Facemask Detection Using MobileNetV3

by Rudrasish Mishra

Submission date: 13-Apr-2022 09:56AM (UTC-0500)

Submission ID: 1809585991

File name: MINOR_REPORT_Facemask_Detection_Using_MobileNetV3.docx (2.63M)

Word count: 5862

Character count: 30711

ABSTRACT

The outspread of the corona virus types has caused pandemic across the globe.The virus is known for severe acute respiratory issues. The ailment it causes is called coronavirus disease 2019 or (COVID-19) affecting our day-to-day life disrupting world trade and movements. Wearing a mask is one of the effective methods to prevent us from being affected by the vicious COVID virus. Due to carelessness and unawareness of the local public towards this issue a facemask detection system can be used by the authorities to keep a check on them and create awareness about the same.

This project titled Face Mask Detection has been developed using a machine learning technique known as transfer learning and by using image classification method of MobileNetV3small .Steps followed while building the model are data collection, data preprocessing , data splitting , model-rectification ,model testing and model implementation. Model is capable of predicting if individuals are wearing or not wearing masks at an accuracy of 99.81 percent. We also explored using other image classification models like VGG16 with an accuracy of 96 percent, MobileNetV2 with an accuracy of 97.21 percent and ResNet101 with an accuracy of 97.79 percent to detect masks. We have implemented the Transfer learning method that uses already acquired knowledge by avoiding overfitting of the models.

INTRODUCTION

This project titled Face Mask Detection has been developed using a machine learning technique known as transfer learning and by using image classification method of MobileNetV3small .Steps followed while building the model are data collection, data preprocessing , data splitting , model-rectification ,model testing and model implementation. Model is capable of predicting if individuals are wearing or not wearing masks at an accuracy of 99.81 percent. We also explored using other image classification models like VGG16 with an accuracy of 96 percent, MobileNetV2 with an accuracy of 97.21 percent and ResNet101 with an accuracy of 97.79 percent to detect masks. We have implemented the Transfer learning method that uses already acquired knowledge by avoiding overfitting of the models.

COVID-19 as a pandemic resulted in consequential loss of human life across the globe and it has been a strenuous challenge to public health , human life and financial prosperity to a great extent.The financial and public turmoil caused by the pandemic is cataclysmic.Countless ventures across the world are facing threat of extinction . Kimberly Chriscaden of WHO [1] presented a survey that nearly fifty percent of the world's workforce are at risk of losing their means of livelihood.Sustained efforts have been made to diminish the chances of being affected by this fatal virus by developing vaccines and manufacturing generic medicines but still the issue is not completely healed and all efforts went in vain .The adverse effect of this pandemic has affected all financial enterprises, education sector, tourism sector, employment rate, entertainment industry, food supply,security and other industries. With the new upcoming variants like delta and omicron of the fatal virus and constant ignorance and disregard of the public towards this grave issue had led to 488,163,510 confirmed affected cases and 6,142,579 million deaths worldwide.Many precautionary measures have been taken by the government and NG organisations to restrict the outspread of this mortal virus like providing primary amenities to underprivileged, donating medical equipments and protective gear and voluntarily assisting in awareness campaigns.WHO [2] specified that washing hands, keeping a safe distance of 1 metre, putting on a protection mask, refraining from touching eyes, nose, and mouth are the basics, where wearing a mask is the easiest one . Stella [3] further added that according to a study by scientists at Monash University and the University of Edinburgh based on 30 studies across the world and found that wearing a mask can significantly reduce the chances of incidence of Covid-19 virus by 53 percent. Gandhi [4] further analysed that wearing a mask reduces the amount of virus the wearing individual might receive, leading to infections that are milder or even non symptomatic. A substantial viral dose results in a more hostile inflammatory response, if more individuals get mild cases, that might help to enhance immunity at the population level without increasing risks of death or severe ailment. Hindustan Times [5] according to a study by the union public health ministry of India it revealed that more than 50 percent of the Indian population doesn't wear a mask and out of the rest wearing inhabitants only 14 percent people wear the mask properly. Hence the administration needs a suitable solution to be able to properly supervise the implementation of law and order.

This project introduces facemask detection in three phases, training, testing and finally the deployment of the model.In the training phase trained dataset using two classes with-mask and without-mask using transfer learning, tensorflow and OpenCV with the help of an image classification method MobileNetV3Small which is the 3rd version of the MobileNet architecture developed by google promoting image analysis in many trending mobile apps. In the testing phase the developed model is tested using an image to examine the accuracy. Finally the model is mounted to detect masks on the human face using embedded systems that could significantly decrease the cost of production of such face mask detection techniques . Andrew Howard et al. [6] presented that MobileNetV3-Large is 3.2 percent more accurate as compared to imagenet reducing the latency by 20 percent. They further compared them with their predecessors and concluded that MobilenetV3Small is 6.6 percent and MobilenetV3Large is 25 percent faster than their predecessor MobileNetV2. Jesus Rordriguez [7] added that MobileNetV3 has shown significant advancement in case object detection tasks over older architectures and it operates at 25 percent lower latency and with identical accuracy. Choosing a suitable activation function in DNN has a huge impact on training as well as performance . Hence we chose this architecture for our project. An activation function is necessary to introduce nonlinearity between input and output. They primarily determine when to activate a neuron and when not to. If no activation function is applied then there will be a linear association between input and the output and it will be unfit to resolve complex problems. Random Nerd [8] analysed that although the Rectified Linear Unit activation function has many alternatives but till now none of them have managed to replace it due to their inconsistent gains. The Google brain team proposed a new activation function swish which has experimentally proved not only to match but also outperform rectified linear unit activation function on the DNN models applied across a wide range of challenging spheres like image classification and machine translation. We used the swish activation function due to its non monotonic attribute and as it outperforms ReLu for all values of batchsize. Vandit [9] surveyed that sigmoid function in terms of computation is quite immoderate and as we heed a lot about computational outlay hence the makers modified it with hard-swish (h-swish) activation function.Nieto-Rodríguez et al. [10], proposed a system which triggers an alarm in the operating room when any of the healthcare personnels in the institution are not wearing face masks. The system involves two layer detectors, running for faces and for masks in the testing phase, by utilising images from the dataset, around 95 percent of true-positive and near 5 percent of false-positive figures are achieved. The detectors implement colour processing in order to intensify the true positives to false positives ratio. The importance of facemasks is quite significant as well as their correct use. Aiello et al. [11] examined the efficiency of wearing facemasks and incidence of the virus and they came to a conclusion that it reduces the chances of being affected by 43 percent. This leads to an increase in the use of non medicinal products that reduces the transmission of the virus. For this reason it is necessary to

