**PROJECT REPORT**

**Project Title:** Face Mask Detection

**Group Members:**

ADDAGIRI NAGA RAKESH BT17GCS003 D3

CHENNUPATI SAI REVANTH BT17GCS152 D3

HITESH PARA BT17GCS154 D2

YENDURI SASANK BT17GCS128 D2

**Introduction:**

The Corona Virus has affected the world very seriously. One major protection from that virus is to wear a Face Mask. Due to this unimaginable situation wearing the Face Mask has been a new norm and compulsory. At present there are people assigned to check if everyone is wearing a face mask or not. But with our neural network model it can detect if people are wearing a mask or not in a Live video stream. For that we designed an efficient and accurate neural network model that can detect if a person is wearing face mask or not.

**Motivation:**

Many people are neglecting to wear a face mask. So, we decided to make a face mask detector which can detect if a person is wearing a mask or not.

**Work done:**

Our Project is about building a neural network model for building a Face mask detector. We divided the project into four phases and worked accordingly.

1. *Initiation*
2. *Data Pre-Processing*
3. *Training the Model*
4. *Testing the Model*

***1.Initiation:***

In the Initiation Phase, we gathered a lot of information about our project. We used Python programming language and mostly we used PyCharm IDE for training, testing and execution of our model. We developed the model using Keras, Tensorflow, MobileNet and OpenCV.

We read a Research paper and understood the concepts of applying the neural networks in real life. We read about different classifiers and finalized Mobile-Net for our project.

***2. Data Pre-Processing:***

Firstly, we collected lot of images of people having mask and without mast from different sources. After we converted all images into arrays to create a Model. And converting the size of all the images into 224 x 224 so that our model is perfect in that shape and we done this by using “load\_image” function from “Keras.prepocessing.image”. And we used Data augmentation techniques like flipping the image to multiply the images in the dataset. Converting the dataset into With-Mask and without Mask and categorizing them 0 and 1. Splitting the data into testing and training. And we used test\_size=0.20, which means 20% of images in the dataset will go to testing set and the remaining will go into training set.

***3.Training the Model:***

We used the gathered dataset for training the model. We used AveragePooling2D, Flatten, Dense and Dropout as the layers in our neural network model. In the last Dense layer, we use the *S*oftMax function to output a vector that gives the probability of each of the two classes.

For training we finalized into Resnet and Mobile net and we found out that Resnet is providing more accuracy and Mobile net gives less when compared to Resnet. But the accuracy of Mobile net will increase as we train and also it mostly used for detection. Whereas Resnet can be used to make anything more complex model as it can handle a lot of layers. Mobile Net is to use to build lighter deep neural networks. So, we are using Mobile net for training the neural network model. We trained our model for 20 Epochs and having the learning rate 0.0001. Here we are taking very less learning late so that our loss will be calculated properly which means we will get better accuracy soon. After successfully creating a model we obtained an accuracy of around 98%. After all the 20 Epochs the neural network model will be created which will be used for mask detection. And we save this model in H5 format.

After saving this model we are going to plot our Accuracy and metrics by using Matplotlib library.

