**FACE MASK DETECTOR**

**INTRODUCTION**:

As COVID-19 pandemic continues and has caused death to millions of people world-wide, it has become mandatory to wear face masks in grocery stores, restaurants, offices and other public places. Wearing a mask, especially when in close proximity to others, is imperative to slowing the spread of COVID-19. Considering the present scenario, we have come up with a face mask detector project which helps in detecting whether a person has worn a face mask or not.

**OUR TEAM:**

D20009 Chithra Nair

D20010 David Jose Paliakkara

D20012 Eric Gonsalves

D20013 Gadivemula Vamsitej

D20030 Saurabh Gupta

**TECHNOLOGY USED:**

convolutional neural networks(CNN) ,Python, TensorFlow and OpenCV

**DATASET:**

The dataset is originally created by combining different datasets available in Kaggle and GitHub. The dataset is augmented and consists of 37776 images containing I890 face images with masks and 1886 without masks

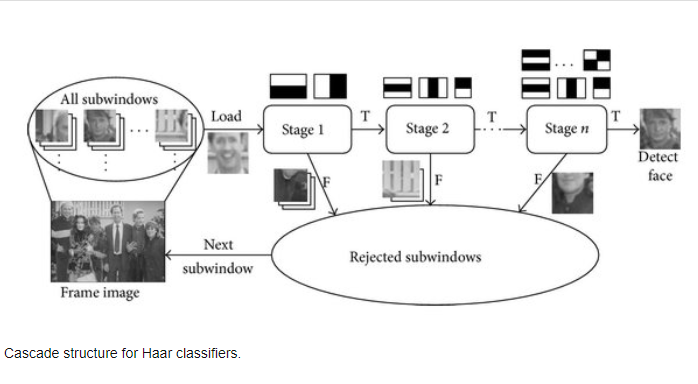
**PROCESSES INVOLVED**:

We have divided our project into 3 stages.

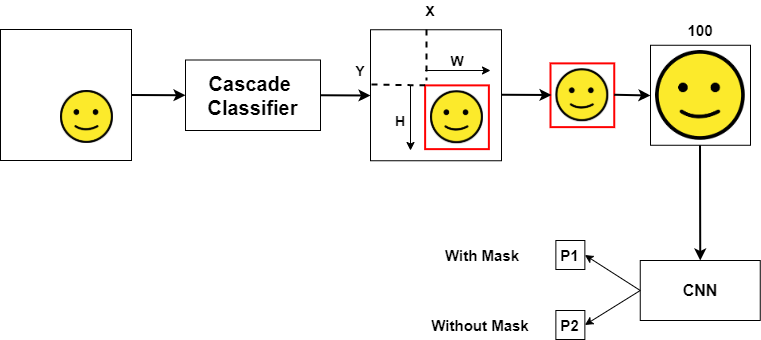
1. Data pre-processing
2. Training CNN model
3. Prediction on test data
4. Predicting on uploaded image
5. Live testing with webcam

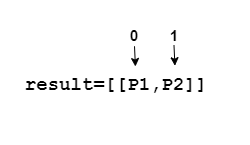
**PROJECT OVERVIEW:**

An image with a face is fed to a Haar cascade classifier. It is an Object Detection Algorithm used to identify faces in an image or a real time video. The algorithm uses edge or line detection features from the input image and thus gives the region of interest of the face. An architecture of Haar classifiers is given below:



After the region of interest of the face image is obtained, the image size is re sized to 100x100 image size. This is again passed to a pretrained CNN and we get prediction as mask detected or no mask detected. The overall process is given below:





