Detection of the Presence of Face Mask in Real-Time

Rishab K S
Computer Science & Engg.
PES University
Bengaluru , India
rishab12360@gmail.com

Rishab Kashyap B S
Computer Science & Engg.
PES University
Bengaluru , India
rishab182001@gmail.com

Pranav M R
Computer Science & Engg.
PES University
Bengaluru , India
pranav804@gmail.com

Adhithya Sundar
Computer Science & Engg.
PES University
Bengaluru , India
adhithya.sundar007@gmail.com

***Abstract*—The main aim of this project is to develop a deep learning model to detect whether a face mask is worn or not, in real-time. In this project, first a model is developed and trained on images contained human faces with and without mask. Then input frames from a video camera is fed into the trained model which produces the probability of a face mask being worn or not in real-time.**

***Index Terms*—MobileNet, computer vision, image processing, face detection**

I. Introduction

Since the infectious coronavirus disease (COVID-19) was first reported in Wuhan, it has become a public health problem in China and even around the world. COVID-19 pandemic has rapidly affected our day-to-day life disrupting the world trade and movements. It is an emerging respiratory infectious disease caused by Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2). All over the world, especially in the third wave, COVID-19 has been a significant healthcare challenge. Many shutdowns in different industries have been caused by this pandemic. In addition, many sectors such as maintenance projects and infrastructure construction have not been suspended owing to their significant effect on people's routine life. Individuals with COVID-19 have had a wide scope of symptoms reported – going from mellow manifestations to serious illness. Respiratory problems like shortness of breath or difficulty in breathing is one of them. Elder people having lung disease can possess serious complications from COVID-19 illness as they appear to be at higher risk. Some common human coronaviruses that infect public around the world are 229E, HKU1, OC43, and NL63. Before debilitating individuals, viruses like 2019-nCoV, SARS-CoV, and MERS-CoV infect animals and evolve to human coronaviruses. Persons having respiratory problems can expose anyone (who is in close contact with them) to infective beads. Surroundings of a tainted individual can cause contact transmission as droplets carrying virus may withal arrive on his adjacent surfaces

Therefore, to prevent rapid COVID-19 infection, many solutions, such as confinement and lockdowns, are suggested by the majority of the world's governments. Wearing a protective face mask has become a new normal. Wearing a face mask prevents the transmission of droplets in the air and maintaining an appropriate physical distance between people, and reducing close contact with each other can still be beneficial in combating this pandemic.

Therefore, this project focuses on implementing a Face Mask Detection model using pre-trained models such as MobileNet and ResNet. After developing the model we test our model in real-time using a webcam. As a surveillance task performer, it can also detect a face along with a mask in motion. This can be used an application in an office building or at airport terminal/gates. Therefore, face mask detection has become a crucial task to help global society.

II. Literature Survey

A. Face Mask Recognition System with YOLOV5 Based on Image Recognition

The authors have divided the whole system into four parts: facial mask image enhancement, facial mask image segmentation, facial mask image recognition, and interface interaction. Facial mask image enhancement is used to improve the resolution of the mask worn for easy detection. Face mask image segmentation is used to extract mask information. The facial mask recognition part classifies the extracted mask information.

- Facial mask image enhancement- This is the first stage in which enhancement of part of face is done, Smoothing is done to reduce image noise and extract useful information, this is done by using common filtering methods such as mean filtering, median filtering and gaussian filtering.
- Segmentation of Facial Mask Images Face segmentation is performed using the C-V model.
- Face Mask Image Recognition YOLOV5 (You Only Look Once: Unified, real-time Object Detection) model is used in this phase which can quickly and accurately identify whether a mask is worn or not.
- Interface Interaction: The authors design a man-machine interface for the recognition of face to wear masks, when people enter the store into the door of the mall will be prompted to "face look at the camera," since then, the computer whether to wear masks to customers, which can identify if the customer does not wear masks gate won't open.

In this paper, YOLOV5 is applied to identify whether a face mask is worn so that the gate at the entrance of the shopping mall can be opened and closed successfully. However, this kind of recognition is only for mask recognition, if in some special circumstances, the customer covers part of the mask with his hand, it will not be recognized successfully.

B. Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV

The proposed method consists of a cascade classifier and a pre-trained CNN which contains two 2D convolution layers connected to layers of dense neurons.Two datasets have been used for experimenting the current method. Dataset 1 [5] consists of 1376 images in which 690 images with people wearing face masks and the rest 686 images with people who do not wear face masks.Dataset 2 from Kaggle [6] consists of 853 images and its countenances are clarified either with a mask or without a mask. In this dataset, some face collections are head turn, tilt and slant with multiple faces in the frame and different types of masks having different colors as well.

The model is trained, validated and tested upon two datasets. Corresponding to dataset 1, the method attains accuracy up to 95.77. Dataset 2 is more versatile than dataset 1 as it has multiple faces in the frame and different types of masks having different colors as well. Therefore, the model attains an accuracy of 94.5 on dataset 2.

III. Technologies Used

A. MobileNetV2 - as Mask Detector

As an upgrade from the previous version MobileNetV1, Depthwise Separable Convolution is introduced in this version which dramatically reduce the complexity cost and model size of the network, which is suitable to Mobile devices, or any devices with low computational power. In MobileNetV2, a better module is introduced with inverted residual structure and non-linearities in narrow layers are removed.
In MobileNetV2, there are two types of blocks (Figure 1). One is residual block with stride of 1. Another one is block with stride of 2 for downsizing.There are 3 layers for both types of blocks.

- The first layer is 1×1 convolution with ReLU6.
- The second layer is the depthwise convolution.
- The third layer is another 1×1 convolution but without any non-linearity. It is claimed that if ReLU is used again, the deep networks only have the power of a linear classifier on the non-zero volume part of the output domain.

B. ResNet - as Face Detector

When we increase the number of layers in a neural network, there is a common problem in deep learning associated with that called Vanishing/Exploding gradient. This causes the gradient to become 0 or too large. Thus when we increases number of layers, the training and test error rate also increases. ResNet, which was proposed in 2015 by researchers at Microsoft Research introduced a new architecture called Residual Network.

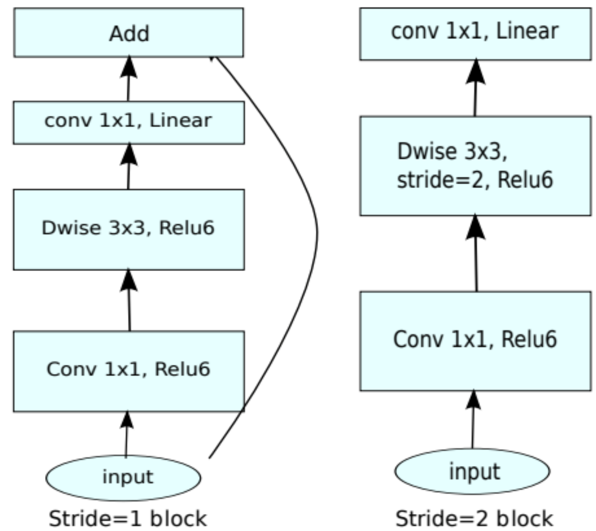

Fig. 1: MobileNetV2

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Fig. 2: MobileNetV2 Overall Architecture
where t=expansion factor, c=number of output channels, n=repeating number, s=stride

1) Residual Block: In order to solve the problem of the vanishing/exploding gradient, this architecture introduced the concept called Residual Network. This network uses a technique called skip connections. The skip connection skips training from a few layers and connects directly to the output. The approach behind this network is instead of layers learn the underlying mapping, we allow network fit the residual mapping. The advantage of adding this type of skip connection is because if any layer hurt the performance of architecture then it will be skipped by regularization. So, this results in training very deep neural network without the problems caused by vanishing/exploding gradient.

2) Network Architecture: This network uses a 34-layer plain network architecture inspired by VGG-19 in which then the shortcut connection is added. These shortcut connections then convert the architecture into residual network.

Convolutional stage	Output size	Layer
Conv1	200×150	7×7, 64, stride 2
Conv2_x	100×75	3×3 max pool, stride 2 $\begin{bmatrix} 3\times 3, 64 \\ 3\times 3, 64 \end{bmatrix}$
Conv3_x	50×38	$\begin{bmatrix} 3\times 3, 128 \\ 3\times 3, 128 \end{bmatrix}$
Conv4_x	25×19	$\begin{bmatrix} 3\times 3, 256 \\ 3\times 3, 256 \end{bmatrix}$
Conv5_x	13×10	$\begin{bmatrix} 3\times 3, 512 \\ 3\times 3, 512 \end{bmatrix}$

Fig. 3: ResNet10 Architecture

IV. Approach

Our project is composed of 2 parts - developing a mask detector model and evaluating the developed into in real-time.

