Covid Face Mask Detection Using Neural Networks

Rajaram Yadav[1], Safal Gautam[2], Rahul Ratna Das[3], Prarthana Shiwakoti[4]

[1 2 3 4] U.G. Student Of Computer Science and Engineering Department, Vellore Institute of Technology, Vellore, Tamil Nadu, India

---***---

Abstract - *It has often been attributed that face masks can prevent the spread of COVID-19. Many scientists argue that it prevents virus-carrying droplets from reaching other hosts (people) while coughing and sneezing. This helps break the chain of spread. However, people do not like to cover their face with a proper face mask and some of them do not know how to wear it properly. Checking it manually for a large group of people, especially at a crowded place like a train station, theater, classroom or an airport, can be time-consuming and expensive. Also, people can be biased and gullible. Therefore, an automated, accurate and reliable system is required for the task. To train the system adequately, lots of data is required: images. The system should recognize if a person is not wearing a face mask at all, wearing it improperly or if the one is wearing it properly. In this paper, we are using MobileNetv2, which is a convolutional neural network (CNN) based architecture, to build such a face mask detection/recognition model. The developed model can classify people who are wearing masks, not properly wearing and not wearing it with an accuracy of 97.25 percent.*

Key Words: COVID-19; Face-mask; Face recognition; CNN; MobileNetv2

1. INTRODUCTION

The detrimental effects of COVID-19 have been prevailing for around three years now. It has affected people all around the world and there is no sector left which has not been touched upon by the impact of this pandemic. The negative impacts are easily seen in the health sector, economic sector, tourism sector, et cetera. This global impact could be reduced significantly if most people were more aware of wearing face masks properly. A surveillance system that can detect the mask-wearing population can be a huge aid in this scenario. Hence, a mask detection system is of huge significance as people can be made cautious about efficiently wearing masks in public areas. The massive impact of the pandemic can be minimized if the system is implemented carefully and effectively.

Despite many guidelines and warnings by WHO and government, many people still do not wear masks and it has contributed a lot to the spread of the virus. Wearing a mask is a must and it should be monitored carefully for preventing the spread of COVID-19 and many other airborne diseases. So, a mask detection system is currently essential.

And it is about time that we care more and be responsible enough so that any new wave of novel coronavirus does not wreak havoc. Vaccination programs are being conducted worldwide and many people have been vaccinated. But the vaccines do not guarantee "No COVID". So, we need to keep following health guidelines for the prevention of the spread of COVID-19. The work proposed in our paper has real-world application in several crowded places like bus/railway stations, hospitals, offices, educational institutions and so on.

2. Literature Review

In paper [1], the authors propose a PCA (Principal Component Analysis) facial recognition system. Principal Component Analysis (PCA) is a statistical method under the heading of factor analysis. The goal of PCA is to reduce the large amount of data storage to the size of the function space required to represent data economically. The broad one-dimensional pixel vector composed of two-dimensional facial images in the compact main elements of the spatial function is designed for PCA facial recognition. This is called self-space projection. The appropriate spacing is determined by identifying the vectors of the covariance matrix itself, which are centered on the collection of fingerprint images.

In paper [2], the authors describe a new mobile architecture, MobileNetV2, that improves the latest performance of mobile models in multiple tasks and benchmarks and different model sizes. They also described effective methods for applying these mobile models to object detection in a new framework that is called SSDLite. This article introduces a new neural network architecture specifically designed for mobile and resource-constrained environments. Our network promotes the latest model mobile personalized computer vision technology by significantly reducing the number of operations and memory required while maintaining the same precision.

In paper [3], the methods of usage of AI in combating the problems associated with Covid-19 and likewise epidemics have been discussed. The authors of this paper have described various ways in which we can understand the clinical problems better using AI. They have presented their case on the basis of the fact that there has been a surge in clinically available data together with the increase in the hype about AI. The combination of these two could help the doctors to prescribe medicines better and help us to understand the causative and preventive methods for Covid-

19. They have also conducted a survey to demonstrate the usefulness of AI in the medical world with 90% accuracy.

In paper [4], it has been highlighted that the current medical facilities are not adequate to deal with a pandemic like situation. According to the authors the solution to this problem could be found in the form of blockchain and artificial intelligence. The authors have discussed how the use of blockchain can be helpful in predicting the early outbreak of the pandemic and recognizing the high-risk zones. Similarly, they have also discussed that the use of artificial intelligence can be taken as an intelligent measure to know the symptoms of the disease. They have gone on to introduce a state-of-the-art system that collaborates block chain and AI and this combined method could be an interesting example about how to deal with the pandemic effectively.

