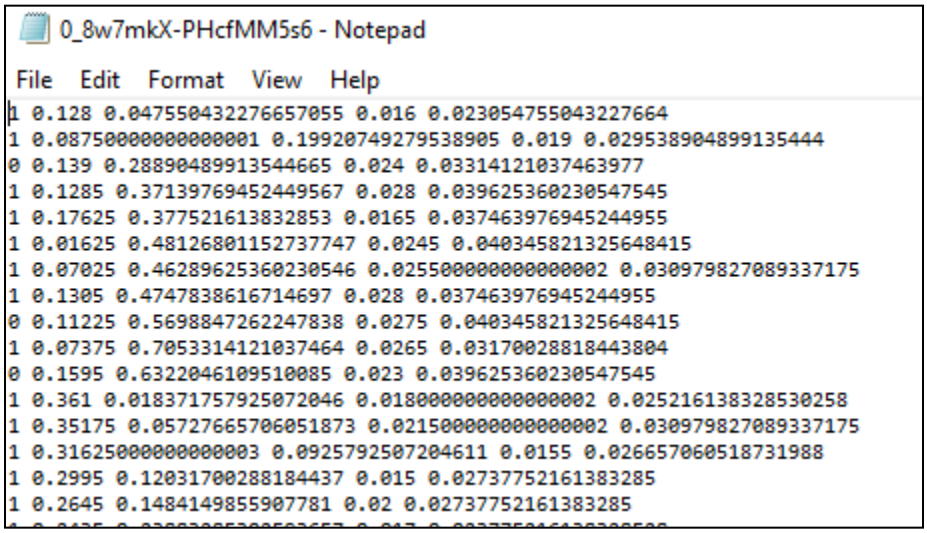

Fig 4. The annotations of the image in [Fig 3].

The Software:

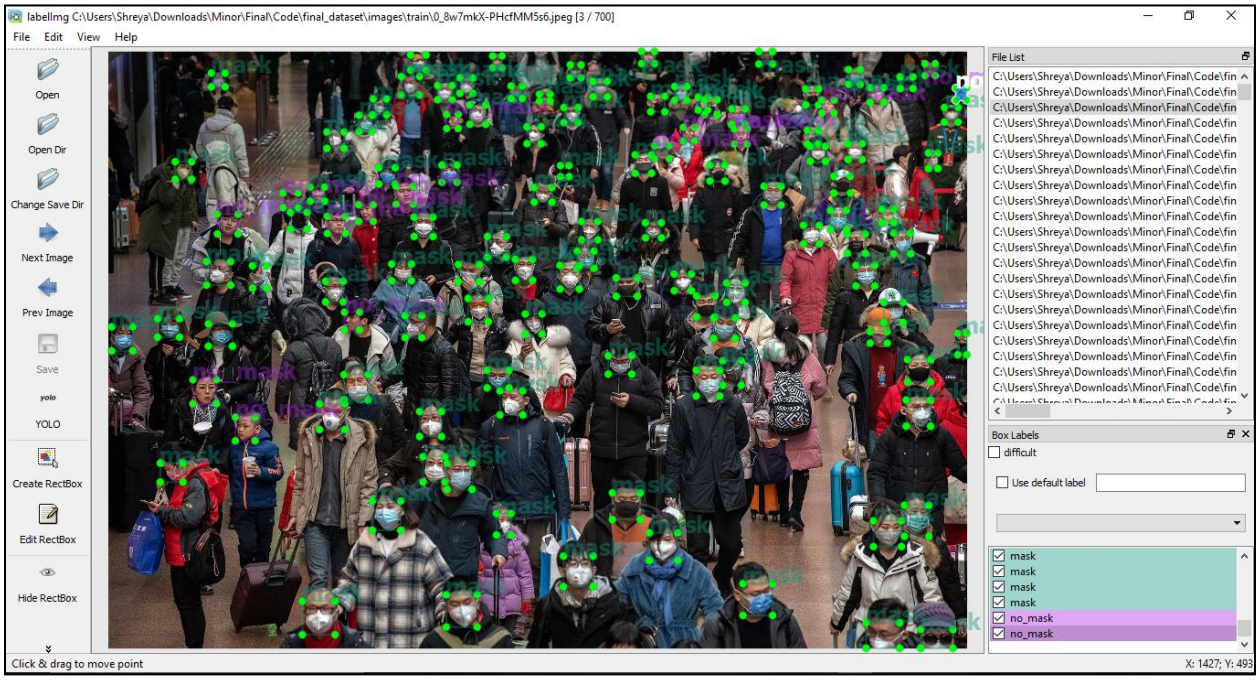

Fig 5. The LabelImg software being used over image in [Fig 3].