A. Data Pre-Processing

The dataset consisting of 7553 images is converted into NumPy arrays using img_to_array(image) and preprocesss_input(image) functions imported from keras.prerocessing and keras.applications.mobilenet_v2 respectively.
The labels (”with_mask” and ”without_mask”) is also converted to NumPy arrays using LabelBinarizer.fit_transform(labels) and to_categorical(labels) imported from sklearn.preprocessing and keras.utils respectively.
We use ImageDataGenerator() imorted from keras.preprocessing.image to produce new images with slightly altered features (rotation, zoom, width, height) which is appended to our dataset.
The dataset is now split into training and testing datasets using train_test_split() imported from sklearn.model_selection. 75% of the dataset is randomly split into training and the remaining 25% is split into testing.

B. Mask Detector

The full model is divided into 2 parts - Convolutional Layers and Fully Connected Layers.
For the Convolutional part, we implement the MobileNetV2 model which is pre-trained on ”ImageNet” dataset.
For the Fully Connected model, we stack the AveragePooling2D, Flatten, Dense, Dropout (0.5) and Dense layers on top of each other. The activation function used in the hidden layers is ”relu”, since among all activation functions ”relu” provides the best results for image classification. The activation function for the output layer used is ”softmax” as it provides the confidence or probability (proportion) of face mask being detected or not.The model is compiled using ”Adam” as optimizer and ”binary_crossentropy” as the loss function since we only have 2 output labels - with_mask and without_mask.

C. Real-Time Mask Detection

Using OpenCV’s VideoStream function, we take the video input from the webcam. Each input frame is read and converted to a blob using blobFromImage() method imported from cv2.dnn.
For detecting the number of faces, we employ the ResNet10 model which produces the number of faces detected and the location of each face in the input frame.
If the number of faces is greater than 0, we call the Mask-Detector model to provide the probability (confidence) of a face mask being worn or not for every face detected.
For every face detected, we provide a bounding box over the face with colors - green if mask is detected or red if a mask is not detected. The labels over the bounding box include - Mask or No Mask and the confidence with which we can detect a face mask.

V. Dataset

The dataset used can be found on Kaggle under the title Face Mask Detection Dataset [1] from Omkar Gurav.
The Data set consists of 7553 RGB images in 2 folders as with_mask and without_mask. Images are named as label ”with_mask” and ”without_mask”. Images of faces with mask are 3725 and images of faces without mask are 3828.

VI. Experimental Results

A. Evaluation Metric - Accuracy

For the training dataset, we have obtained an accuracy of 98% and for the testing dataset, we have achieved an accuracy of 90%.

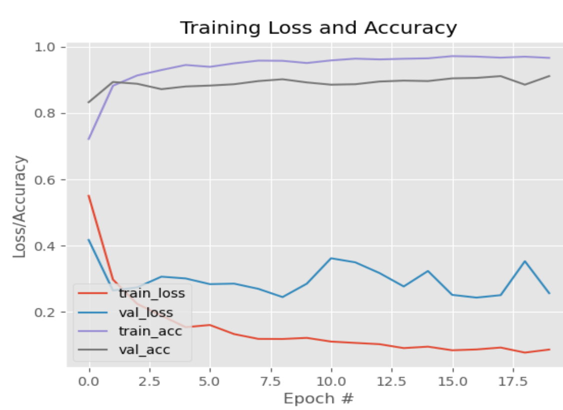

Fig. 4: Training and validation accuracy

B. Real-Time Evaluation

The real time application of our model works as intended and produces accurate results.