In paper [5], the authors have stated that Analog devices Inc.'s Cross core embedded studio and HOG SVM were used for detecting person and distance from camera. Face detection and face parts like eyes, nose and mouth is implemented by Viola Jones's algorithm. Viola Jones face detection procedure classifies images based on the value of simple features. There are three features, namely two rectangle, three rectangle and four rectangles. The value of a two-rectangle feature is computed by calculating the difference between the sum of the pixels within two rectangular regions. This proposed work may not have been able to detect the person when they are wearing a mask so to improve this accuracy of eye detection can be increased to help recognizing the person through his eye and eye line.

In paper [6], the authors have pointed out that to configure YOLOv3 object names created to contain the name of the classes which model needs to detect, an input image is passed through the YOLOv3 model, the object detector finds the coordinates that are present in an image. For producing model output the neighboring cells with high confidence rate of the features were added in the model output. 80 % of the data was used for training and rest is for validation. Fast RCNN object detection architecture can be used with YOLOv3 or the new version of YOLOv4 to increase the performance of the face detection system in real time video surveillance.

In paper [7], the authors have stated that Transfer learning has been used by using a pre-trained model Mobile Net to use existing solutions to solve new problems. Global Pooling block transforms a multi-dimensional map into a 1D vector having 64 characteristics. Finally, a SoftMax layer with 2 neurons takes the 1D vector and performs binary classification.

In paper [8], the authors' proposed method consists of a cascade classifier and a pre-trained CNN. For image in the dataset 1) Visualize the image in two categories: mask and no mask. 2) Convert the RGB image to Grey-scale image and resize this image into 100 X 100. 3) Normalize the image and convert it into a 4D array. To build the CNN model a convolution layer of 200 filters have been added and a second layer of 200 filters. A flatten layer to the network classifier has been added. In the end a final dense layer with 2 outputs for 2 categories has been inserted and the model is trained.

In paper [9], YoloV4 has been implemented using two stage detectors. The first-stage detector consists of: Input- Resolution of 1920*1080. Backbone- Darknet53 chosen as detector method contains 29 Convolutional layers by 3*3 and each layer sent to the neck detector. Neck-PANet applied as the neck detector method. Dense prediction- YOLO v3 model used in this stage to generate the prediction. Second-stage detector: It has a sparse prediction which applies the faster R-CNN. Input to this is 3*3 layers got from the neck and the input prediction from the dense prediction.

In paper [10], collect data (masked and unmasked data), pre-processing (resizing, converting to array, pre-processing using MobileNetV2, etc.), split data (75:25), build model, testing, and implementation. MobileNetV2, a convolutional neural network architecture is used. The model has 96.85 accuracy. The data collected from different cities can be used for statistical analysis of people wearing masks and appropriate action could be taken for preventing the spread of COVID-19.

3. Gap Analysis

Previously, researchers generally used heavy and complicated models such as VGG16, which needs many backend network parameters and requires high hardware configuration. Hence, it will be difficult to run on mobile devices with limited computational power. Also, even though they are heavy and complicated, they provide a similar performance to other light-weight neural network models. In other papers, where they used YOLOv4, the accuracy obtained was mere 94.75%, which is less than our proposed augmented MobileNetv2 model 3. Most of the pre-existing face-mask detection models were based on a single subject and there seemed to be a problem in detecting faces in crowded scenarios. Also, in the majority of the papers, the use of low-resolution and highly compressed images was not prevalent during the training of the models. Hence, their models were not performing that good on these highly compressed images.

4. Methodology

The architecture of this approach can be explained as below.

