CS 584 PROJECT REPORT (SPRING 2021)
FACE MASK DETECTION

Neha Reddy | G01281501
Samriddhi Dashora | G01269042

ABSTRACT

The COVID-19 pandemic has adversely affected every aspect of life. Academic institutions and other organizations are badly affected. There is no hope left for the situation other than bringing things back to normal. Normalizing the operations in various sectors is the need of the hour but it is important to keep safety in mind while doing so. Most people are getting vaccinated but that doesn't guarantee safety from the virus until everyone is vaccinated. So, wearing face masks is essential until everyone is immune to the disease. But very often people forget to wear their masks and that increases the risk of being infected. As a solution to this we created a system which can alert the people when they are not wearing a mask. Our model uses OpenCV, TensorFlow, and Keras packages to perform object detection, feature extraction, and classification. We have used various techniques to improve the image quality such as Auto White Balance, Gaussian Blur, Median Blur and Unsharp Filter. We have trained our models on 10k images with mask and without mask and achieved excellent accuracy in predictions for test data. For this project, we have trained models like ConvNet, InceptionV3, NASNETMobile, and MobileNetV2, and did a comparative study to find the best model fit for the task. MobileNetV2 gave the best test accuracy of 99.60% and since it can be easily deployed on any mobile devices it was used for real-time facemask detection along with Haar cascade classifier to determine if a person in front of the camera is wearing a mask or not. This project is promising in tackling the spread of the virus and ensuring that the safety guidelines are followed.

1. INTRODUCTION

Currently, the world is facing many challenges due to the pandemic. According to the latest report, COVID-19 has globally infected over 147 million people causing over 3.11 million deaths. The coronavirus spreads across people mainly through droplets and particles released from an infected person while he/she is sneezing or coughing. One might think that isolating infected people is enough to solve this problem but it is not because not only people who look sick but anyone could be a carrier of the virus. So, wearing face masks is essential for everyone to prevent the spread of the disease. Apart from risk to one's life due to COVID-19, there has been another major issue and that is the fall of the world economy. To deal with this crisis, opening up the government, public and private organizations all over the world and normalizing the operations is the only way. But with the lockdown lifted, there is even more danger for the virus to spread so it is important for the organizations to keep a check of whether people are following COVID-19 safety guidelines or not and it can be achieved through our automated face mask detection model.
In the rest of the report, the motivation for choosing this topic, the dataset used, models implemented, metrics for evaluating our model's performance, and future work shall be discussed.

2. PROBLEM STATEMENT

With the rising cases of COVID-19, it has become very important to find a way to stop the disease from spreading. Before the whole population is completely vaccinated, it is essential to reduce the death toll as much as possible. This can be achieved if people follow the safety guidelines such as wearing a mask in public, maintaining proper social distance, and washing hands regularly. But these rules are new to the people and there are times when they forget to wear a mask or maintain social distancing. This is especially the case of employees whose work requires them to be outside their homes. In such cases, organizations have to take the responsibility to ensure the safety of employees. Yet, manually monitoring people to check whether they are wearing a face mask or not is very difficult. One solution to this problem is to use automated face mask detection technology integrated with surveillance cameras to check if the people are following safety guidelines or not and alert them if they are not following the rules.

3. LITERATURE REVIEW

We have reference from the following sources to build our model-

TITLE/LINK	METHODS USED	COMMENTS
1) **Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV**	ConvNet	The paper outlined the basic methodology of using cascade classifier and sequential convolutional neural networks to build face mask detection models.
2) **Prototype for Integration of Face Mask Detection and Person Identification Model – COVID-19**	MobileNetV2	The paper discussed a way to identify people who do not wear masks and alert them. MobileNetV2 is used both for classification as mask or no mask and person identification.
3) **Real-Time Face Mask Detector Using YOLOv3 Algorithm and Haar Cascade Classifier**	YoloV3 and Haar Cascade.	The paper discussed how a real time face mask detector can be built by using a haar cascade classifier for the face detection and YOLOv3 for classification.
4)**Covid-19 Facemask Detection with Deep Learning and Computer Vision**	MobileNetV2	The paper explained in detail what different packages can be used for this task. They used MobileNetV2 for feature extraction and three machine learning algorithms for classification.
5) **Face Mask Detection using MobileNet and Global Pooling Block.**	ResNet50, MobileNet, GoggleNet and	The paper explains how Global Average Pooling Block when used along with ResNet50, GoogleNet and MobileNet models

	Global Pooling Block(GP)	reduces overfitting and increases accuracy in predictions.
6) **Smoothing Images** - *https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_filtering/py_filtering.html*	Median Blur, Gaussian Blur and Bilateral Blur	The website explained different filtering techniques and how they can be applied to the images.
7) **Unsharp Filter-** *https://homepages.inf.ed.ac.uk/rbf/HIPR2/unsharp.htm*	Gaussian Filtering	The website discussed in detail on how unsharp filters work and how it can be used to enhance the image edges.

