**FACE MASK PREDICTION**

**FINAL PROJECT REPORT**

**Introduction**

Coronavirus disease 2019 has affected the world seriously. One major protection method for people is to wear masks in public areas. Studies have proved that wearing a face mask significantly reduces the risk of viral transmission as well as provides a sense of protection. Furthermore, many public service providers require customers to use the service only if they wear masks correctly. However, it is not feasible to manually track the implementation of this policy, using a combination of image classification, object detection, object tracking, and video analysis, I’ve developed a system that can detect the presence and absence of face masks in images as well as videos. The main motivation of this project is to detect the people who all are not wearing masks in a public place with the help of real time surveillance. This project is implemented using CNN (Convolutional Neural Network) model, which is a Deep Learning algorithm, and executed in Python using Keras, TensorFlow and OpenCV.

As mentioned above, this model can detect face masks in either image as input or videos which are integrated with capturing devices like CCTV camera, so this is more convenient to track safety violations, promote the use of face masks, and ensure a safe working environment.



**Fig1. Recognizing the face and detecting the mask**

**Source / Exploratory Data Analysis**

There are two face mask classifier models which are trained on the dataset. The dataset containing images which are of two classes, ‘masked faces’ and ‘unmasked faces’, these datasets were available in Kaggle. In this particular dataset the ‘unmasked faces’ has nearly 1800 photos of random people around the world, on the other hand in ‘masked faces’ there are almost 1800 photos of random people, but here the images are morphed with a cropped face mask (as shown in the figure) over their nose and mouth.

**Fig2a. Unmasked faces Fig2b. Masked faces**

The reason why the ‘masked faces’ are morphed with a cropped face mask is to train the model more efficiently, in the case of a regular photo with a person wearing an actual mask, which is much less efficient because of the color of mask, and the position of the mask varies image to image. In this case, by morphing with a same mask at similar position to every image makes the model to train more efficient.

As this is a supervised learning, for testing the model, I’ve similar dataset which I call it as a test data classified into two different classes, ‘unmasked faces’ and ‘masked faces’, but now, in the ‘masked faces’, the testing is done on an image of person wearing an actual mask not a morphed photo, so that the output would be more realistic. These are around 700 images of each class. Using Keras tool, I categorized these images into a target file, which represents (1,0) for ‘mask’ and ‘no\_mask’ and the target file is directly linked to the test dataset.

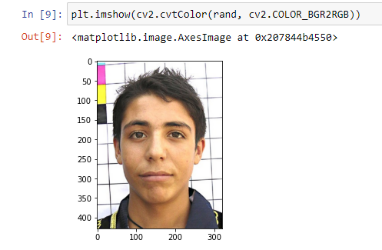
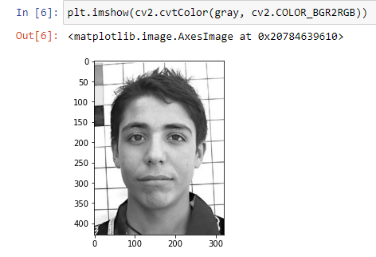
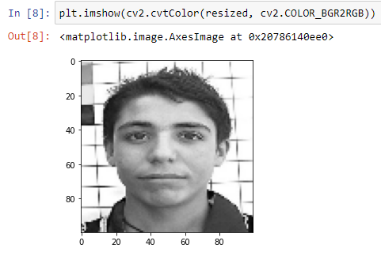
  **** ****