There is mainly one class that was involved for License Plate Annotation: **“LP”.**

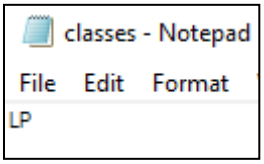

Fig 6. Classes of License Plate Dataset.

The Image:

Fig 7. A real-time image of a car with license plate [19].

The Annotation:

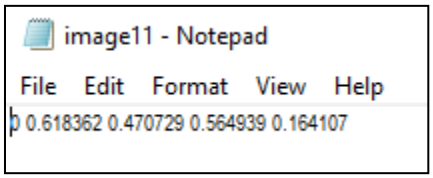

Fig 8. The annotations of the image in [Fig 7].

The Software:

Fig 9. The LabelImg software being used over image in [Fig 7].

The LabelImg software makes it easier for us to easily classify mask, road traffic and license plate to create a training dataset for detection. Similarly, remaining images were annotated and the Dataset in YOLO Format was created.

METHODOLOGY

To carry out the process of classifying items within this dataset, several methods and processes have been used. In general, Google Colab, a GPU (Graphical Processing Unit) from Google has been utilized on the datasets based on the tri-part process of training, testing and validation. Further, Convolutional Neural Network Algorithms are used to optimize and generate the training weights.

The study involved using specific methods. These include

- Darknet
- You Only Look Once Version 3 (YOLOv3) model
- OpenCV

Following are the details of all the above discussed terms and processes.

GPU

Graphics Processing Unit can be found in devices ranging from mobile phones to Nintendo with the motive to increase the rate at which images are processed and displayed on such devices that require a high degree of image processing capabilities. To achieve this, the GPUs have specific electronic circuits employed within the device, such that chunks of data are processed in a parallel manner and not linearly, as that in a general CPU (Central Processing Unit).

While GPUs have been originally used for above-mentioned image processing tasks, they can also be used for deep learning computations. Instead of leveraging the GPU's capabilities for image data, its parallel-processing methods are used for processing datasets. In fact, GPUs have now become an integral part of new-age artificial intelligence hardware infrastructure, and are even optimized specifically for deep learning tasks.

GOOGLE COLABORATORY

Nowadays, these GPUs need not exist on each one's working computer but can even be accessed in a virtual fashion, via a browser. For example, Google Colaboratory or 'Colab', a research project by Google, offers its users the opportunity to write and run Python code in their browsers

without any configuration and fees. As of October 13, 2018, Google Colab provides a single 12GB NVIDIA Tesla K80 GPU that can be used for free for 12 hours continuously [12]. The Nvidia Tesla K80 is the most popular GPU in the world. It provides commendable performance at a very low data center cost. It is composed of 4992 NVIDIA CUDA cores with a dual-GPU design and 24 GB of GDDR5 memory, and has an aggregate memory bandwidth of 480 GB/s [13]. For key applications, it delivers a performance which is 5-10 times stronger than only using the CPU or using the CPU along with the former's predecessor, the Tesla K20 [13].

CONVOLUTIONAL NEURAL NETWORKS

The Convolutional Neural Network (CNN) is a class of deep neural networks that are inspired by biological processes [3]. A Convolutional Neural Network Block typically consists of Convolution, Pooling and Non-Linear Activation.

Convolutional refers to the combination of two functions mathematically to produce a third function. In Convolutional Neural Network, the convolution takes place when a filter is used over the input data which results in a feature map [14].
Pooling helps to reduce the spatial size of the network. This leads to reduction of amount of parameters and time of computation in the neural network [15].
Non-linear Activation is used because the output can't be generated from the linear combination of the inputs [16].

More specifically, the Darknet neural network framework is used for computation purposes. Additionally, the YOLOv3 algorithm has been employed, which uses the methodology of transfer learning and custom object-detection. Lastly, OpenCV has been implemented to detect the testing images. Following are the details of all the above discussed terms and processes.

