REAL TIME FACE MASK DETECTION USING MACHINE LEARNING

[1]Seeju Singh, [1]Ritu Swami, [3]Mohan V Bonde

[1]B. Tech Student, [2]Asst. Professor,
[1]Usha Mittal Institute Of Technology, [2]Usha Mittal Institute Of Technology,
SNDT Women's University, Mumbai, India.

Abstract: The outbreak of Coronavirus disease has thus far killed over 2.85M people and infected over 131M all over the world, causing global health crisis. Due to this the government was forced to impose lockdown all over the world. As made mandatory by World Health Organization (WHO), the only effective protection method is to wear face mask every time we are out in public and maintain social distancing. Wearing face masks will automatically reduce the risk of spreading of the deadly virus. An efficient approach used for building Deep learning model for face detection will be presented. Here, we will have dataset that consists images that are with mask and without mask and later use OpenCV real-time face mask detection from our webcam. We will use the dataset to build a COVID-19 face mask detector with computer vision using Python, OpenCV, and Tensor Flow and Keras. Our aim to identify if the person in the image/video is masked or unmasked. The model achieves 98.7% accuracy on distinguishing people with or without face mask. We hope that our study would be useful to reduce the rapid spread of virus.

Keywords— Machine learning, Deep learning, computer vision, face mask, OpenCV, TensorFlow, Keras, CNN, MobileNetV2, YOLO.

I. INTRODUCTION

The World Health Organization (WHO) reports suggest that the two main routes of transmission of the COVID-19 virus are respiratory droplets and physical contact. Respiratory droplets are generated when an infected person coughs or sneezes. Any person in close contact (within 1 meter) with someone who has respiratory symptoms (coughing, sneezing) is at risk of being exposed to potentially infective respiratory droplets. As soon as the first patient was detected in December, COVID-19 has become pandemic all over the world which led us all to challenging situations. Everyday large number of people are infected and died. At the time of writing this paper, almost 131M infected cases have been confirmed and 2.85M are death[1]. The number keeps increasing day by day. Fever, dry cough, tiredness, diarrhea, loss of taste, and smell are the major symptoms of coronavirus which is declared by the World Health Organization (WHO). Many precautions have been taken to fight this virus such as washing hands, maintaining social distancing in public areas, wearing masks, avoiding touching eyes, nose and mouth area, where wearing mask is the simplest one. By ensuring proper use of face mask, we can stop the spread of COVID-19. As we can see, the second wave of the virus has hit our country with mutation the virus. Therefore, we can still limit the spread if people strictly maintain social distancing and wear facial mask. But sadly, people are not obeying the protocols which is speeding the number of cases around the globe. Hence, detecting these people who are not following the rules and informing the authorities can be a solution in reducing the spread of coronavirus.

Artificial Intelligence (AI) based on Machine learning and Deep Learning can help to fight Covid-19 in many ways. A face mask detection is a technique to find out whether someone is wearing a mask or not. Object detection plays major role in this. Deep learning techniques are highly used in medical applications. These techniques can be incorporated in detecting the mask on face. This project can be applicable anywhere but specially in cities that are highly populated which have IoT sensors to collect data. In section II, illustration of related work has been mentioned followed by section III in which, detailed information about technologies used in this project is explained. In section IV overall design of the project has been explained. In section V the implementation of the model is illustrated. In section VI the result of the model is shown with comparisons and graphs.