  ****

**Fig2c. Testing unmasked faces Fig2d. Testing masked faces**

**Methods / Preprocessing Parameters**

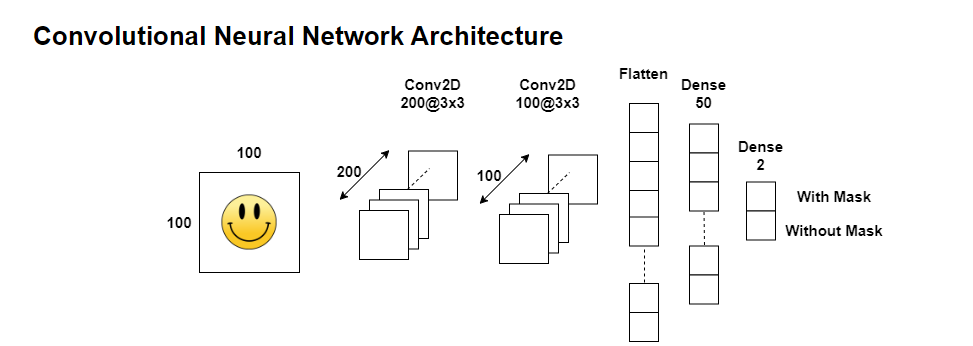
Initially the dataset which I got from Kaggle has more noise, here the noise indicates uneven dimensions of image which varies image to image and high contrast colors. For cleaning this dataset, I converted the image color into gray scale using OpenCV tool, covert color method and resized the gray scale into 100x100 using same tool, resize method, so that there will be a fixed common color and size for all the images in the dataset. As shown in the image below, the same process is done for all images in the dataset. This way all the input dataset is cleaned.

  ****

RGB2GRAY Resize 100x100

**Fig3a. Converting image into gray scale and resizing 100x100**

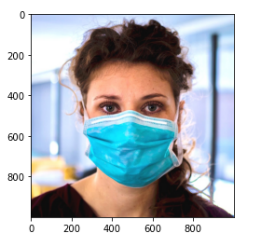
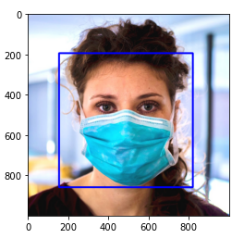
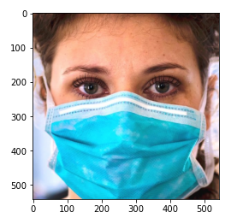
I implemented this algorithm using Convolutional Neural Networks (CNNs) which is a key aspect in modern Computer Vision tasks like pattern object detection, image classification, pattern recognition tasks, etc. A CNN uses convolution kernels to convolve with the original images or feature maps to extract higher-level features, thus resulting in a very powerful tool for this project. Here the input image of size is 100x100, after importing, it has 2 convolutional layers as shown in the below image, after the pooling stage, the layers are flattened and then finally they’re connected to a dense layer of 50 neurons, and the last layer will be two neurons ‘With Mask’ and ‘Without Mask’.

****

**Fig3b. Convolution Neutral Networks Architecture**

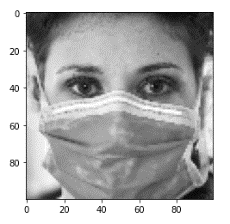
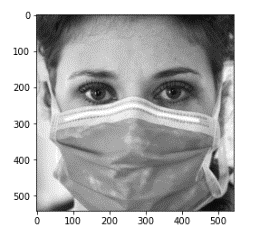
In a neural network, the activation function is responsible for transforming the summed weighted input from the node into activation of the node or output for that input. Here in this algorithm, I used Rectified Linear Activation function (ReLU), this will output the input directly if it is positive, otherwise, it will output zero. In the final layer, I used SoftMax activation function to classify the result.

For testing the algorithm, I used a Cascade Classifier called ‘haarcascade\_frontalface\_default.xml’, this basically identifies the facial features in any input image. After detecting the face, it returns the parameters of the detected facial feature in the image, so I used those parameters and cropped the image to that facial parameters. Now its all set to the format of the dataset which I used to train data, so now following the same data cleaning process, converting into gray scale, and resizing to 100x100, then after importing it into the algorithm it predicts the classifier either ‘with facemask’ or ‘without facemask’. The entire process is visualized in image below.