DARKNET

Darknet is written in C and CUDA and is open source neural network [5].
In this study, Google Colab along with the Tesla K80 GPU were initiated. The Google Drive was mounted in which the Dataset was saved inside a folder. The Darknet Repository from Github was cloned, configured and compiled [9].
The Darknet Repository contains neural networks which help in image recognition. More specifically, in this work, the pre-trained Darknet weights from the darknet-53 model were used to start training over the Face Mask Dataset in the YOLO format.

YOU ONLY LOOK ONCE VERSION 3 (YOLOV3) MODEL

The YOLO- You Only Look Once model is an object detection system in real-time [1].

The YOLO version 3 model has high accuracy and has high speed.

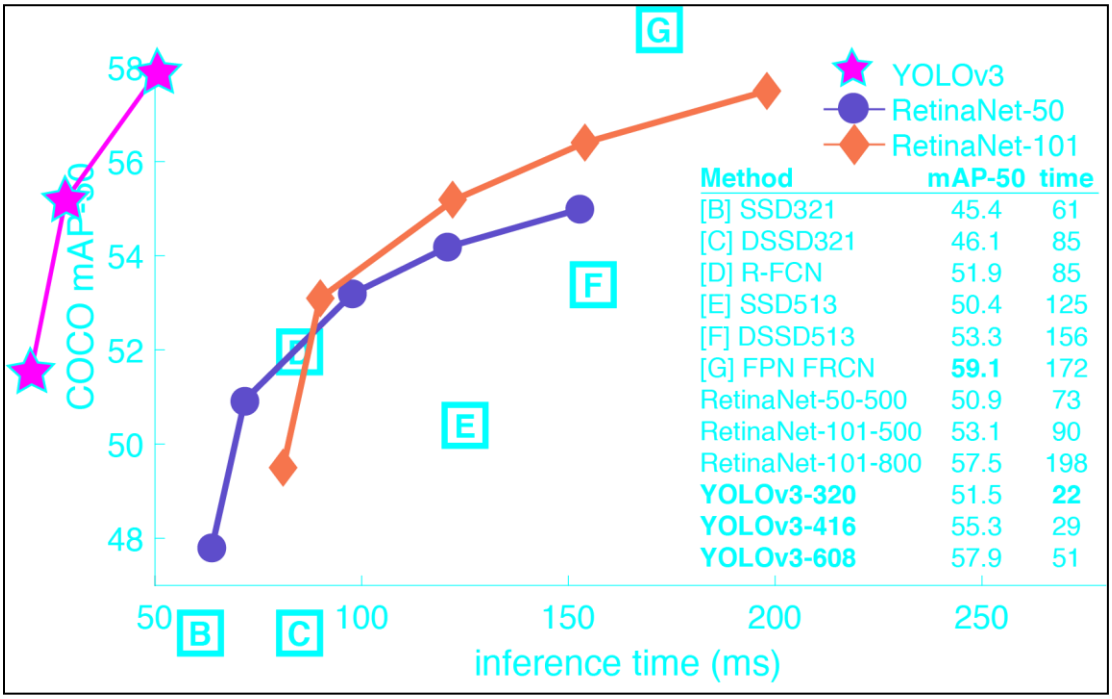

Fig 10. YOLOv3 and Coco model [1].

In this study we used YOLOv3 configuration and pre-trained weights from the Darknet-53 neural network model to train our Face Mask Dataset [6].

We have 53 layers which are the convolutional layers from Darknet [2]. For detection of the Face Mask Dataset, 53 more layers are stacked on the initial 53 convolutional layers, giving us a total of 106 layers [2].