inspect and control the use of masks in public spaces , inhabiting regions and operational areas with a large population.
There is no proper structured face mask detection system, hence that increases the challenge for a coherent system to predict face masks on human faces at various locations like, densely populated areas, Enterprises ,residential localities to ensure public safety. The built model can be directly deployed on CCTV supervised cameras to hinder the dissemination of the fatal COVID-19 virus by determining if the individuals are wearing or not wearing a facemask. This system can help the authorities to keep a check on the localities that require more attention as well as maintain law and order and take initiatives in order to prohibit the transmission of the lethal virus.

Chapter 2

Basic Concepts/ Literature Review:

3.1 TENSORFLOW:

Tensorflow is an open source deep learning library and an interface for expressing machine learning algorithms for fast and high performance numerical computing was developed by the Google brain team for research was released in 2019 and development and is currently maintained by Google under the Apache 2.0 open source licence.. It is used in a variety of areas for sentiment analysis, voice recognition,image classification, flaw detection , text summarization , information retrieval and many more. It works perfectly with python 2.7 and python 3.3+. In our project we have used tensorflow in the back end for training purposes as well as for the testing purpose while building our entire transfer learning model. Its flexible architecture helps in its easy deployment in a variety of platforms from mobiles to servers to a variety of other [12] computational platforms (CPUs, GPUs, TPUs) .Due to its comprehensive ,simple and flexible computational core it is used in other scientific domains as well.

3.2 KERAS:

Keras is a high level open source neural networks library developed by François Chollet, a Google engineer that is capable of running on top of TensorFlow .Previously till version 2.3 it was compatible with multiple backends like TensorFlow, Microsoft Toolkit, Theano, and PlaidML but since the upgrade to version 2.4 it only supports Tensorflow.It supports [13] both CNN as well as RNN and the combination of both of them .It was developed to [14] ensure fast experimentation with DNN and it acts as an interface for the tensorflow library as well as an python interface for ANN.All the layers in a CNN model are implemented using keras and it's core data structure consist up of models and layers as well as it helps in compilation of the model.

3.3 NUMPY:

Numpy or numeric python is an open source library in python that supports operation on large multidimensional arrays and matrices, and was created by Travis Oliphant in 2005 integrating features of the competing Numarray into Numeric which is considered to be its predecessor , by incorporating considerable modifications. In this project numpy is used to convert images into an array of height,width,channel format in the preprocessing phase.

3.3 SCI-KIT LEARN:

SciKit Learn also known as sklearn is an open source ML library compatible with python built on top of SciPy and distributed under [15] the 3-Clause BSD licence was developed initially by David Cournapeau as a Google project in 2007 and is currently maintained by a group of 22 volunteers. It attributes a variety of classification ,clustering and regression algorithms and uses numpy for algebra and array operations. It facilitates train-test-split operation that is used to predict performance of various machine learning algorithms.Latest version of sk-learn requires python version 3.7+,Matplotlib version 3.1.2+ and Pandas version 1.0.5+.

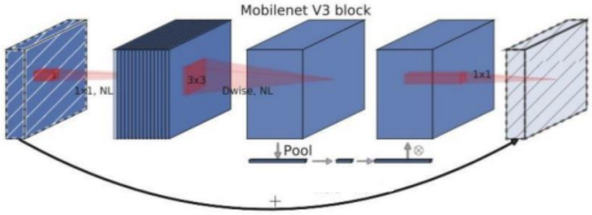
3.4 OPENCV:

Open source computer vision library is a real time optimised computer vision library of programming function mainly focussed on real time computer vision was initially developed by intel corporation as an initiative to advance CPU-intensive applications it is designed to function in multiple computing platforms and is open source free under Apache 2 Licence. It was initially developed and written using [16] C++ and its interfaces are C++, Python, Java and MATLAB.It supports almost all the operating systems.It is used extensively to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects,and many more.Pictures and real time camera (videostream) can be manipulated based on the needs of the user whenever necessary.

3.5 MOBINETV3 SMALL:

MobileNetV3 Small is the third version of the convolutional neural network architecture that supports image analysis developed by Google. It targets low resource expenditure and is 6.6 percent more accurate as compared to a MobileNetV2 model. It is tuned to work efficiently with mobile phone CPUs through a mixture of hardware aware NAS [17] complemented by the NetAdapt algorithm that automatically adapts a pre-trained deep neural network to a mobile platform and then sequentially enhanced through numerous enhancements. We used MobileNetV3Small model as it is faster than its counterparts in predicting if an individual is wearing or not wearing a mask.We removed the last layer of the model replaced it with an average pooling layer ,a system with a network of 128 layers, an activation function 'relu' and a dropout of 0.5 followed by another system with 2 layers that signifies the two classes that we used as labels.

FIGURE-1:Block diagram of MobileNetV3



3.6 ACTIVATION FUNCTION:

An activation function is necessary to introduce nonlinearity between input and output. They primarily determine when to activate a neuron and when not to. If no activation function is applied then there will be a linear association between input and the output and it will be unfit to resolve complex problems.