II. RELATED WORK

In the meantime, many systems have been developed for COVID-19. Deep learning has progressed quite fast over the past few years and implemented in almost every field today. Likewise, Convolutional neural network or CNN which is type of feed forward artificial neural network in which connectivity pattern between its neurons is inspired by human eye is found to be used in many applications of almost every domain. A study on using facemask to restrict the growth of COVID-19 has been introduced in many papers. This study is the evidence of how wearing mask can reduce the transmissibility. Public mask wearing is most effective at reducing spread of virus when compliance is high. This studies also show that even the cloth mask can be as effective as wearing N-95 or surgical masks. "Mask classifier" has been introduced, where web scrapped data has been used with classes people wearing mask and no mask and trained it. Different algorithms were used such as Haar cascade classifier and CNN. Haar cascade classifier is used in object detection. It is machine learning based approach where many positive and negative images are used to train the classifier. Positive images focus on what our classifier wants to identify whereas negative images are everything else which do not contain the object we want to detect. Another system was built by Mk Gurucharan called "Face-Mask-Detection" in which the model was built using TensorFlow framework and OpenCV library which is highly used of real-time application. It gives accuracy rate of 96% after using algorithms like MoblieNetV2[4]. Another system was built named "Mask Classifier" by Prajna Bhandary in which scrapped data was used with classes "with mask" and "without mask" and trained resent 50[8]. Since mask is an object and our region of interest, we came across various object detection studies. Object detection is inextricably linked to other similar computer vision techniques like image recognition and image segmentation which helps us to understand and analyses scenes in images. An object detection model named "Region-based Convolutional Network for Accurate Object Detection and Segmentation"[15] has been proposed. The developed model includes two stage frame work i.e., region proposal stage and region classification and refinement stage. Another model was developed by Joseph Redmon, Ali Farhadi named “An Incremental Improvement”[2]. YOLOV3 came in picture after YOLOv2 which resulted in more powerful backbone feature extractor and RetinaNet-like detection. It fully embraced FPN’s multi-scale predictions design.

III. TECHNOLOGY ADOPTED

Convolutional Neural Network

A convolutional neural network, or CNN, is a deep learning neural network which are very easy at identifying on designs in input images, such as lines, gradients, circles, eyes, nose and faces. In neural network there are mainly 3 layers: input layer, hidden layer, output layer. The input layer we give input to our model. The total number of neurons present in this layer is equal to number of features in our data. The input from this layer is then feed to hidden layer. Hidden layer may vary depending on our data size and mode. Each hidden layer has different neurons. The output is calculated by matrix multiplication of output of previous layer with suitable weights and later adding the activation function which makes the layer nonlinear. Finally, the output layer is then fed into logistic functions like SoftMax. Convolution layers has set of filters which has width and height and same depth as that of input. For instance, image with dimension 34x34x3. Here size of filters can be axax3, where ‘a’ can be 3,5,7 etc. but small compared to image dimension[6].

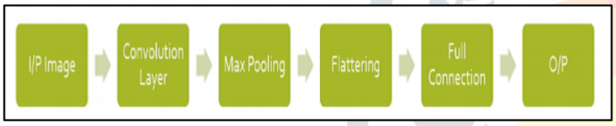

Figure 1. CNN Model

The input layer in Fig 1 holds the raw input of image of width 32, height 32 and depth 3. Convolution layer gives output by calculating dot product between all filters and image patch. Suppose there are 14 filters, then output is of dimension 32x32x14. Activation function like RELU is used which is given by 1/(1+e^-x). Pool layer reduces the size of volume resulting in fast computation and prevents from overfitting. Lastly fully connected layer takes input from previous layer and converts it into 1-D array of size equal to number of classes.

MobileNetV2

MobileNetV2 is the next generation of mobile vision applications. It is improvement over MobileNetV1. Having 53 layers deep where more than 1 million images can be pretrained, it is an effective feature extractor for object detection and segmentation. MobileNetV2 are faster with same accuracy. Two features are added to the architecture as seen in Fig 2 1) Linear bottlenecks between the layers and 2) shortcut connection between bottlenecks. The architecture of MobileNetV2 contains fully convolutional layer with 32 filters, followed by 19 residual bottleneck layers[11]. It delivers high accuracy results while keeping the parameters and mathematical operations as low as possible.