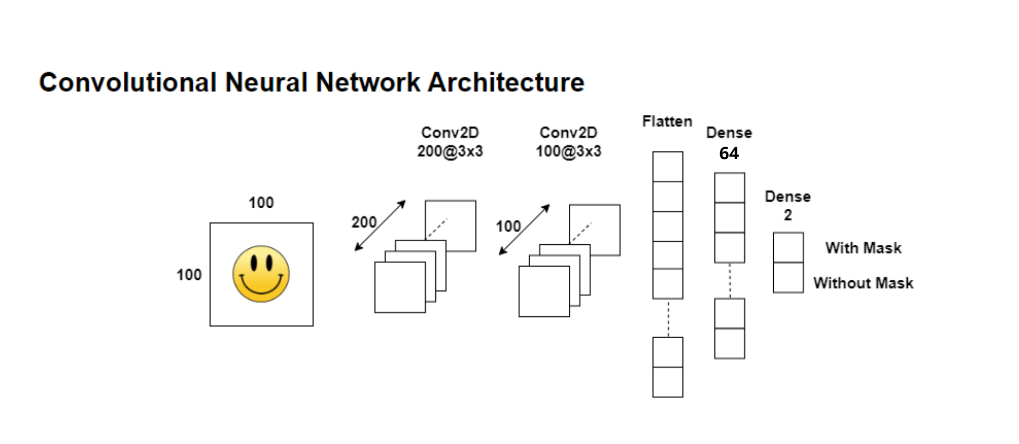
1. **DATA PREPROCESSING**

The dataset contains images with different colours, sizes and orientations. The image is first converted to gray scale and resized into 100x100 since a fixed size is required for all images. Then the resized images are appended to data and target is appended to 0 or 1 where 0 is with mask and 1 is without mask. Exceptions are handled for any errors that might come. The data are then normalized by dividing by 255 so that the pixel range is converted into 0 and 1. Since the neural network needs a 4-dimensional array, the array is reshaped into 4-dimensional array. Likewise, the target is also converted into a NumPy array. Finally, the target is converted to a categorical representation as the last output layer has 2 neurons such as with mask and without mask. Now the data and target are saved. Data contains images and target contains the prediction with, without mask.

1. **TRAINING CNN MODEL**

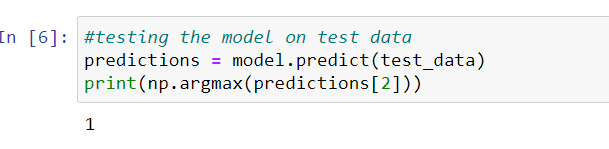
The pre-processed image is fed to the CNN model. It has 2 convolutional layers. The first layer contains 200 convolutional filters of size 100x100 each and second consists of 100 convolutional filters of size 100x100 each. Here ReLu activation function is used followed by max pooling layer of size 2x2. The convolutions are flattened and connected to a dropout layer to prevent overfitting and then to a dense layer of 64 neurons. The last layer is the output layer which has the 2 neurons with output with mask, without mask. The loss function used is categorical cross entropy, with adam optimizer and metrics as accuracy is set.

The dataset is divided into train and test. The model is fitted with 20% of validation data. We received a training accuracy of 0.9831 and validation accuracy of 0.9647. The model is then evaluated and it gives a testing accuracy of 0.933.



1. **PREDICTION ON TEST DATA**

The model that was tested was used to predict an image in the test data and the out put was accurate it predicted 1 for non mask and and 0 for mask

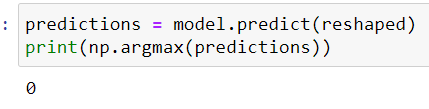


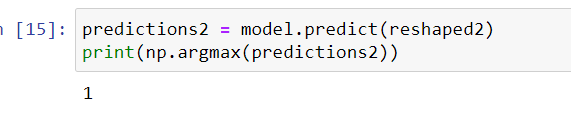


1. **PREDICTION ON UPLOADED IMAGE**

We tested the model on our own uploaded image which was not present in the test or train dataset and the model predicted accurately the images with and without mask





1. **LIVE TESTING WITH WEBCAM**

The cascade classifier is loaded and for capturing images, video capture(0) webcam is used . Two dictionaries are created such as labels\_dict which gives the label ‘MASK’ for 0 and ‘NO MASK’ label for 1. The second dictionary color\_dict gives green colour for faces with mask and red for faces without mask. Then a for-loop is created to read the frames one by one from camera and further process like converting into gray scale is done as in the training data. The faces are detected and the region of interest are cropped, resized, normalized, and finally reshaped into 4-D. The reshaped image will be predicted and saved in result. The result gives probability of having mask and not having mask. The argmax function returns the column number with maximum probability. Hence output is obtained.