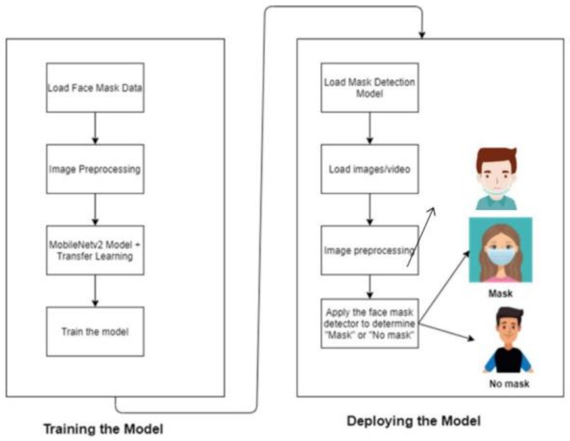

Fig 1. Architecture diagram of Face mask recognition

Above architecture explains the deep learning model that has been created followed by an image processing pipeline. The steps followed in the architecture can be described as follows:

3.1 Data gathering

The development of our face mask recognition model starts with a collection of Kaggle dataset. The referred dataset contains three different classes (with mask, without a mask and wearing mask incorrectly). Then the trained model can categorize input and detected images into wearing masks correctly, wearing masks incorrectly and not wearing masks.

The dataset is highly imbalanced and uncleaned. For proper fitting, images are augmented in such a way that each class has an equal distribution of images and removing noisy images which could be considered as outliers.

Sample labelled images (mask weared incorrectly, no mask, mask) taken from the dataset for training and testing are shown below:

Fig 2. Incorrectly wearer mask

Fig 3. Mask weared

Fig 4. Without Mask

3.2 Image pre-processing

First step here begins with the iteration through all the images and storing the labels

of each image (without mask, with mask and wearing mask incorrectly). The image sizes are non-uniform. We need some standard size so that we can feed it into our neural network. So, it is resized to 224 x 224 x 3 image which is our standard image size. Then the image is converted to a NumPy array. CV2 is better with BGR channel images, and we are using it in a later phase, so we convert our image into BGR channel. After the image is preprocessed, it is ready to feed it into our model.

3.3 Splitting data

The data was split into two collections, which are training data named 80 percent, and the rest of the part (20 percent) is testing data. Each collection consists of all the with-mask, without- mask and wearing mask incorrectly images.

3.4 Training the model

Steps:

a. Read the dataset and store the image-paths.
b. Iterate through all the images and
   i. Store the labels of each image (without mask/ with mask).
   ii. Convert image to array and preprocess the image.
   iii. Append the image to data.

c. Convert data and labels to a NumPy array.
d. Transform multiclass levels to binary levels using labelBinarizer.
e. Split the training and testing data. (20% data is used in testing and 80% data is used in training).
f. Create instantiation of mobilenetv2 model and remove the last layer.
g. Add Average pooling 2D, Flatten, Dense (with relu activation), Dropout and Dense (with SoftMax activation) layers.
h. Assign trainable properties of mobile net v2 base model as false.
i. Declare learning rate (0.001), epochs(20) and batch size(12).
j. Use Adam as optimizer and binary cross entropy as loss.
k. Fit the model giving following parameters:

   i. Use image augmentation flow to artificially expand the size of the training dataset. (Use rotation range, zoom range, width_shift_range, etc. as parameters)
   ii. Steps per epoch: Calculated as length of training data divided by batch size.
   iii. Validation data: test data.
   iv. Validation steps: Calculated as length of testing data divided by batch size.
   v. Epochs
l. Save the model.
m. Test the model with the test splitted data and create classification reports, plots as needed.

3.5 Testing the model

Steps:

a. Read photo path and weights path for Open CVV DNN and create a model using the same parameters.
b. Load the saved trained model.
c. Read the image to be tested.
d. Use blob from image to preprocess image and convert image to the format that has been used by DNN while training the model.
e. Detect the faces from the model and save the detections.
f. Iterate over the detections and do the following steps:

   i. Find the confidence (confidence of detection of faces).
   ii. Store the start and end coordinates (both x and y coordinates).
   iii. Convert the image to RGB channel and resize, convert to array, and preprocess the image.
   iv. Predict the probability of with and without mask using the model loaded initially (our trained model).
   v. Draw a rectangle surrounding the face and put text as "Mask" in green color if mask probability is greater than without mask probability and put text as "Without mask" in red color if without mask probability is greater than mask probability.

g. For mask detection on video stream, start video stream and Repeat step 5 and 6 for each frame.