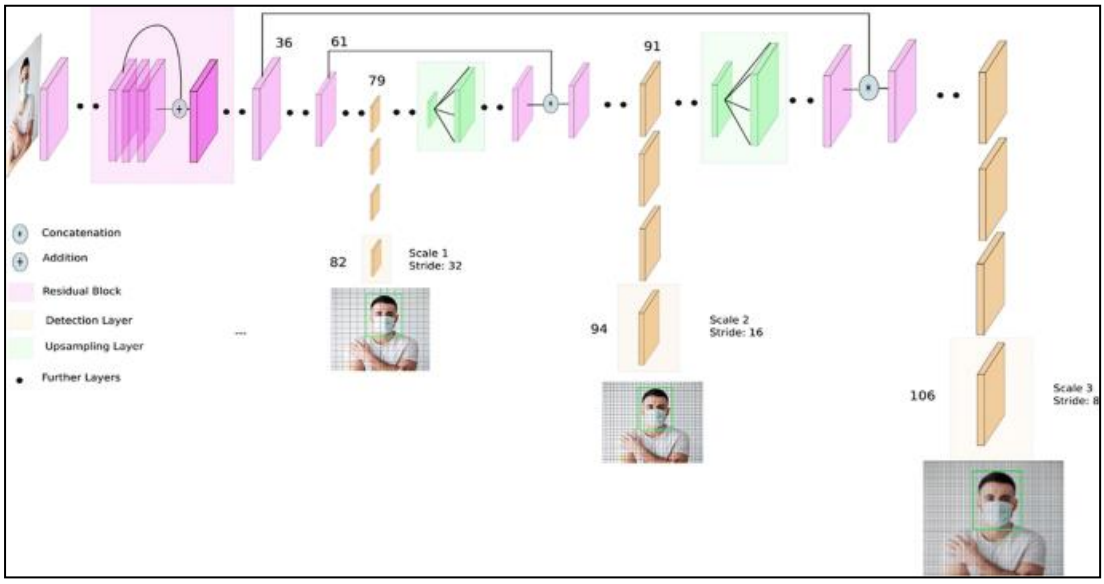

Fig 11. Architecture of YOLOv3 [2].

OPENCV

OpenCV is focused at real-time computer vision [17].

In this work, OpenCV was used to get the desired result by implementing the code for testing images/videos. The following steps were followed to get the boundary box image of the test image/video:

- Deep neural network of "cv2" and applied the readNet () function to read the configurations and trained weights from YOLOv3 and the image/video was read [8].

Fig 12. Test Image [20].

Fig 13. Test Video Snippet [21].

- Deep neural network of "cv2" and applied the blobFromImage() function to convert the image into a blob.
- Lastly, boundary-boxes and resizing of image took place using "cv2" to get the desired result.

Fig 14. Boundary-Box “Mask” Image [Fig 12].

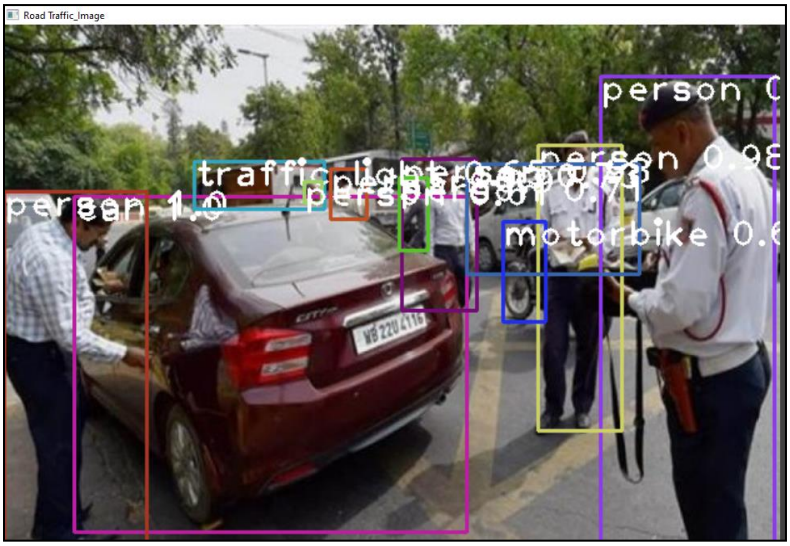

Fig 15. Boundary-Box “Road Traffic” Image [Fig 12].

Fig 16. Boundary-Box “License Plate” Image [Fig 12].