4. METHODS AND TECHNIQUES

Data Processing-

The OpenCV package was primarily used for pre-processing the images. The following are steps we followed for pre-processing our images-
1) The images were resized according to the minimum size requirements of the neural network used. Used *cv2.resize()* method from the OpenCV package for this purpose.
2) Convert the images to grayscale using cvtColor() method with cv2.COLOR_BGR2GRAY as a parameter to reduce the dimensions.
3) Filtered the images to remove noise using filtering techniques like Gaussian Blur, Median Blur, and Bilateral Blur. For the median blur, a filter size of 5x5 was used whereas for gaussian blur a filter size of 9x9 was used to cover large portions of the image.
4) Used Auto White Balance to remove the unwanted color from the images. A simple auto white balance algorithm was used with 30% of the top and bottom pixels removed.
5) Enhanced the important edges of the images using an unsharp filter. The white-balanced image was blurred with a gaussian filter of size 9x9 and standard deviation 8 along the x and y-axis. Then, the original image is sharpened by adding it to the difference of the original image and blurred image with the weight factor 4.
6) Normalized the pixel values to a range of (0,1).
7) Performed one-hot encoding on the labels with_mask and without_mask to convert them into numerical labels 0 and 1. Reshaped the image data to 4-Dimensional arrays.

Models-

A comparative study of four neural networks was done to find the best model in terms of accuracy and time taken to train.

1) Convolutional Neural Networks (ConvNet)-

The simplest model used so far in this project is Convolutional Neural Networks. ConvNet captures the spatial dependencies of image pixels through layers of filters. The layers created for the CNN in this project are-

a) Convolution layer with 32 filters, 3x3 filter size and same padding to capture important features of the image.
b) Max Pooling Layer with filter size 2x2 to reduce the number of dimensions.

c) Dropout Layer with 10% units dropped to prevent overfitting.
d) Convolution layer with 64 filters, 3x3 filter size and same padding to capture high level features of the image.
e) Max Pooling Layer with filter size 2x2 and dropout layer with 10% drop rate.
f) Flatten Layer to convert the input to 1-D tensor to give as input to FC layer.
g) Dense layer with activation function ReLu and 512 units.
h) Final dense layer with 2 neurons to classify as 'with_mask' or 'without_mask' with the help of activation function softmax.

The optimizer used for the CNN was 'Adam' with default LR of 0.001 and the loss function used was categorical cross entropy. The model trained for 10 epochs.

2) InceptionV3-

InceptionV3 is a convolutional neural network of 48 layers used for object detection and feature extraction. We have used a pre-trained version of the neural network with weights of 'Imagenet'. The minimum size requirement of images is 75x75 for this model and it takes only colored images as input. The last fully connected layer of InceptionV3 model is removed and replaced with the Global Average Pooling Layer with the pooling parameter set as 'avg' to prevent overfitting. The weights of the model are kept the same by freezing the model by setting layers.trainable as 'False'. The first dense layer was given 128 units and activation function as ReLu followed by a dropout layer of 40% dropout rate to further reduce overfitting. The final dense layer used 2 neurons and a softmax activation function for classifying. The optimizer, loss function and number of epochs are kept the same as the ConvNet model.

3) NASNETMobile-

NASNETMobile is a convolutional neural network which can be easily deployed on any device because of it's low memory usage and good performance in terms of accuracy. The minimum size requirement of images is 32x32 for this model and it takes only colored images as input. The last fully connected layer of NASNETMobile model is removed and replaced with the Global Average Pooling Layer with the pooling parameter set as 'avg' to prevent overfitting. The weights of the model are kept the same setting layers.trainable as 'False'. The first dense layer was given 512 units and activation function as ReLu followed by a dropout layer of 50% dropout rate to further reduce overfitting. The final dense layer, optimizer, loss function and number of epochs are kept the same as the InceptionV3 model.

4) MobileNetV2-

MobileNetV2 is another neural network which can be deployed on mobile devices for use for feature extraction. We have used a pre-trained version of the neural network with weights of 'Imagenet'. The minimum size requirement of images is 96x96 for this model and it takes only coloured images as input. The last fully connected layer of NASNETMobile model is removed and replaced with the Global Average Pooling Layer with the pooling parameter set as 'avg' to prevent overfitting. The weights of the model are kept the same setting layers.trainable as 'False'. The first dense layer was given 512 units and activation function ReLu followed by a dropout layer of 30% dropout rate to further reduce overfitting. We have used the 'sigmoid' as the activation function in the final dense layer.