3.6 Testing the model

To implement our model in the real world, live video is captured, frame by frame. This video feed is then fed to our algorithm (model). It can proceed only if a face has been detected. After detecting a face, required pre-processing tasks are done. Then, the model converts the pre-processed into an array-form and does the required further processing tasks using MobileNetv2.

Then, for the given frame, it classifies if the subject is not wearing a mask or wearing it incorrectly or wearing it properly as the model requires them to. The classification result is shown by putting a colored rectangle around the subject's face with the probability of that classification in the same color. It uses the following coloring scheme:

a. No mask: Red
b. Mask weared incorrectly: Blue
c. Mask: Green

This process is repeated for each video frame. Sample images classified by the model are shown below:

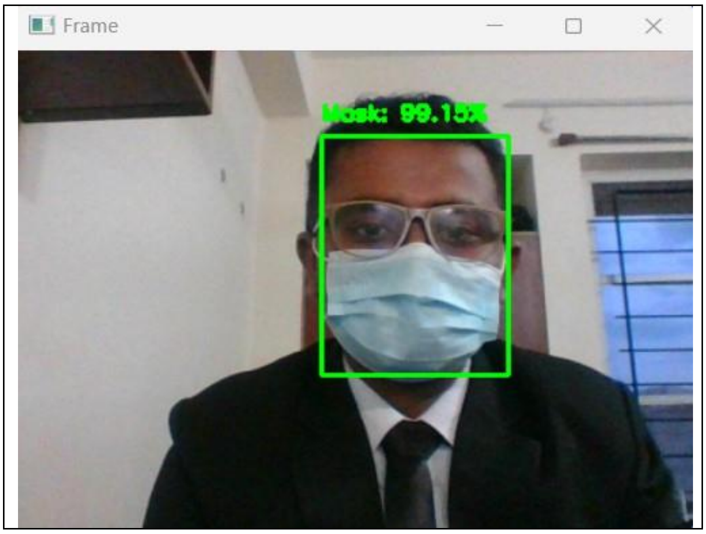

Fig 5. Detection with mask

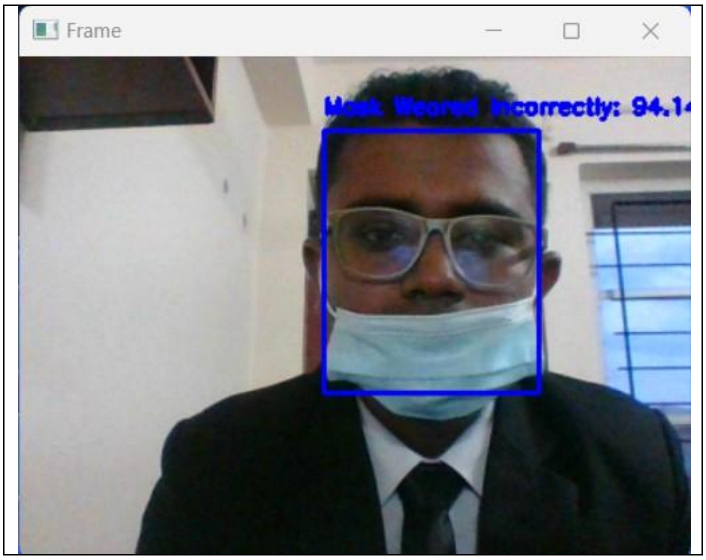

Fig 6. Mask weared incorrectly

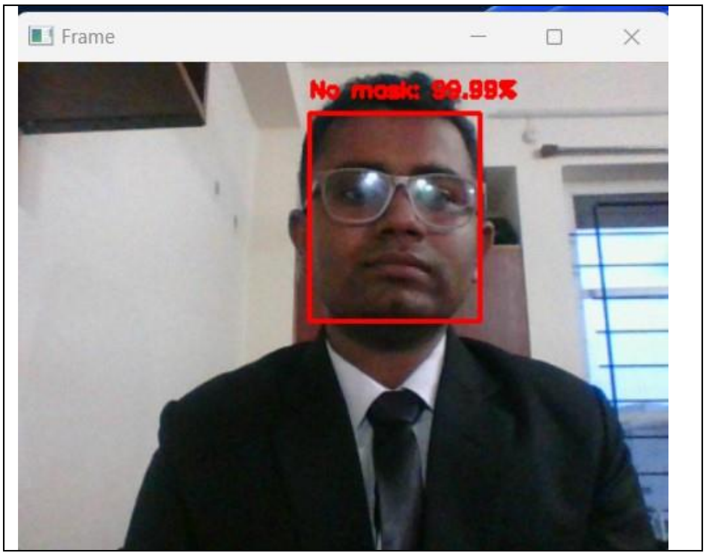

Fig 7. Without mask

5. Result and Discussion

Due to heavy computational costs caused by classical deep learning and machine learning models, we are pushed to implement the optimization technique for the given problem with respect to the available hardware and reliability.

The result for 20 iterations in checking the loss, accuracy, value loss, value accuracy when training the model is shown in Table I.

Table -I: Iteration of checking the loss and accuracy

Epoch	Loss	Accuracy	Val_loss	Val_acc
1/20	0.3497	0.8643	0.184	0.9238
2/20	0.2508	0.9056	0.1423	0.9471
3/20	0.2321	0.9120	0.1772	0.9310
4/20	0.2112	0.9247	0.1306	0.9477