Neurons receive input signals from the external environment, each input is multiplied by a weight that signifies relative importance and is fed into the neuron.Bias is added to the summation of weighted inputs.

$$\text{SUMMATION OF INPUT} =[\sum(\text{WEIGHT} \times \text{INPUT}) + \text{BIAS}]$$

So the activation function basically performs a computation on the sum of weighted inputs of the neuron and bias and produces a transformed output.

SIGMOID FUNCTION : It has a range from zero to one (0-1), hence it is used in the models where we need to compute probability as output and is famous for its ‘S’ shaped curve.The function is monotonic and differentiable. It can be mathematically represented as

$$\square(n)= 1/1+e-n$$

RECTIFIED LINEAR UNIT (ReLU) : Currently ReLu is the most popular activation function used in almost all CNN and DNN in deep learning. This function is half rectified (from the bottom) and monotonic. Major issue with this function is that it converts all negative inputs to zero immediately and affects by not plotting the negative values accurately.

$$F(n) = MAX(0, n)$$

SWISH FUNCTION : The Google brain team proposed a new activation function swish which has experimentally proved not only to match but also outperform rectified linear unit activation function on the DNN models applied across a wide range of challenging spheres like image classification and machine translation. We used the swish activation function due to its non monotonic attribute and as it outperforms ReLu for all values of batchsize.

$$F(n) = n \ . \square(n)$$

H-SWISH FUNCTION: Swish function is quite immoderate due to presence of sigmoid and as we heed a lot about computational outlay hence the makers modified it with hard-swish (h-swish) activation function by replacing the sigmoid function with a linear analogue.

$$F(n)= n\ RELU\ 6(n+3) / 6$$

Chapter 3

Problem Statement / Requirement Specifications

3.1 Project Planning

Aim of the Project

To predict face masks on the faces of human individuals in public places and help the administration to encourage them to wear a face mask in order to avoid the transmission of the lethalCOVID-19 virus..

Covid-19 pandemic has started since early 2020. The covid-9 pandemic is caused by SARS-CoV-2. It affects the human beings through respiratory at first contact because the virus is known to enter the body through this route. The disease has an immense impact on the health of the individual. Hence some form of preventive practice is always considered better before it catches our body system. The entry of the virus can be prevented by wearing a recommended mask. But it has been seen that a major proportion of the public is not using the mask due to negligence or ignorance. This system will detect the individual not wearing a mask and alert them to wear a mask.

3.2 Project Analysis

3.2.1 Train a model to detect mask

3.2.1.1

In this project we took an existing data set consisting of pictures of people wearing masks and without masks. The data set was divided randomly into two different data sets, one for training and other for testing purposes. The ratio of data was 70% for training and 30% for testing.

3.2.1.2

The database will be expanded by image generator using ten parameters. Mobilenetv3 is used in the model. We removed the last layer of the model and replaced it with an average pooling layer ,a system with a network of 1028 layers, an activation function ‘relu’ and a dropout of 0.5 followed by another system with 2 layers that signifies the two classes that we used as labels.

3.2.1.3

Finally we used a batch size of 12 at a learning rate of 0.0001 and epoch value of 20. We got an accuracy of 99.81%. The model is generated and stored in the file.

3.2.2 Mask detection camera

3.2.2.1

In this Python file we created a function. This function can detect faces from the image and applies the classifier to the region of interest (ROI) for detecting face masks.

This function has three parameters:

frame: A frame from our video stream.

faceNet: The model used to detect where in the image faces are

maskNet: Our COVID-19 face mask classifier model

3.2.2.2

The program forms a blob, detects faces, and initialises lists, two of which the function is set to return. The lists include our faces(region of interest), locs (the face locations), and preds (mask/no mask predictions).

3.2.2.3

In the process, the program filters out weak points and draws out bounding boxes.The program ensures that bounding box coordinates do not go outside the frame of the image.

3.2.2.4

Now we initialise

1. Face detector using cv2.dnn.readNet().
2. Face mask detector using the developed model.
3. Webcam stream using VideoStream(src=0).start()

3.2.2.5

Then the looping starts over video frames. Then inside the loop we pick out one frame and resize it. On that we put our model to detect whether the person is wearing a mask or not.

3.2.2.6

Now we created a loop in which we take two parameters .

One of them is to predict the mask and the other is to detect the face and create a box of appropriate colour around the face. If a person is detected with a mask the box outline will turn green otherwise it will turn red.

3.2.2.7

After the frame from video streaming is displayed. We can press the “o” key to quit the video stream.

3.3 System Design

3.3.1 Design Constraints

Hardware used:

- o PC (Laptop, Desktop)
- o 2. External Camera (Optional)

Software used:

o Anaconda3 o Jupyter notebook

o Python

o Python Library

1.TensorFlow. 2. Keras l3. Sklearn

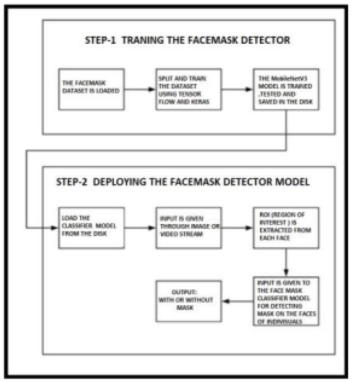
4. Imutils 5. Matplotlib 6. NumPy 7. Cv2

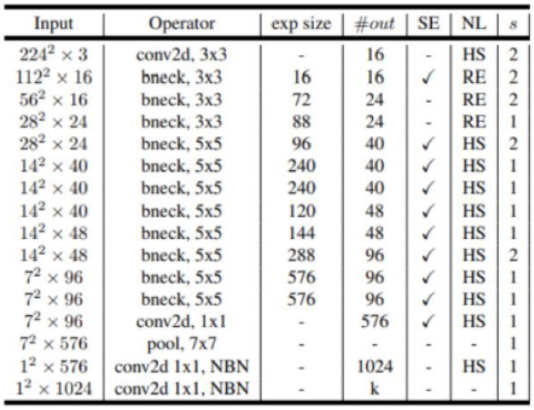
o Models:

1. Mobilenetv3 2. Mobilenetv2

3. ResNet101 4. VGG16

3.3.2 System Architecture (UML) / Block Diagram


STEP-1 TRANING THE FACEMASK DETECTOR
THE FACEMASK DATASET IS LOADED
SPLIT AND TRAIN THE DATASET USING TENSOR FLOW AND KERAS
THE MobileNetV3 MODEL IS TRAINED ,TESTED AND SAVED IN THE DISK
STEP-2 DEPLOYING THE FACEMASK DETECTOR MODEL
LOAD THE CLASSIFIER MODEL FROM THE DISK
INPUT IS GIVEN THROUGH IMAGE OR VIDEO STREAM
ROI (REGION OF INTEREST) IS EXTRACTED FROM EACH FACE
INPUT IS GIVEN TO THE FACE MASK CLASSIFIER MODEL FOR DETECTING MASK ON THE FACES OF INDIVISUALS
OUTPUT: WITH OR WITHOUT MASK



Input	Operator	exp size	$\#out$	SE	NL	s
$224^2 \times 3$	conv2d, 3x3	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	✓	RE	2
$56^2 \times 16$	bneck, 3x3	72	24	-	RE	2
$28^2 \times 24$	bneck, 3x3	88	24	-	RE	1
$28^2 \times 24$	bneck, 5x5	96	40	✓	HS	2
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	120	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	144	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	288	96	✓	HS	2
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	conv2d, 1x1	-	576	✓	HS	1
$7^2 \times 576$	pool, 7x7	-	-	-	-	1
$1^2 \times 576$	conv2d 1x1, NBN	-	1024	-	HS	1
$1^2 \times 1024$	conv2d 1x1, NBN	-	k	-	-	1

Figure-2:MobilenetV3Small architecture specification.

CHAPTER-4

Implementation

4.1 Methodology OR Proposal

In this project we used tensorflow, keras, opencv, numpy, matplotlib and the image classification method of MobileNetV3Small. MobilenetV3 is the third version of the convolutional neural network architecture that supports image analysis developed by Google with improved performance and speed than its predecessors. We have tried our best to avoid overfitting and tried to detect masks on human faces. The model can be produced by following various steps : data collecting, pre-processing, splitting the data,building the model,testing the model, and finally deploying the model.

1. DATA COLLECTION:

We used a dataset [10] that we collected from arXiv.org by Cornell University ,it contains 3833 images out of which 1915 pictures are with mask whereas 1918 pictures are without mask.In order to train the model we specified X that are the images and Y that are the levels to differentiate between images as mask and without mask.

2. DATA PREPROCESSING:

The preprocessing phase is the second phase that comes prior to the training phase and testing phase of the data. There are four steps in the preprocessing phase:scaling size of the image , converting the image into an array, and the last is executing LabelBinarizer on labels.

1. Resizing the image is one of the most important preprocessing steps as smaller the size of the image, the better the model will operate. Here, the resizing of an image {image=load_img (a,target_size =(224,224))} is done by making the image into (224 × 224) pixels.
2. In the next step all the images in the datafile are converted into numpy arrays. The image is converted into the array by calling all the images in the dataset using a for loop.
3. In the last step is LabelBinarizer() is implemented on the labels (with mask and without mask) because it is unattainable for any algorithm to operate on labelled data as they requisite all the input and output variables to be converted into numeric.The labelled data will be transformed into a binary form of 0 and 1, so the model can understand and process the data properly.

3. SPLITTING THE DATA:

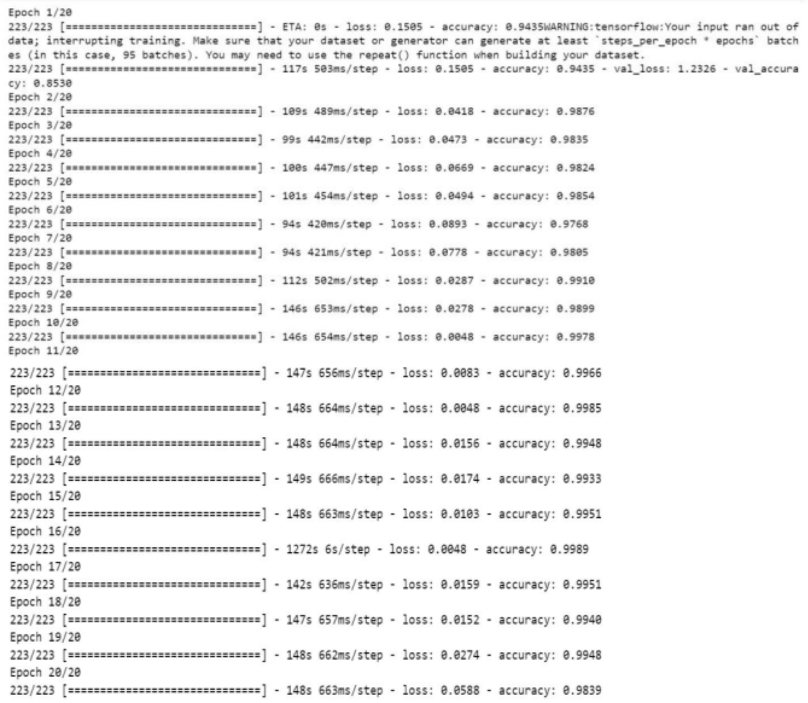
The produced and saved MobileNetV3Small model in this phase is trained with the labelled dataset by splitting it into two categories ,one portion contains 70 % of images and it is used for training while the remaining portion contains the remaining 30 % of images along with its labels and is used for testing the model so that we can know how well our model performs by predicting its accuracy and efficiency. After the model is trained, it can be used for detecting facemask on human faces. Here, x_train means 70 percent images,x_test means 30 percent images, y_train means labels of 70 percent images and y_test means labels of 30 percent images. Imagedatagenator is used to configure random transformations and normalisation operations to be done on the image to increase the database by generating more images by using the images present in the dataset.