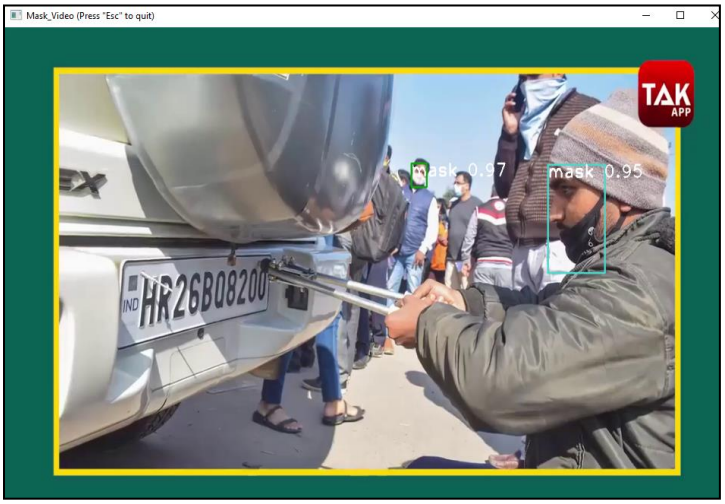

Fig 17. Boundary-Box “Mask” Video Snippet [Fig 13].

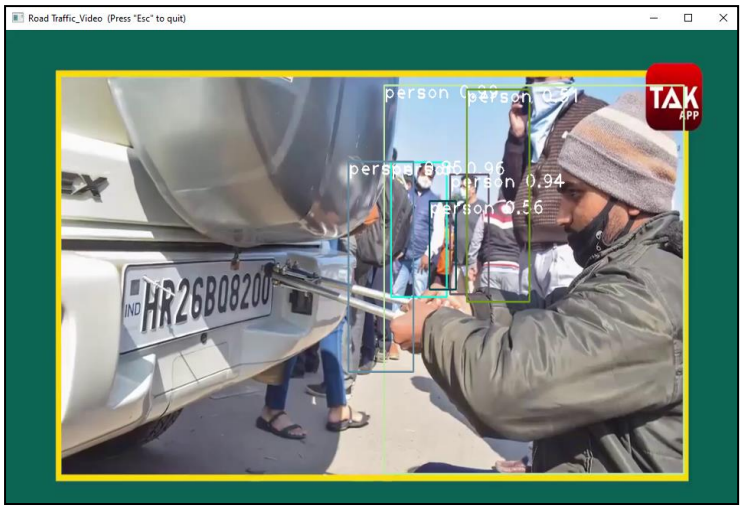

Fig 18. Boundary-Box “Road Traffic” Video Snippet [Fig 13].

Fig 19. Boundary-Box “License Plate” Video Snippet [Fig 13].

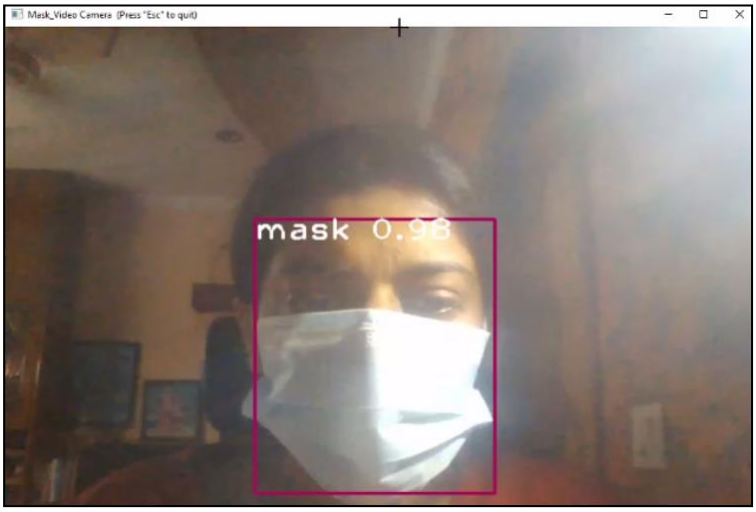

Fig 20. Boundary-Box “Mask” Live Video Snippet.

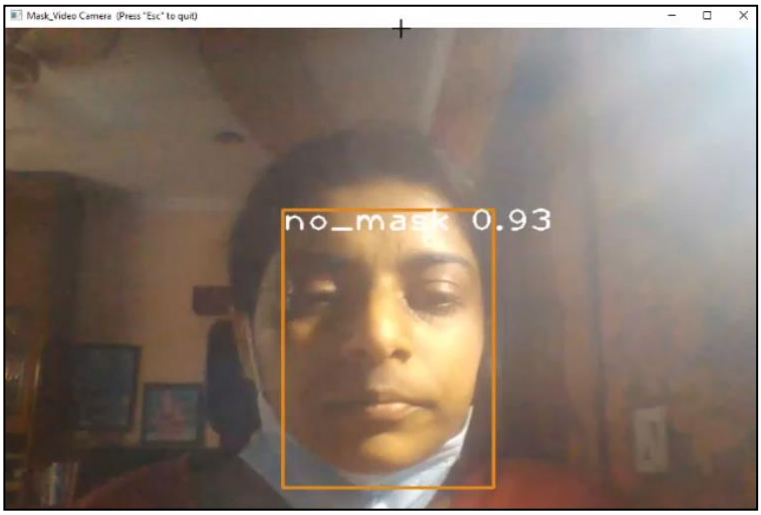

Fig 21. Boundary-Box “No Mask” Live Video Snippet.