   cropping the face

Cascade Classifier

Original image facial recognition

100x100 

CNN

Color change to gray scale

With Mask

Without Mask

**Fig3c. Detecting faces with and without masks**

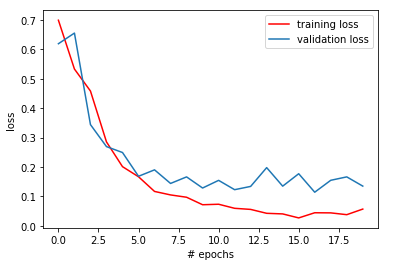
As I mentioned earlier, this algorithm detects face mask in video too, which are with video capturing devices like CCTV camera, so the algorithm captures every single frame in the video and considers it as a input image and then the predicts the same way it does predicts in an regular image and displays the output for every single frame, for more visual results, I designed a frame which fits on the facial features, which indicates a red frame with a red color text ‘No Facemask’ on top of the frame if the person in the video is not wearing a mask, and a green frame with a green text ‘ Facemask’ if the person wears facemask. This applies for every single frame in the video stream and it displays as well.



**Fig3d. Example : Visual illustration of the algorithm in CCTV cameras**

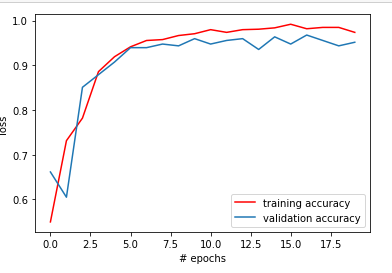
**Results / Obstacles**

In the implementation process, I imported test data file for evaluation, while training the data I used 20 epochs, that was a quiet long duration to train all the 1900 images, yet the results were satisfying. The graph of training loss and validation loss is below. As in graph we can see that initially training loss and validation loss are very high during 1-2 epochs, and they both have similar pattern up to 5 epochs, and from there the validation loss stated deviating, and it tries to stay within 0.1 and 0.3. We can also see that as the number of epochs are increasing the training loss is getting decreased steadily.



**Fig4a. Graph: training loss VS validation loss**

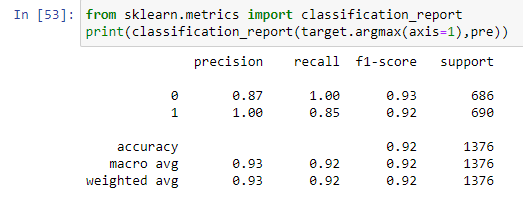
In the graph below, we can see that the training accuracy was increasing as number of epochs increase, in similar ratio the validation accuracy was also increasing with number of epochs increase. While implementing the train data, there was a model check point tool, which monitors the change in validation\_loss, so if there is any increment in validation loss by previous epoch, then it’s not going to save the model, if the val\_loss is going down then the model is saved in the same directory, so now choosing the best model could be the one which have least validation loss, which represents the best model.



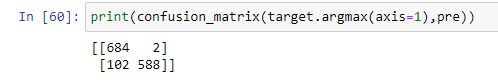
accuracy

**Fig4b. Graph: training accuracy VS validation accuracy**

Here’s the classification report and confusion matrix, the accuracy of the test data is 92%, which is a still decent model to predict and significant recall with 100%, and as per prediction the total number of images with no mask was 689 and 690 images with masks as per prediction.