Real Time Detection-

Haar cascade classifier was used to detect the location of the face in an image and the mobileNetV2 model which performed best of four models is used to classify the image as ‘with mask’ or ‘without a mask’. Since the dataset we used to train our model had images filled mostly with faces and a very little background, so we used the Haar Cascade Classifier to get only the face portion of the image and classified them.

The following steps are followed for real-time face mask detection-

1) A path to the source file containing the Haar Cascade method for frontal face in GitHub is established and the cascade classifier is loaded into the notebook.

2) The camera is turned on to capture the image. The captured image is read and detectMultiScale() method of the cascade classifier is used to find the dimensions of the face from the image.

3) A rectangle of defined color and thickness is created to hover around the face portion of the image. With the help of the dimensions extracted from the detectMultiScale() method, the image can be sliced to get only the face part of the image.

4) The image is pre-processed similarly to the train data and finally the model.predict() method was used to classify the image as ‘with mask’ or ‘without mask’.

5. DISCUSSION AND RESULTS

5.1 DATASETS

The dataset we used for our project was the 'Face mask ~12K Images dataset' taken from the kaggle website. The training and testing datasets are very balanced with an equal number of images with mask and without mask. Additionally, the dataset is diverse, having images with different masks and taken from different angles.

dataset	With Mask	Without Mask
Training	5000	5000
Testing	483	509
Validation	400	400

5.2 METRICS

The metrics we used to evaluate the performance of the models were-

1) Accuracy-

Accuracy is calculated by dividing the number of correct predictions made by the model in each epoch with the total number of predictions made in that epoch. Since the dataset is balanced with an equal number of images in each class accuracy works well in evaluating the performance. The accuracy is calculated for training and testing data in every epoch.

2) Loss -

Loss function is the prediction error that the model seeks to minimize by making necessary changes. Categorical cross entropy and Binary cross entropy were the two probabilistic loss

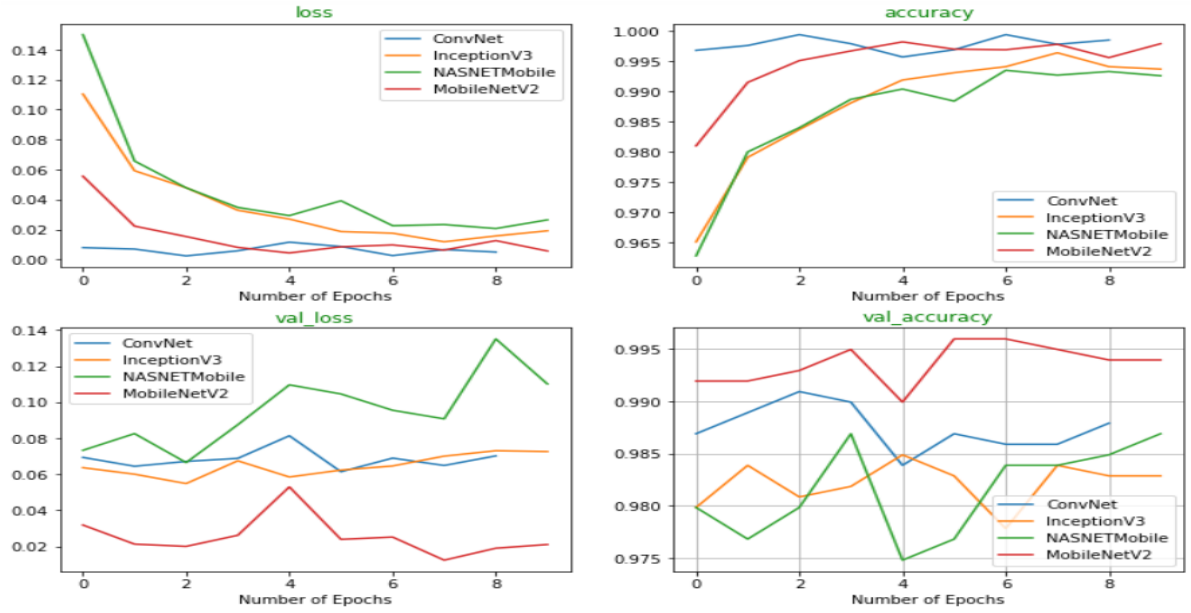
functions used interchangeably in the project. The loss for training and testing is calculated for each epoch. When the loss of any model decreased its accuracy increased.