4. MODEL BUILDING:

The base model is generated using tensorflow , keras and the image classification CNN model MobileNetV3Small with (input_shape=(224,224,3), weights="imagenet", input_ten sor=None, classes=1000, pooling=None,classifier_activation="swish") and by excluding the last layer of the model (include_top=False). Finally we replaced the final layer with the head model at the top.The head model consist up of an average pooling layer, a flatten layer, a dense layer with a network of 1028 neurons with activation function ‘swish’, a layer with a dropout value of 0.7 and finally followed by a dense layer with network of 2 neurons with ‘swish’ activation function. Finally our model is saved for the future prediction process.

5. MODEL TESTING:

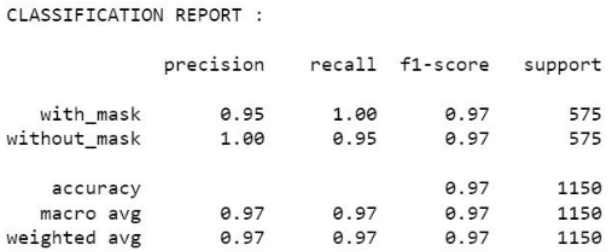
To ensure that the trained model can predict effectively we need to test the model. For testing we took 0.001 as the Learning-rate, as the Batch-size which is the sum total of training examples present in a single batch, and an Epoch value of 20 (one epoch is when the total dataset is passed forward and backward through the entire neural network).predictions on the testing set are done.The result for 20 epoch iterations during the training phase of the model are shown below.The figure manifests the value of loss and accuracy for each individual iteration.



```
Epoch 1/20
223/223 [==============================] - ETA: 0s - loss: 0.1505 - accuracy: 0.9435WARNING:tensorflow:Your input ran out of
data; interrupting training. Make sure that your dataset or generator can generate at least `steps_per_epoch * epochs` batch
es (in this case, 95 batches). You may need to use the repeat() function when building your dataset.
223/223 [==============================] - 117s 503ms/step - loss: 0.1505 - accuracy: 0.9435 - val_loss: 1.2326 - val_accura
cy: 0.8530
Epoch 2/20
223/223 [==============================] - 109s 489ms/step - loss: 0.0418 - accuracy: 0.9876
Epoch 3/20
223/223 [==============================] - 99s 442ms/step - loss: 0.0473 - accuracy: 0.9835
Epoch 4/20
223/223 [==============================] - 100s 447ms/step - loss: 0.0669 - accuracy: 0.9824
Epoch 5/20
223/223 [==============================] - 101s 454ms/step - loss: 0.0494 - accuracy: 0.9854
Epoch 6/20
223/223 [==============================] - 94s 420ms/step - loss: 0.0893 - accuracy: 0.9768
Epoch 7/20
223/223 [==============================] - 94s 421ms/step - loss: 0.0778 - accuracy: 0.9805
Epoch 8/20
223/223 [==============================] - 112s 502ms/step - loss: 0.0287 - accuracy: 0.9910
Epoch 9/20
223/223 [==============================] - 146s 653ms/step - loss: 0.0278 - accuracy: 0.9899
Epoch 10/20
223/223 [==============================] - 146s 654ms/step - loss: 0.0048 - accuracy: 0.9978
Epoch 11/20
223/223 [==============================] - 147s 656ms/step - loss: 0.0083 - accuracy: 0.9966
Epoch 12/20
223/223 [==============================] - 148s 664ms/step - loss: 0.0048 - accuracy: 0.9985
Epoch 13/20
223/223 [==============================] - 148s 664ms/step - loss: 0.0156 - accuracy: 0.9948
Epoch 14/20
223/223 [==============================] - 149s 666ms/step - loss: 0.0174 - accuracy: 0.9933
Epoch 15/20
223/223 [==============================] - 148s 663ms/step - loss: 0.0103 - accuracy: 0.9951
Epoch 16/20
223/223 [==============================] - 1272s 6s/step - loss: 0.0048 - accuracy: 0.9989
Epoch 17/20
223/223 [==============================] - 142s 636ms/step - loss: 0.0159 - accuracy: 0.9951
Epoch 18/20
223/223 [==============================] - 147s 657ms/step - loss: 0.0152 - accuracy: 0.9940
Epoch 19/20
223/223 [==============================] - 148s 662ms/step - loss: 0.0274 - accuracy: 0.9948
Epoch 20/20
223/223 [==============================] - 148s 663ms/step - loss: 0.0588 - accuracy: 0.9839
```

FIGURE-3: Epoch Iteration of checking the loss and accuracy

Once the accuracy of the model becomes constant or it stabilises there is no need for any further iterations; finally we can generate the classification report to evaluate the model.

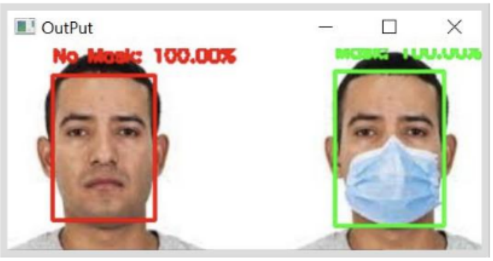


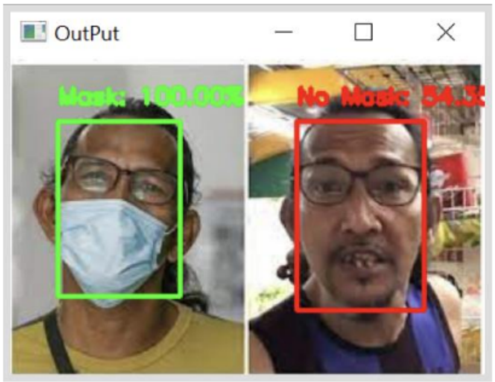
```
CLASSIFICATION REPORT :

              precision    recall  f1-score   support

   with_mask       0.95      1.00      0.97       575
without_mask       1.00      0.95      0.97       575

    accuracy                           0.97      1150
   macro avg       0.97      0.97      0.97      1150
weighted avg       0.97      0.97      0.97      1150
```

FIGURE-4: Model evaluation report or classification report.