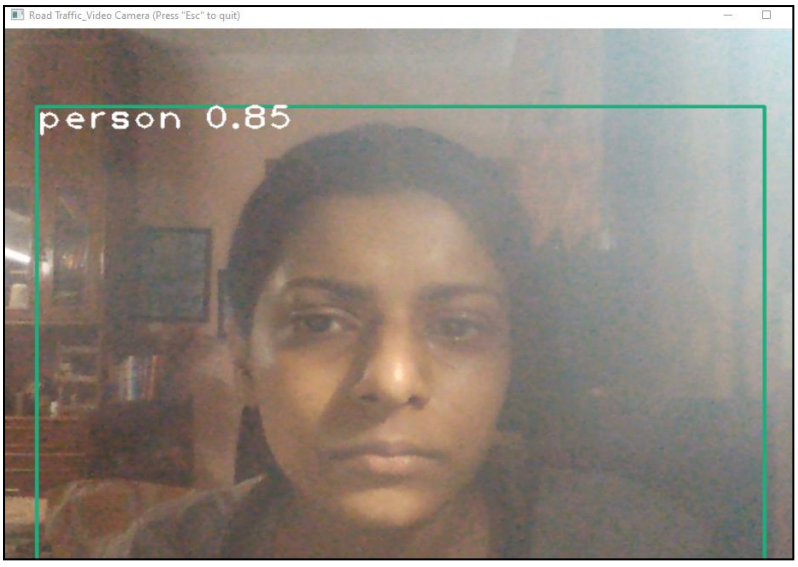

Fig 22. Boundary-Box “Road Traffic” Live Video Snippet.

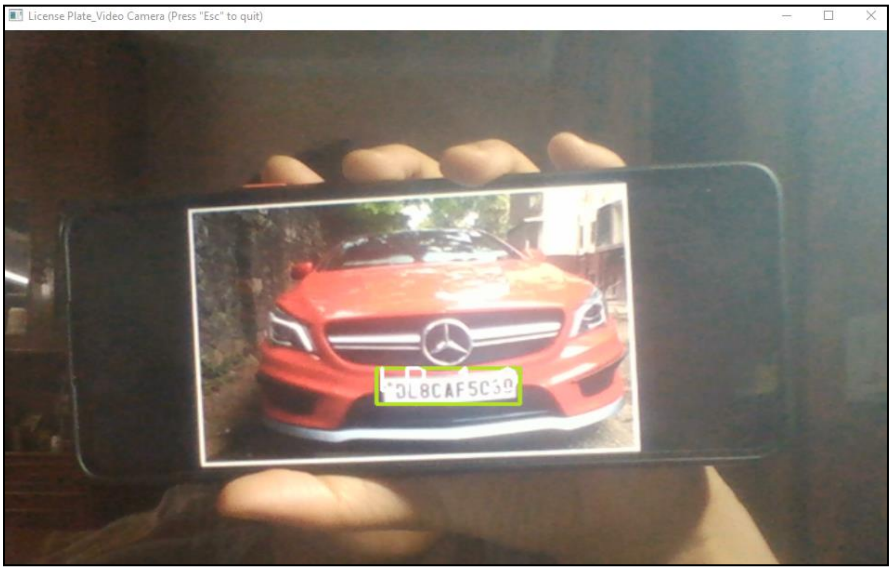

Fig 23. Boundary-Box “License Plate” Live Video Snippet.

RESULTS AND DISCUSSIONS

After successfully validating images and videos, we can conclude that the code is predicting with high accuracy.

The output images produced multicolor boundary-box images, videos and live video feed along with the class as mentioned in the datasets. The boundary box also represented the percentage of accuracy of the detection.