5.3 EXPERIMENTAL RESULTS

We have observed the following while training our models-

1) Using softmax activation function instead of other activation function in the last fully connected layer increased the accuracy in the predictions for 3 models. But, in MobileNetV2 using the 'sigmoid' function in the last layer gave better accuracy than softmax.
2) Pre-processing the images with filtering, white balance and edge enhancement techniques has given a 4% increase in accuracy for ConvNet but it was computationally expensive.
3) Appending a global average pooling layer to the last layer of pre-trained models has greatly increased the accuracy.
The below graph shows the loss and accuracy of each model comparatively-

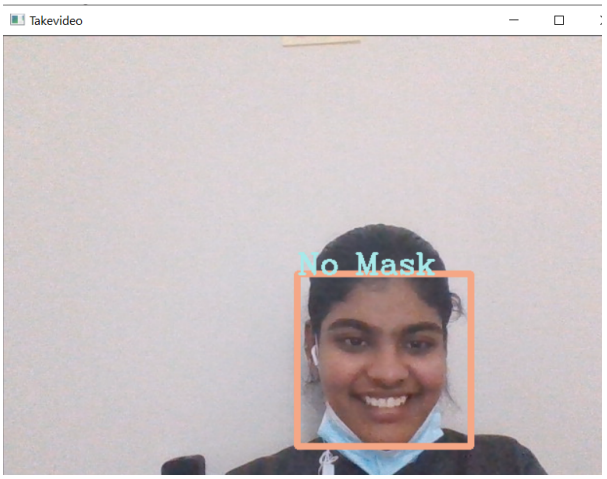
The graph plots the training and validation loss and accuracies for each model over 10 epochs. ConvNet has the highest training accuracy and lowest training loss so it seems to be overfitting compared to other models. Whereas, NasNetMobile has lowest training accuracy and highest training loss compared to other models. MobileNetV2 has the highest testing accuracy and lowest testing loss so it works best in predicting for unknown images out of all models. Whereas InceptionV3 has the least testing accuracy and NasNetMobile has the highest testing loss of all models. Also it can be observed from the graph, that training accuracies and loss do not vary much across the epochs for every model but the testing accuracies and loss differs much between every epoch.



The following table shows the top accuracies of the models implemented-

MODEL	ACCURACY
MOBILENETV2	99.60%
CONVNET	98.79%
NASNET MOBILE	98.68%
INCEPTIONV3	98.49%

Real time mask detection pictures-



6. CONCLUSION

Among all the models implemented in the project, MobileNetV2 performed best with an accuracy of 99.6% in the predictions and so it was used for real-time face mask detection. MobileNetV2 is lightweight and occupies very little memory space of 14MB so it can be deployed and used effectively on many mobile devices. Additionally, our model has proven effective in detecting the presence of face masks irrespective of the color or design patterns on the masks. Also, the MobileNetV2 model has shown good performance in accurately classifying the real-time face images as ‘with mask’ when the mask is worn and ‘without mask’ when the mask is removed. Though filtering, white balance and edge enhancement techniques helped in improving the accuracy of the predictions in ConvNet, it was computationally expensive so we decided not to use these techniques for pre-processing the images for the other 3 models. Softmax activation helps training converge faster than others so using it has increased the accuracy in all models except MobileNetV2. Also, the reason Global Average Pooling Layer worked so well in increasing the Testing accuracy must be because it was greatly reducing the overfitting by reducing the number of features.

6.1 DIRECTIONS FOR FUTURE WORK

In the future, we would like to integrate our face detection model with the face recognition model to identify the person who has violated the rules and alert them about it. Apart from this, we want to integrate our model with the model which calculates the social distance

between any two people and sends an alert whenever the people cross the minimum boundary of 6 feet or do not wear a mask. This ensures people are safe from the spread of the virus in every possible way.

REFERENCES

[1] Arjya Das; Mohammad Wasif Ansari; Rohini Basak Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV Publisher: IEEE

[2] Samuel Ady Sanjaya; Suryo Adi Rakhmawan Face Mask Detection Using MobileNetV2 in The Era of COVID-19 Pandemic Publisher: IEEE

[3] K Suresh; MB Palangappa; S Bhuvan Face Mask Detection by using Optimistic Convolutional Neural Network, Publisher: IEEE

[4] Mohammad Marufur Rahman; Md. Motaleb Hossen Manik; Md. Milon Islam; Saifuddin Mahmud; Jong-Hoon Kim An Automated System to Limit COVID-19 Using Facial Mask Detection in Smart City Network, Publisher: IEEE

[5] Sammy V. Militante; Nanette V. Dionisio Deep Learning Implementation of Facemask and Physical Distancing Detection with Alarm Systems Publisher: IEEE

[6] Vinitha.V, Velantina.V COVID-19 FACEMASK DETECTION WITH DEEP LEARNING AND COMPUTER VISION

[7] Spatial Filters - https://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm

[8] Unsharp Filters - https://homepages.inf.ed.ac.uk/rbf/HIPR2/unsharp.htm

[9]Susanto Susanto; Febri Alwan Putra; Riska Analia; Ika Karlina Laila Nur Suciningtyas The Face Mask Detection For Preventing the Spread of COVID-19 at Politeknik Negeri Batam, Publisher: IEEE

[10]Md. Sabbir Ejaz; Md. Rabiul Islam Masked Face Recognition Using Convolutional Neural Network.