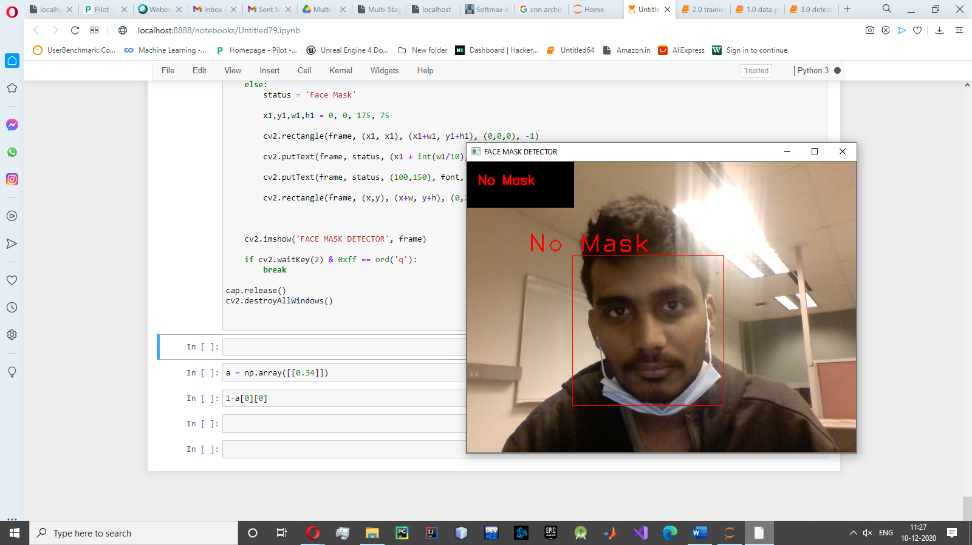
**Fig4c. Classification report**

****

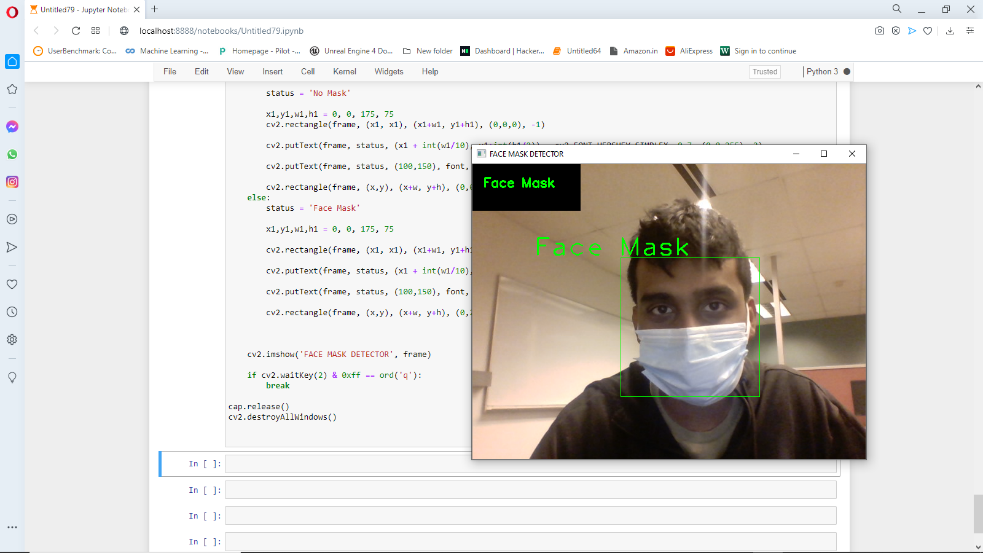
**Fig4c. Confusion Matrix**

**Obstacles**

* Initially in the stage of data preprocessing, I started with a dataset which doesn’t have any morphed photos of a cropped face mask, even though I had a large dataset of 450 images the accuracy was very low. I had to search in GitHub, stack overflow for this issue, so I figured out there’s a dataset of morphed photos with cropped facemask, and their illustration representing to generate those morphed datasets using OpenCV. So, I programmed it and generated those images with morphing for around 700 images.



**Fig4d. Webcam: without facemask**



**Fig4e. Webcam: with facemask**

**Conclusion**

In this report, I mentioned about the main motivation of the project, which is to detect the people who all are not wearing masks in a public place with the help of real time surveillance, and by using few preprocessing steps to clean the data by converting into gray scale and resizing it. The implementation of the algorithm using CNN with multiple layers which is used to classify the input image by ‘with Facemask’ and ‘without Facemask’. After the training of the data, new raw input image datasets are considered to be test data, which consist of all kinds of people few are with and without mask, with a target file which represents its value (0,1). By using cascade classifier facial features are recognized in the input image and it is predicted.

In times of the COVID-19 pandemic, this system can be easily deployed for automated monitoring of the use of face masks at public places, which will help make them safer. This model can be used in few public places like airport, bus stations or shopping malls by keeping a big screen up front with a security camera, so it will detect all the faces who all are captured in the video recording, and highlight the faces by classifying either the person is wearing a mask or not, one of the best instance is Walmart, in any of the branches of Walmart the entrance door has a security camera with a big screen facing to people, so its more helpful if these kind of systems are integrated to it.