5/20	0.1889	0.9283	0.1977	0.9176
6/20	0.1755	0.9324	0.1611	0.9327
7/20	0.1739	0.9336	0.1452	0.9416
8/20	0.1676	0.9385	0.2055	0.9160
9/20	0.1601	0.9397	0.1417	0.9477
10/20	0.1565	0.9446	0.1323	0.9471
11/20	0.1504	0.9454	0.1409	0.9516
12/20	0.1540	0.9438	0.1272	0.9505
13/20	0.1430	0.9445	0.1321	0.9499
14/20	0.1509	0.9478	0.1132	0.9605
15/20	0.1425	0.9468	0.1658	0.9310
16/20	0.1354	0.9489	0.1778	0.9277
17/20	0.1312	0.9546	0.1414	0.9455
18/20	0.1226	0.9576	0.0969	0.9622
19/20	0.1255	0.9548	0.1044	0.9549
20/20	0.1221	0.9538	0.1162	0.9555

From Table 1, we can interpret that the accuracy is increasing at the start of the second epoch, and loss is seen decreasing after it. At the 20th epoch the loss reduced to 0.1221. Similarly, accuracy reached 0.95. The process of training into the Deep neural network was much faster than the expectation. The table is plotted in the graph shown for better understanding in Figure 2.

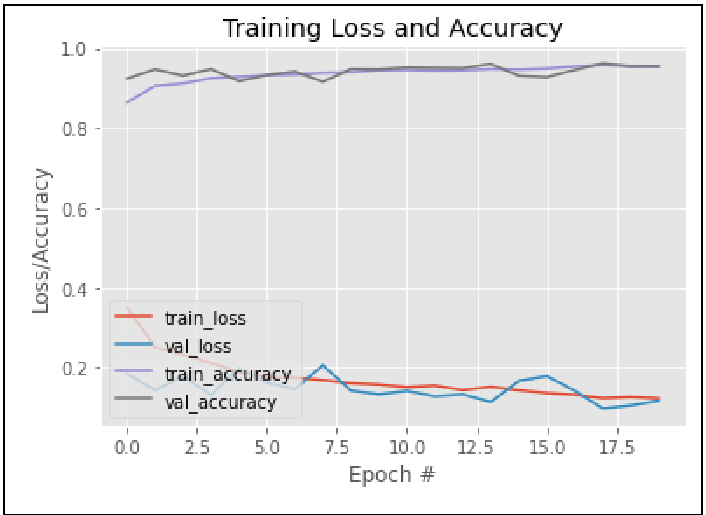

Fig. 8. Graph of Training Loss and Accuracy vs Epoch

Table II. Model Evaluation

	Precision	**Recall**	**F1-score**	**Support**
With mask	0.92	0.98	0.95	599
Without mask	0.97	0.96	0.97	599
Mask weared incorrectly	0.98	0.92	0.95	599
Accuracy			0.96	1797
Macro average	0.96	0.96	0.96	1797
Weighted average	0.96	0.96	0.96	1797

Our proposed model was able to classify the mask-wearing image with faces. The accuracy provided by the model in such a scenario was near about perfect as the model detected all the faces that had masks on them accurately along with the accuracy level and it also accurately detected the faces that weren't wearing masks. Though the model doesn't give an impression about people with an incorrectly worn mask, the results at the current level are satisfactory.