CONCLUSIONS AND FUTURE STUDIES

Due to Covid-19, it is important that real-time face mask detection systems are used and optimized. YOLOv3 models are preferred for these detections as they perform single-shot detection because they have high speeds [2].

Face Mask Detection System can be used for surveillance in several places like offices, public transport, public spaces and many such more. The detection system would help in keeping an easy check on the people for their own safety.

Future studies can identify ways to use the model in advanced technology in order to achieve real-time application based systems:

- Smart Cities can envelop this model to improve the way of living and healthcare [2].
- Collection of more data and inculcating different models including YOLOv3 can

contribute hugely to global health [3].

- Advanced models of YOLO can impact how we train the models as well as their accuracy.

REFERENCES

1 J. Redmon, A. Farhadi (2018): YOLOv3: An Incremental Improvement. arXiv:1804.02767
https://arxiv.org/abs/1804.02767v1

2 S. Singh, U. Ahuja, M. Kumar, K. Kumar, M. Sachdeva (2021): Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment. SpringerLink, Multimedia Tools and Applications volume 80, pages 19753–19768.
https://doi.org/10.1007/s11042-021-10711-8

3 X. Jiang, T. Gao, Z. Zhu, Y. Zhao (2021): Real-Time Face Mask Detection Method Based on YOLOv3. Electronics 2021, 10(7), 837.
https://doi.org/10.3390/electronics10070837

4 T. Q. Vinh and N. T. N. Anh (2020): Real-Time Face Mask Detector Using YOLOv3 Algorithm and Haar Cascade Classifier, 2020 International Conference on Advanced Computing and Applications (ACOMP), pp. 146-149.
https://doi.org/10.1109/ACOMP50827.2020.00029

5 J. Redmon (2013-2016): Darknet: Open Source Neural Networks in C.
http://pjreddie.com/darknet/

6 Github: emasterclassacademy/Single-Multiple-Custom-Object-Detection (last updated 2020)
https://github.com/emasterclassacademy/Single-Multiple-Custom-Object-Detection

7 Kaggle: Face Mask Dataset (YOLO Format) (2020)
https://www.kaggle.com/aditya276/face-mask-dataset-yolo-format

8 Github- adiyap27/face-mask-detector (last updated 2021)
https://github.com/adityap27/face-mask-detector

9 Github- Alexey/darknet (last updated 2021)
https://github.com/AlexeyAB/darknet

10 LabelImg Software (last updated 2021)
https://tzutalin.github.io/labelImg/

11 Visio.ai (2021): YOLOv3: Real-Time Object Detection Algorithm
https://viso.ai/deep-learning/yolov3-overview/

12 Ori Bar-El (2018): Getting the Most Out of Your Google Colab (Tutorial). Medium.
https://medium.com/@oribarel/getting-the-most-out-of-your-google-colab-2b0585f82403

13 NVIDIA Tesla K80: Cloud & Data Center.
https://www.nvidia.com/en-gb/data-center/tesla-k80/

14 Daphne Cornelisse (2018): An intuitive guide to Convolutional Neural Networks. freeCodeCamp.
https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/

15 Harsh Pokharna (2016): The best explanation of Convolutional Neural Networks on the Internet! Medium.
https://medium.com/technologymadeeasy/the-best-explanation-of-convolutional-neural-networks-on-the-internet-fbb8b1ad5df8

16 Doug (2012): Why must a nonlinear activation function be used in a backpropagation neural network? Stack overflow.
https://stackoverflow.com/questions/9782071/why-must-a-nonlinear-activation-function-be-used-in-a-backpropagation-neural-net

17 K. Pulli, A. Baksheev; K. Kornyakov, V. Eruhimov (2012): Realtime Computer Vision with OpenCV. Queue. 10 (4): 40:40–40:56.
https://dl.acm.org/doi/10.1145/2181796.2206309

18 Kaggle: License Plate Dataset (2018)
https://www.kaggle.com/datasets/thamizhsterio/indian-license-plates

19 Open Source Train Image (2020)
https://indusscrolls.com/wp-content/uploads/2020/11/high-security-number-plate-630x420.jpg

20 Open Source Test Image (2016)
https://cdn.dnaindia.com/sites/default/files/styles/half/public/2016/04/18/450837-delhi-oddd-even-pti.jpg\

21 Open Source Test Video (2021)
https://www.youtube.com/watch?v=bkjAtmeMHaw