6. DEPLOYING THE MODEL:

The trained model is deployed and the caffemodel is used. CaffeModel was created by Caffe and is a machine learning model that is used for image classification or image segmentation and has been trained using Caffe (convolutional architecture for fast feature embedding) and is implemented as well as created using prototxt files.A image containing human faces with or without mask is given as an input and it is dispatched to the face detecting model for the observation and diagnosis of human faces. This is achieved by scaling the figure and by detecting the binary large object in it. Then the binary object is sent to the default facemask detecting model which resizes the image and gives output as the cropped face of a person.In the next phase input data is predicted from the saved model and the output of the previous model is given to the pre-trained MobileNetV3 model which determines whether the individual in the picture is wearing a facemask or not.





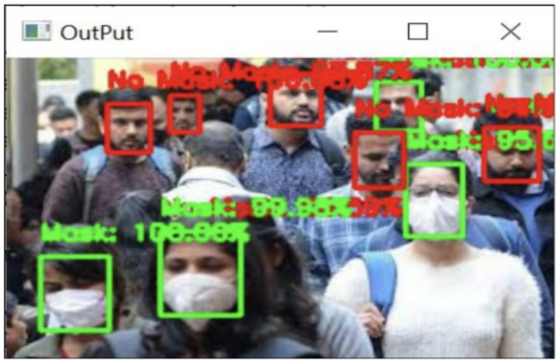
Finally the model is implemented in the video by using OpenCV, caffemodel and our pre-trained and saved MobileNetV3Small model. When an input video is given to the CV2 model,from frame to frame , then the face detection algorithm works. and detects facemask masks on the individuals. In the next phase input data is predicted from the saved model and a bounding circle is drawn around the face of the person that predicts whether the person is wearing a mask or not. . The video frame will also classify whether the person is wearing a mask or not.

4.2 Testing OR Verification Plan

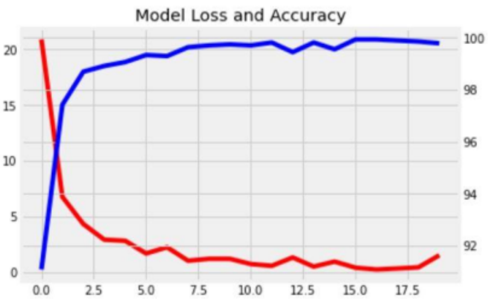
T01	1. IMAGE-1 2. IMAGE-2 (image of an individual masked and unmasked)	Detecting Facemask on the face of the individual	Able to predict and differentiate accurately	Model should be able to differentiate between masked and unmasked individuals in the image.

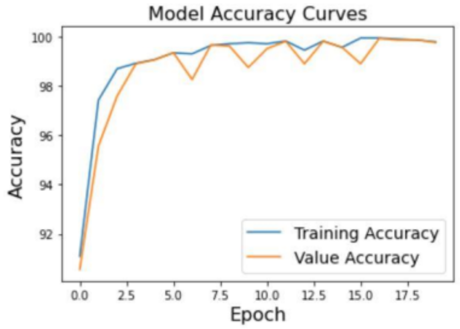
T02	1. 1.IMAGE-3 (image of a group of different masked and unmasked individuals)	Detecting Facemask on the face of the individual	System is able to detect and differentiate between masked and unmasked individuals clearly and accurately	Model should be capable of predicting and differentiating between masked and unmasked individuals
T03	1. 1.VIDEO (Camera Input)	Detecting Facemask on the face of the individual	System was able to take input clearly and is capable of detecting and differentiating between masked and unmasked individuals faultlessly and accurately.	System should be able to take input properly and should be capable of predicting and differentiating between masked and unmasked individuals

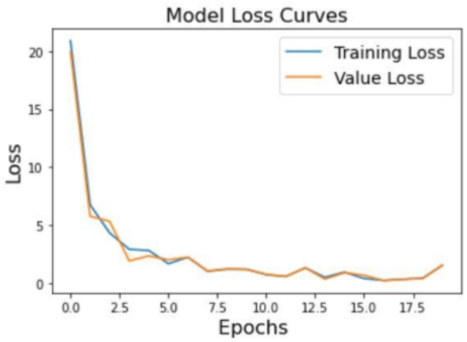
4.3 Result Analysis OR Screenshots:

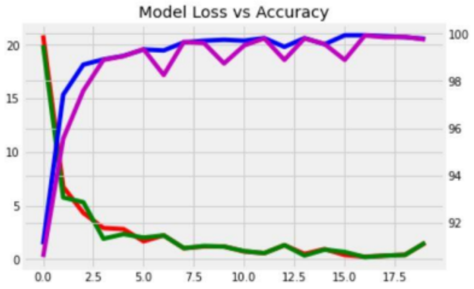


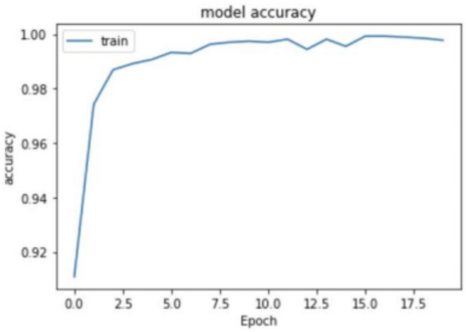
GRAPHS:

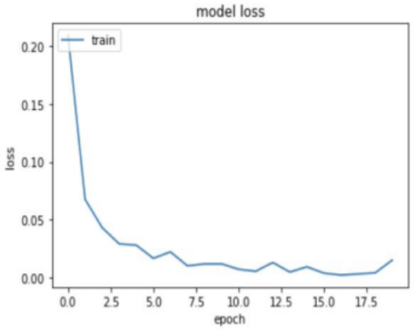


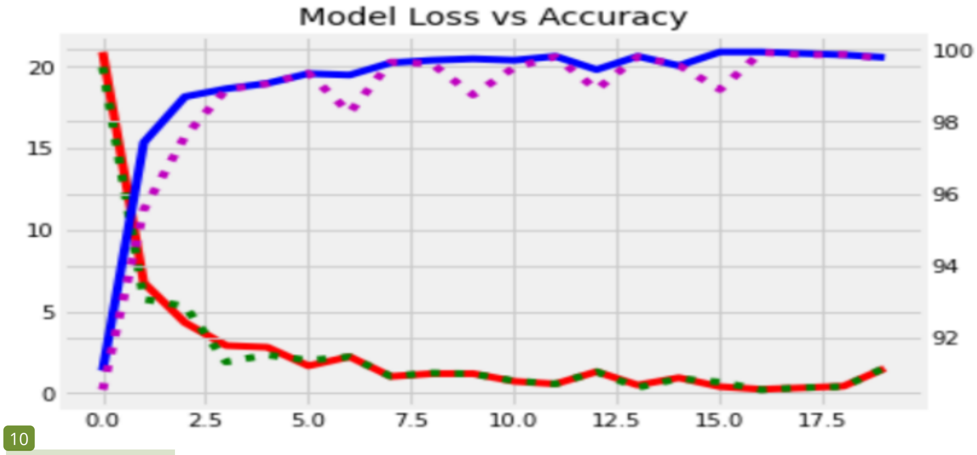












10

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

With the help of this project we tried to explain the facemask prediction system using MObileNet V3Small model. We selected this model after analysing and studying the functioning of various model like ResNet 101, VGG16 and MobileNetV2 but we implemented our project using MobileNetV3Small image classifier because it is 6.6 percent faster and is 3.2 percent more precise in image-net classification by shortening the latency by 20 percent when compared to its its precursor. We used the swish non linearity which has experimentally proved not only to match but also outperform rectified linear unit activation function on the DNN models applied across a wide range of challenging spheres especially in image classification and machine translation. We used the swish activation function due to its non monotonic attribute and as it outperforms ReLu for all values of batchsize. We tried to implement the hard-swish (h-swish) activation function and surveyed that sigmoid function in swish non linearity is quite immoderate in terms of computation as we heed a lot about the computational outlay. H-Swish nonlinearity was created by replacing the sigmoid function with a linear analogue and it is quite similar to swish in terms of performance.We deployed the trained model with the help of convolutional architecture for fast feature embedding (Caffel) which is used for image classification or image segmentation and has been created using prototxt files.A image containing human faces with or without mask is given as an input and it is dispatched to the face detecting model for the observation and diagnosis of human faces. This is achieved by scaling the figure and by detecting the binary large object in it. Then the binary object is sent to the default facemask detecting model which resizes the image and gives output as the cropped face of a person.Finally the model is implemented using the OpenCV (cv2) library and camera input is given, input data is predicted from the saved model and a bounding circle is drawn around the face of the individual that predicts whether the person is wearing a mask or not.

The implemented model will contribute enormously to the public health care system by predicting the presence of masks on the faces of the individuals and will restrict the spread of the deadly virus. It will act as an human alternative for detecting masks on the faces of human individuals thus making lives more convenient. It will also give information about the person who is not wearing a facial mask as a precautionary measure in the outset of COVID-19 to encourage people to wear their mask and make it a habitual activity.

6.2 Future Scope :

1. In the future it can be seen as a system to detect face masks and encourage individuals to wear masks for their own safety. This model can be extended in future to detect if a person is wearing the mask properly or not and also to identify them. The model can be further improved to detect if the mask is virus prone or not i.e. the type of the mask is surgical, N95 or not.
2. It would be extremely hard to identify every person without a mask for our proposed system. Hence in order to optimise perfectly and implement our project on a large scale, it should be implemented on the large number of CCTV cameras in the city to monitor facemasks around the whole city as well as dedicated manpower should be recruited to enforce proper laws on the law violating individuals.
3. In the future we would like to put forward a face mask detection system in which the law breakers will get a mail, fax or mobile notification for not wearing a facemask so that our government can recognise them and take initiatives like interactive sessions on COVID safety and send them emails and notifications on COVID safety .
4. This would work in a way such that CCTV supervised cameras would detect an individual's face and generate an identification number to identify that person with incorporation of embedded IoT and telecommunication devices which will also help to alert the irresponsible citizens and rectify them , and rest of the action is taken care by the government employed personnels. This model can further be used in corporate organisations , trading groups and economic enterprises to ensure that employees are abiding by the rule and wearing their masks
5. This system can also be deployed to identify their presence and mark their attendance for the working days. For an individual without a mask the system would clang an alarm to alert the security guards about the safety issue and to help the individual abide by the rules .
6. Some other horizon of implementation of this project lies within the healthcare industry where wearing masks are mandatory.

INDIVIDUAL CONTRIBUTION REPORT:

FACE MASK DETECTION USING MobileNetV3

Abstract: This project titled Face Mask Detection has been developed using a machine learning technique known as transfer learning and by using image classification method of MobileNetV3small .Steps followed while building the model are data collection, data preprocessing , data splitting , model-rectification ,model testing and model implementation. Model is capable of predicting if individuals are wearing or not wearing masks at an accuracy of 97 percent.