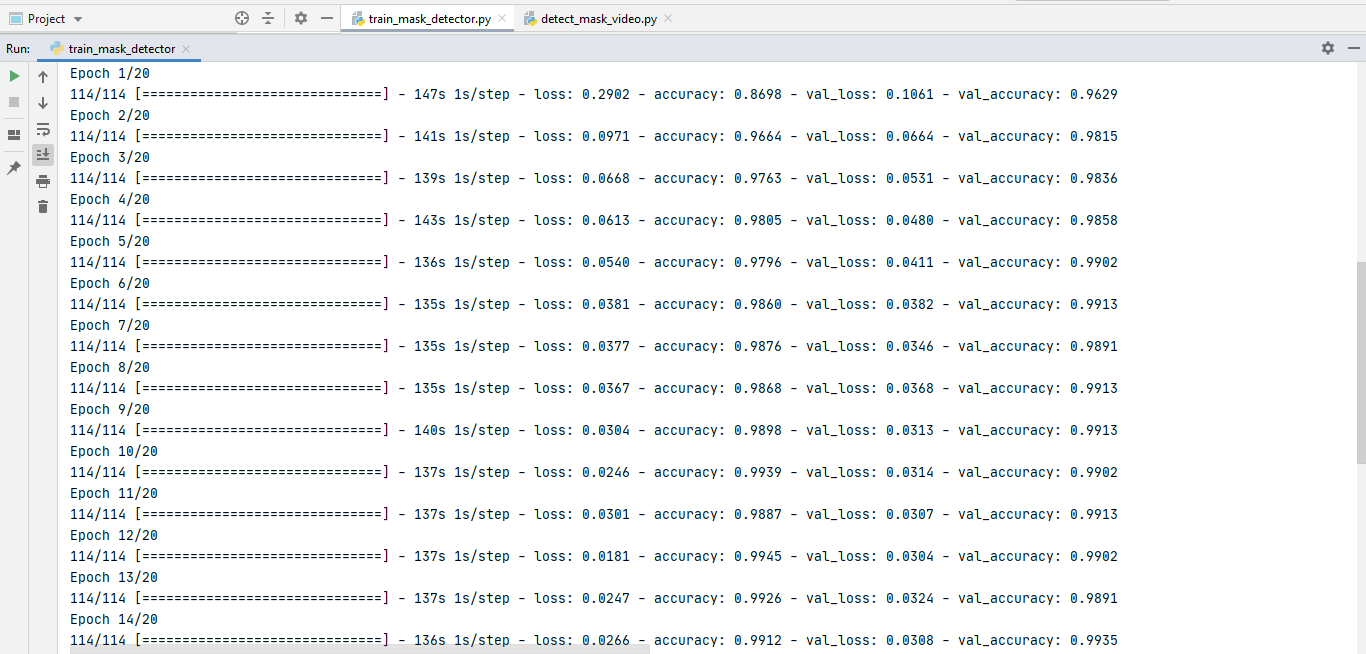
****From the diagram below we can see Training loss and Accuracy and from the graph we also conclude that our model has good accuracy and the loss is also to be reduced.