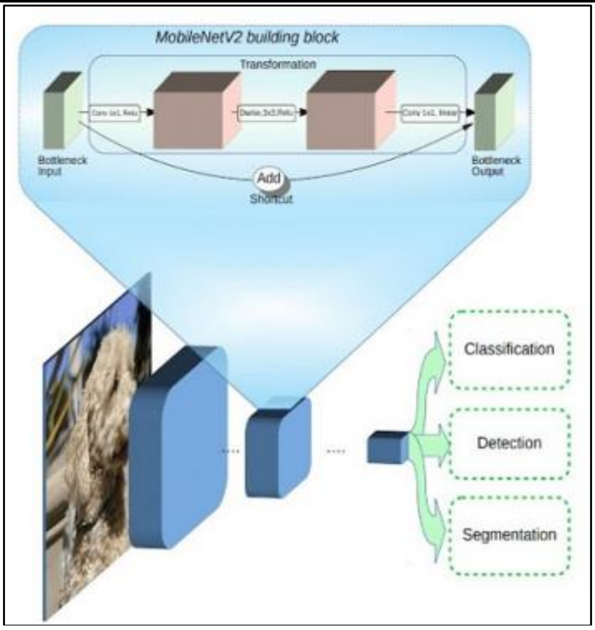

Figure 2. Architecture of MobileNetV2

Yolo

YOLO (You only look once) is convolutional neural network for doing object detection in real-time. A single neural network algorithm is applied to full image which divides the images into regions along with the prediction of bounding boxes. These bounding boxes are weighted by predicted probabilities. It is one of the popular techniques because it can achieve high accuracy in real-time. It requires one forward propagation pass to make predictions. Once it checks that the object detection detects each object only once, then it gives outputs with the bounding boxes. For each bounding box encloses an object and probability of the enclosed object being a particular class. There are many versions YOLO that have been released. Three major version of YOLO are YOLO V4, YOLO V5 and PP-YOLO. It takes image and divides it into grid of n x n. A pixel is responsible for prediction. Out of all detected boxes, only one object is observed and rest other detection are rejected[12]. Figure 4 below shows the flow of how the process is carried out in YOLO.

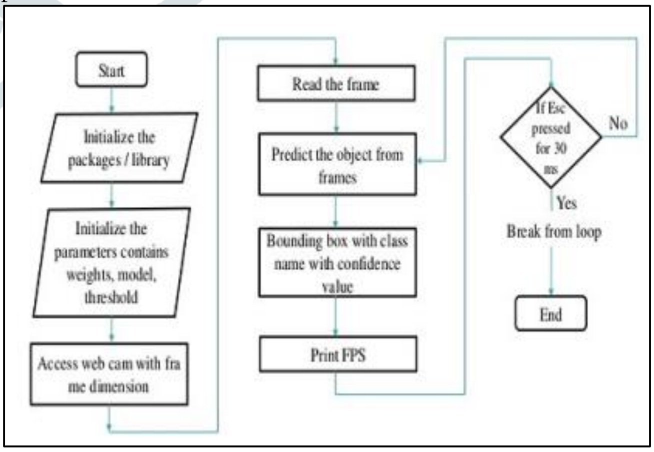

Figure 3. Flowchart of YOLO

IV. DESIGN

i. *Face Detection*

Face detection has been active area of research in past two decades. Facial tracking is finding more growth of use in security and safety applications to detect various situations. The basic idea of face detection is to construct facial features by down sampling components of face such as eyes, nose, mouth and whole face. In this approach, cascade function is trained from lot of positive images (images with faces) and negative images (images without faces). Then features are extracted from the images which are just like convolutional kernel[3]. Fig 4 is illustration of face detection model.

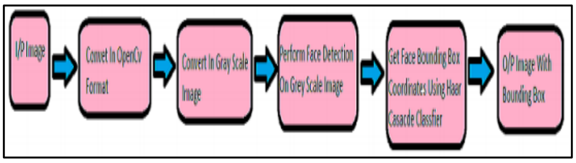

Figure 4. Face Detection Model

Figure 5. Output of Face Detection

ii. *Object Detection*

An object detection system finds objects of the real-world present either in a digital image or a video, where the object can belong to any class of objects namely humans, cars, etc. The basic input is the image or a video from which an object is to be detected. Once the objects are detected, then the system simply needs to categorise various objects into respective object classes. In our system, there will be two classes which are people with mask and people without mask. Before giving the output, it passes two phases namely: the learning phase and testing phase. After pre-processing is done on image, template matching is done that generates features of object in image. The main aim of testing phase is to see if object is present in an image and if yes, then which object class it belongs to. The Fig 6 shows object detection model.