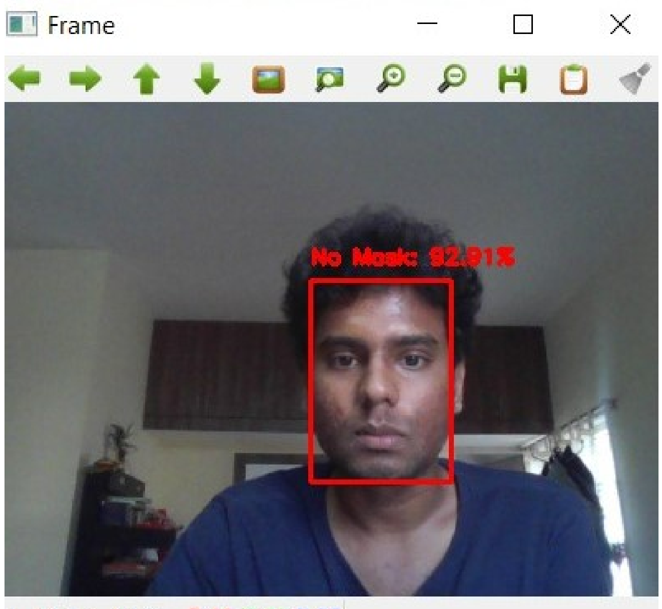

Fig. 5: Real-Time application of Mask-Detector (No Face Mask)

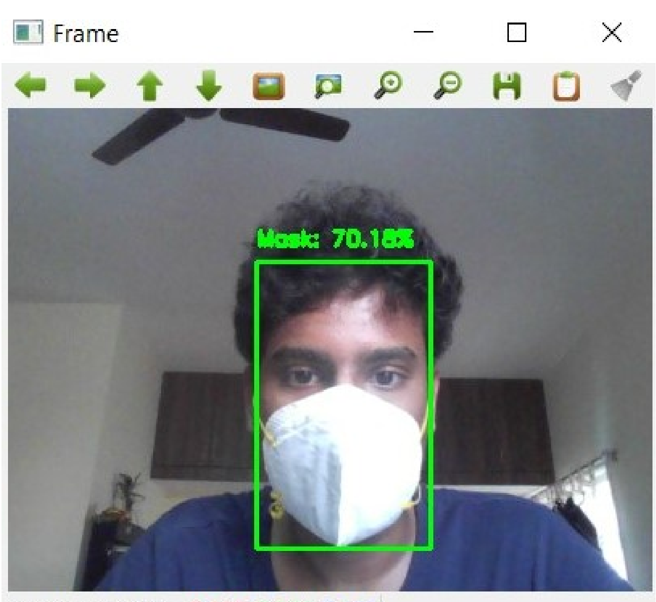

Fig. 6: Real-Time application of Mask-Detector (With Face Mask)

VII. Conclusion

In this project we have implemented a model which can be used as an application to detect the presence of a face mask in real time. This model has achieved an accuracy of over 90% on the testing dataset. In the real-time application, the model produces acceptable results. In the case of a mask being worn but the nose is not covered, produces ambiguous results. The probability flickers at around 50% and the model finds it hard to make a decision with high confidence. As an improvement, we can use other pre-trained models (ResNext, GoogleNet) which provide better accuracies.
With the global impact of COVID-19, we believe that our project can be incorporated in institutions with huge workforce (offices, airport terminal, railway terminals, etc). People who are not wearing face masks can be tagged almost immediately and informed to respective authorities.

A. Member Contributions

- **Rishab K S** : Real-time mask detection, model building and report.
- **Rishab Kashyap B S**: Data Pre-processing and presentation.
- **Pranav M R**: Model building, report and evaluation.
- **Adhithya Sundar**: Literature survey and presentation.

All the members referred to several research papers and contributed towards literature survey.

References

[1] https://www.kaggle.com/datasets/omkargurav/face-mask-dataset
[2] ”Coronavirus disease 2019 (covid-19): situation report 205”, 2020.
[3] https://ieeexplore.ieee.org/abstract/document/9345042/authorsauthors
[4] G. Yang et al., ”Face Mask Recognition System with YOLOV5 Based on Image Recognition,” 2020 IEEE 6th International Conference on Computer and Communications (ICCC), 2020, pp. 1398-1404, doi: 10.1109/ICCC51575.2020.9345042.
[5] ”prajnasb/observations”, 2020, [online] Available: https://github.com/prajnasb/observations/tree/master/experiements/data.
[6] ”Face Mask Detection”, 2020, [online] Available: Kaggle.com.