6. Future works

We faced some issues like varying angles and lack of clarity of frame. Constantly moving faces in the video steam input produces several still frames and this makes it more difficult to detect masks with high accuracy. However, this improves over time and following the trajectories of several frames of the video helps to create a better decision.

In certain edge scenarios where people are wearing the mask incorrectly, the model may get confused and give out a classification probability of 40-60%. This is an erratic situation for the model and thus the model cannot accurately classify between "mask weared incorrectly" and "no mask". We aim to improve these shortcomings in future editions of the work.

7. Conclusion

Our paper presents a model using deep learning for face mask detection for the prevention of COVID-19. Our system classifies if a person is wearing a face mask properly, wearing incorrectly or not wearing at all. The model we built has higher accuracy because of heavy training with almost 9,000 labeled facial images. The final product shows a colored box with accuracy percentage in the upper part, which includes the head of the person. This research work can be crucial for curbing the spread of COVID-19 in society. Also, our model hence can be integrated in several applications for successful scalability and real-world impact.

8. REFERENCES

[1] Khan, M., Chakraborty, S., Astya, R. and Khepra, S., 2019, October. Face Detection and Recognition Using OpenCV. In 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) (pp. 116-119). IEEE.

[2] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).

[3] A. A. Hussain, O. Bouachir, F. Al-Turjman and M. Aloqaily, "Notice of Retraction: AI Techniques for COVID-19," in IEEE Access, vol. 8, pp. 128776-128795, 2020, doi: 10.1109/ACCESS.2020.3007939.

[4] Nguyen, D.C., Ding, M., Pathirana, P.N. and Seneviratne, A., 2021. Blockchain and AI-based solutions to combat coronavirus (COVID-19)-like epidemics: A survey. IEEE Access, 9, pp.95730-95753.

[5] Deore, G., Bodhula, R., Udpikar, V. and More, V., 2016, June. Study of masked face detection approach in video analytics. In 2016 Conference on Advances in Signal Processing (CASP) (pp. 196-200). IEEE.

[6] Bhuiyan, M.R., Khushbu, S.A. and Islam, M.S., 2020, July. A deep learning based assistive system to classify COVID-19 face mask for human safety with YOLOv3. In 2020 11th International Conference on Computing,

Communication and Networking Technologies (ICCCNT) (pp. 1-5). IEEE.

[7] Venkateswarlu, I.B., Kakarla, J. and Prakash, S., 2020, December. Face mask detection using mobilenet and global pooling block. In 2020 IEEE 4th Conference on Information & Communication Technology (CICT) (pp. 1-5). IEEE.

[8] Das, A., Ansari, M.W. and Basak, R., 2020, December. Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV. In 2020 IEEE 17th India Council International Conference (INDICON) (pp. 1-5). IEEE.

[9] Susanto, S., Putra, F.A., Analia, R. and Suciningtyas, I.K.L.N., 2020, October. The Face Mask Detection For Preventing the Spread of COVID-19 at Politeknik Negeri Batam. In 2020 3rd International Conference on Applied Engineering (ICAE) (pp. 1-5). IEEE.

[10] Sanjaya, S.A. and Rakhmawan, S.A., 2020, October. Face Mask Detection Using MobileNetV2 in The Era of COVID-19 Pandemic. In 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI) (pp. 1-5). IEEE.

BIOGRAPHIES

Mr. Rajaram Yadav
UG Student VIT, Vellore,
Tamil Nadu

Mr. Safal Gautam
UG Student VIT, Vellore,
Tamil Nadu

Mr. Rahul Ratna Das
UG Student VIT, Vellore,
Tamil Nadu

Ms Prarthana Shiwakoti
UG Student VIT, Vellore,
Tamil Nadu