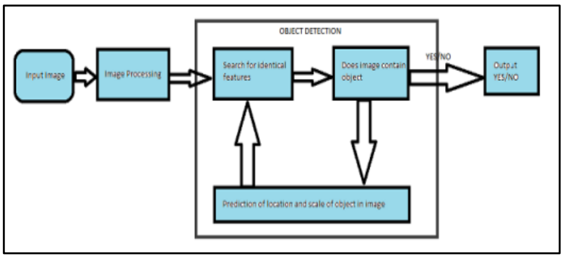

Figure 6. Object Detection Model

Figure 7. Output of object detection

V. IMPLEMENTATION

iii. *Face Mask Detection using CNN and MobileNetV2*

Due to high rising covid cases, face mask detection applications are highly in demand. This system can be used in real-time applications which require face mask detection for safety purposes. Here we have implemented face mask detection using MobileNetV2 and convolutional neural network. The dataset consists of 4095 images belonging to two classes: with_mask: 2165 images without_mask: 1930 images. For better understanding below is the algorithm:

Step 1: Start
Step 2: Initialize the webcam
Step 3: Extract image from the frame
Step 4: Load face detection model and detect face
Step 5: If face is detected apply image processing else return to Step 3.
Step 6: Load the face mask detection model and detect the masked face.
Step 7: If mask detected display bounding box and accuracy percentage with title "mask" else display bounding box and accuracy percentage with title "no mask".
Step 8: Return to Step 3.

Figure 8. Output of face mask detection

Figure 9. Output of face mask detection

iv. *FACE MASK DETECTION USING YOLOV4*

Using another algorithm, YOLOV4 we have again detected the face mask. Calculating the accuracy is bit complex but the output is more accurate and speed is fast. It has various layers: - Input: where image is taken. Backbone: object detection in image. Neck: feature maps from different layers are collected. Head: output with bounding boxes and classes for objects. The dataset for this pretrained network is provided by VictorLin000[18] and contains 678 images of people with and without masks. It has improved the YOLOV3's AP and FPS by 10% and 12% respectively. For better understanding below is the algorithm:

Step 1: Start
Step 2: Clone Darknet
Step 3: Load the helper function
Step 4: Load YOLO V4 weights
Step 5: Connect to drive
Step 6: Load the data sets
Step 7: Train the model
Step 8: Test the model on images/videos
Step 9: If mask detected display bounding box and accuracy percentage with title "mask" else display bounding box and accuracy percentage with title "no mask".
Step 10: Run the model with real time webcam.
Step 11: Outcome will be same as Step 9
Step 12: Return to step 10.

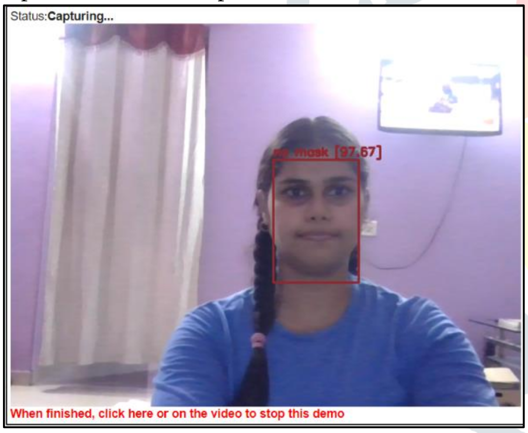

Figure 10. Output of face mask detection

Figure 11. Output of face mask detection

Figure 12. Output of face mask detection

VI. RESULT

a) Result of face mask detection using YOLOV4

With dataset of 687 images which contains images of people with mask and without mask after iterating the model with 5000 iteration. We can successfully detect whether the person is wearing mask or not.
Average loss = 0.3210
Mean average precision = 85%

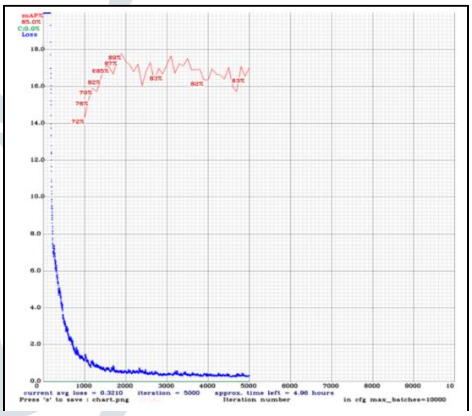

Figure 13. Graph of iteration number Vs loss

b) Result of face mask detection using CNN and MobileNetV2
With dataset of total 4095 images in which, with_mask: 2165 images without_mask: 1930 images, we trained the model and it can successfully detect if a person is wearing mask or not.
Average loss = 0.0827
Mean average precision with mask = 98%
Mean average precision without mask = 99%