Individual contribution and findings:

It was my proposal to do this project on FaceMask detection using the third generation CNN Image classifier developed by google MobileNetV3Small. I single-handedly trained the model using keras. tensorflow and I plotted graphs using matplotlib for loss and accuracy during training phase and found the accuracy to be 97 percent.I used the caffe model that is implemented using prototxt for image segmentation and testing using input in the form of image and video (camera).I found that the model is able to effectively predict, compare and differentiate between masked and unmasked individuals with utmost accuracy

Individual contribution to project report preparation:

I took the initiative to start the project report and personally designed the report ensuring all other members are contributing to this project.I personally wrote the Abstract, chapter-1, chapter-2,Chapter-4 and added all the 12 diagrams myself. I used 11 references to write this report and edited the entire report myself.Finally I compiled the entire report for submission

Individual contribution for project presentation and demonstration:.

I took the initiative to design the presentation and encouraged all the team members to attend the presentation positively. I personally ensured that all the members of the group are presenting and are given some part of their own to present in order to ensure the success of the project.

Full Signature of Supervisor: ……………….…………..
Full signature of the student: …………………….……….

INDIVIDUAL CONTRIBUTION REPORT:

Individual contribution and findings:

Created the function in the program in which a face mask gets detected using a camera. In this function a face is detected in a frame from the video stream. It creates a circle around the face and colours it appropriately green for mask and red for without mask. The frame exhibited gets closed when we press “o”. This function implements OpenCV-Python cv2.imread() method to create the frame and draw the circle around the face using the coordinate of the ROI (face area).OpenCV is a library for machine learning and processing the image. By using OpenCV we can process images and videos to identify objects, faces and human handwriting.

Individual contribution to project report preparation:

Did the Chapter 3 of the Project report:

Problem Statement / Requirement Specifications (SRS)

·1.Project Planning 2.Project Analysis

In this part of the report, I explained the implementation of the project in step-by-step format. Chapter 3 also contain the diagrammatic implementation of the project. The (UML) diagram contain “training the face mask detector” and “deploy the face mask detector model ” in detail.

Individual contribution for project presentation and demonstration:

I elaborate and explain the Image testing and video testing slides with their results in the Project presentation. The slides include the detailed explanation of how the model is implemented in the programs to detect face masks in an image or by using a video(camera) andimage to give appropriate results.

Full Signature of Supervisor: ……………………………...
Full signature of the student: …………………………….

INDIVIDUAL CONTRIBUTION REPORT:

Individual contribution and findings:

The project enabled me to apply several machine learning implementations to the project.I understood about the models and its uses. After an intense discussion, we decided the credential we needed to choose and I oversaw the progress of this project which ran smoothly.

Individual contribution to project report preparation:

In the report was the conclusion where my teammates needed my help. I contributed to chapter 3 where the conclusion and future scope were written. I also proofread the SRS document and other chapters if requiring any changes.

Individual contribution for project presentation and demonstration:

We showcased our project in its completion in a presentation. I explained why our project is needed, explaining the use of Neural Networks and showing mathematical models where necessary. For the preparation of the presentation I helped choose the format in the presentation.

Full Signature of Supervisor: ……………………………..
Full signature of the student: …………………………….

Facemask Detection Using MobileNetV3

ORIGINALITY REPORT

10%	6%	7%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Harish Adusumalli, D. Kalyani, R.Krishna Sri, M. Pratapteja, P V R D Prasada Rao. "Face Mask Detection Using OpenCV", 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), 2021 Publication	2%
2	virinchisai.medium.com Internet Source	2%
3	www.researchgate.net Internet Source	1%
4	medium.com Internet Source	1%
5	www.nature.com Internet Source	1%
6	Trisiladevi C. Nagavi, R.B Sharanya, S Minchu, Raja Kumar, Manuprasad. "COVID-19 Face Mask Detection Using Deep Learning", 2021 6th International Conference on Signal	<1%

	Processing, Computing and Control (ISPCC), 2021 Publication	
7	ijircce.com Internet Source	<1%
8	Ying He, Zheng Zhang, F. Richard Yu, Nan Zhao, Hongxi Yin, Victor C. M. Leung, Yanhua Zhang. "Deep-Reinforcement-Learning-Based Optimization for Cache-Enabled Opportunistic Interference Alignment Wireless Networks", IEEE Transactions on Vehicular Technology, 2017 Publication	<1%
9	Ting Sun, Miklos A. Vasarhelyi. "Predicting credit card delinquencies: An application of deep neural networks", Intelligent Systems in Accounting, Finance and Management, 2018 Publication	<1%
10	dspace.chitkara.edu.in Internet Source	<1%
11	M.I Shiny, Albin Fredy, Jeena Jose, Jeseena.K. P, Rohith P V. "Covid-19 Protocol Enhanced Using Computer Vision System", 2021 Innovations in Power and Advanced Computing Technologies (i-PACT), 2021 Publication	<1%

12	Srinidhi Srinivasan, Rubasri Sundar, Hemavathi S, Sumathi Sokkanarayanan, Mithileysh Sathiyanarayanan. "Face Recognition and Mask Detection Technology for Surveillance Systems", 2021 4th International Conference on Computing and Communications Technologies (ICCCT), 2021 Publication	<1 %
13	www.pyimagesearch.com Internet Source	<1 %
14	Chinthaka Gamanayake, Lahiru Jayasinghe, Benny Kai Kiat Ng, Chau Yuen. "Cluster Pruning: An Efficient Filter Pruning Method for Edge AI Vision Applications", IEEE Journal of Selected Topics in Signal Processing, 2020 Publication	<1 %
15	Lu Cheng, Ruocheng Guo, Huan Liu. "Causal Mediation Analysis with Hidden Confounders", Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, 2022 Publication	<1 %
16	Submitted to Thapar University, Patiala Student Paper	<1 %
17	Submitted to University of Surrey Student Paper	<1 %

18	Yiyue Luo, Yunzhu Li, Pratyusha Sharma, Wan Shou et al. "Learning human–environment interactions using conformal tactile textiles", Nature Electronics, 2021 Publication	<1 %
19	www.bach-phys.ritsumei.ac.jp Internet Source	<1 %
20	www.groundai.com Internet Source	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		