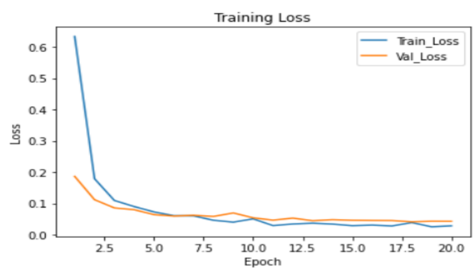

Figure 14. Loss curve of CNN & MobileNetV2 model

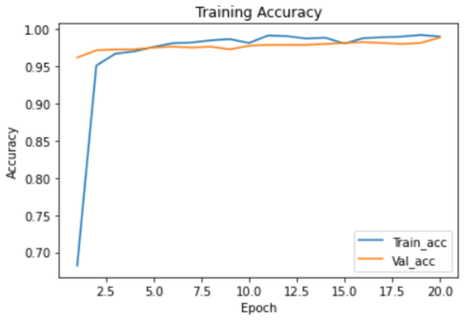

Figure 15. Accuracy curve of CNN & MobileNetV2 model

CONCLUSION

With the rapid rise of Covid-19 cases, government was forced to impose lockdown all over the world. And wearing mask has become necessary part of our lives as suggested by WHO. Hence, in this project we have implemented two Real Time Face Mask Detection Model one with MobileNetV2 and CNN. Another with YOLOV4. In comparison to both, the model with CNN and MobileNetV2 is much better than YOLOV4 as its accuracy is 98% where as YOLOV4 accuracy is 88.92. Since we used the MobileNetV2 architecture, it’s also computationally efficient and thus making it easier to deploy the model to embedded systems (Raspberry Pi, Google Coral, etc.).

REFERENCES

[1] Worldometers.info. 2021. *India COVID: 26,930,220 Cases and 306,865 Deaths - Worldometer*

[2] J. and Farhadi, A., 2021. YOLOv3: An Incremental Improvement | Request PDF

[3] GreatLearning. 2020. Real-Time Face Detection | Face Mask Detection Using Opencv.

[4] GitHub. 2020. Mk-Gurucharan/Face-Mask-Detection.

[5] Kumar, S., Balyan, A. and Chawla, M., 2021.

[6] Medium. 2020. A Comprehensive Guide To Convolutional Neural Networks—The ELI5 Way.

[7] The Times of India. 2020. Coronavirus: People Who Don't Wear A Mask Have Antisocial Traits, Finds Brazilian Study | The Times Of India.

[8] GitHub. 2020. Prajnasb/Observations.

[9] GeeksforGeeks. 2020. Python | Haar Cascades For Object Detection – Geeksforgeeks.

[10] Rosebrock, A., 2020. COVID-19: Face Mask Detector With Opencv, Keras/Tensorflow, And Deep Learning – Pyimagesearch.

[11] Sinha, S., Sinha, S. and Sinha, S., 2021. Why Google’S Mobilenetv2 Is A Revolutionary Next Gen On-Device Computer Vision Network.

[12] Started, G., Tensorflow, K., Guide, R., Courses, O., (Old), C. and Consulting, A., 2021. Deep Learning Based Object Detection Using Yolov3 With Opencv (Python / C++) | Learn Opencv. [online] Learn OpenCV | OpenCV, PyTorch, Keras, Tensorflow examples and tutorials.

[13] Medium. 2021. Face Detection Using OpenCV With Haar Cascade Classifiers.

[14] Opencv-python-tutroals.readthedocs.io. 2021. Face Detection using Haar Cascades — OpenCV-Python Tutorials 1 documentation.

[15] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Region-Based Convolutional Networks for Accurate Object Detection and Segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 1, pp. 142-158, 1 Jan. 2016, doi: 10.1109/TPAMI.2015.2437384.

[16] Ahmad, F., Najam, A. and Ahmed, Z., 2021

[17] Ideas2it. 2021. Face Mask Detector using Deep Learning (PyTorch) and Computer Vision (OpenCV) - Ideas2it.

[18] GitHub. 2021. *VictorLin000/YOLOv3_mask_detect*.

[19] Google.com. 2021. face mask detection model using mobilenetV2 - Google Search.