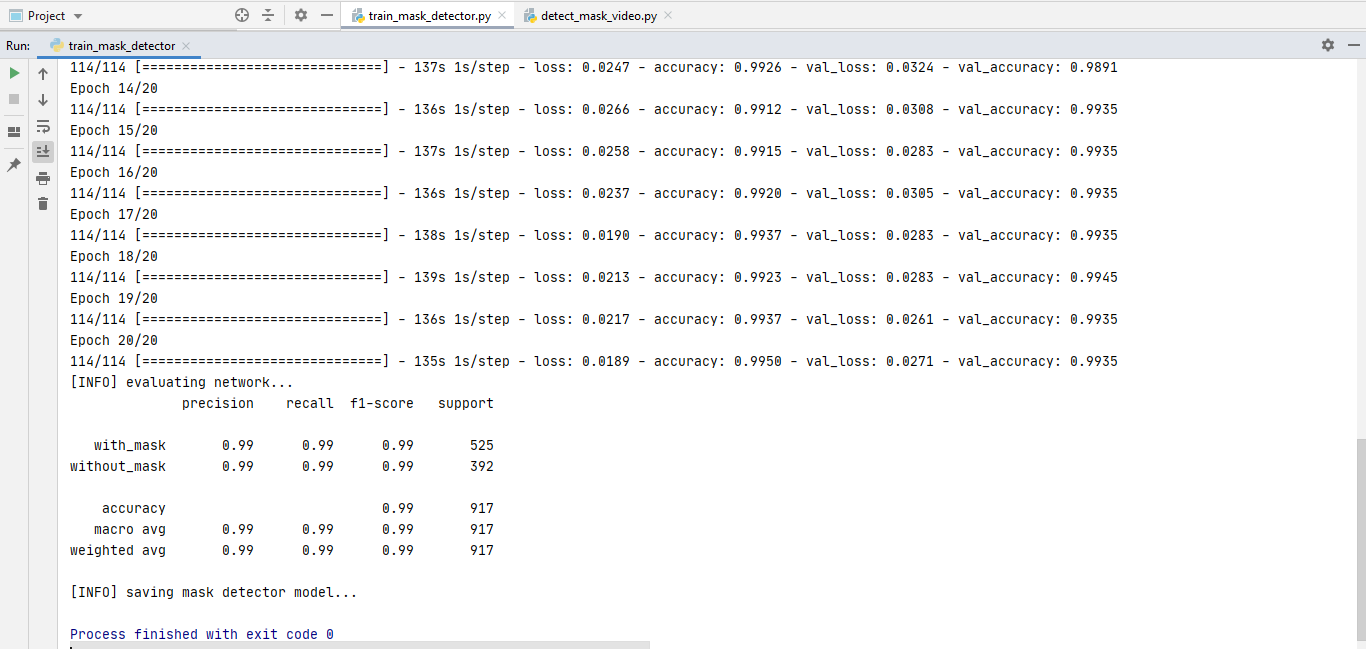
***4.Testing the model:***

In this phase we are testing the model that we have trained in the previous phase. Up to here we successfully created model for mask detector so in order to do face detection we have downloaded a couple of files and by using these files in our code we can detect faces and with the help of mask detector model we are going to detect where the person is with mask or without mask and we used OpenCV, a python library, for integrating the Live video stream into our project.

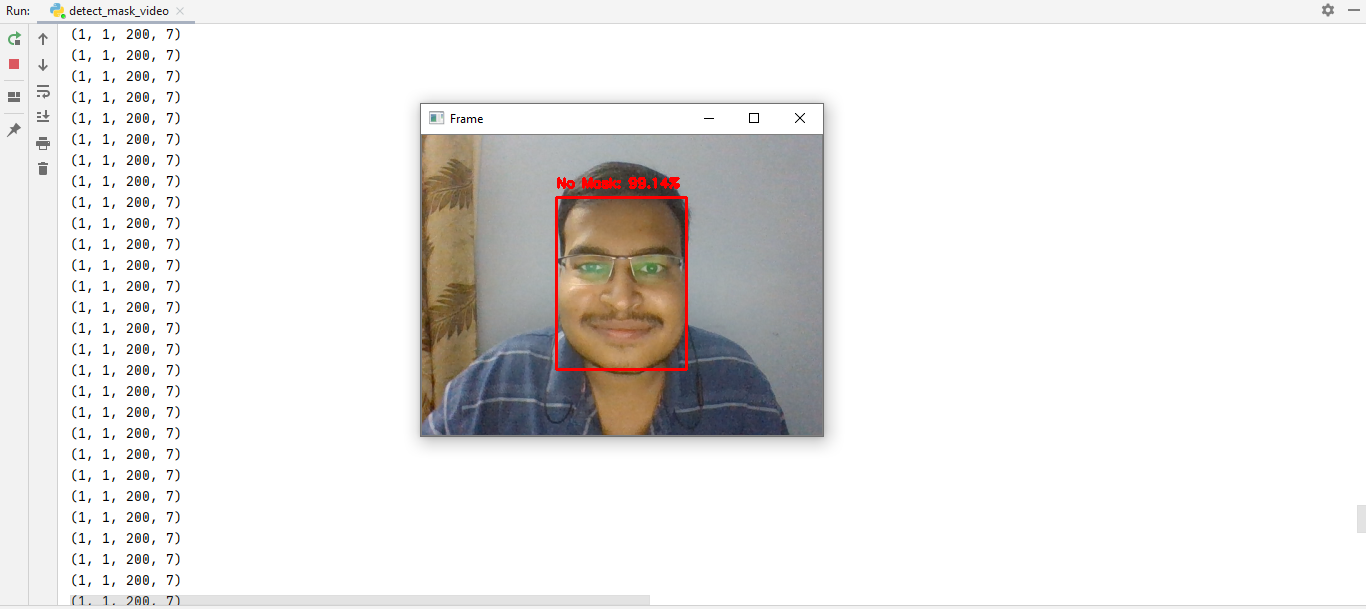
The model will predict the possibility of two class Mask and NoMask. If the person in the live video is wearing a Mask then it will detect the face of the person and shows a square in green colour with heading Mask and if the person is not wearing a mask then it will show the persons face in a red square with accuracy and No Mask

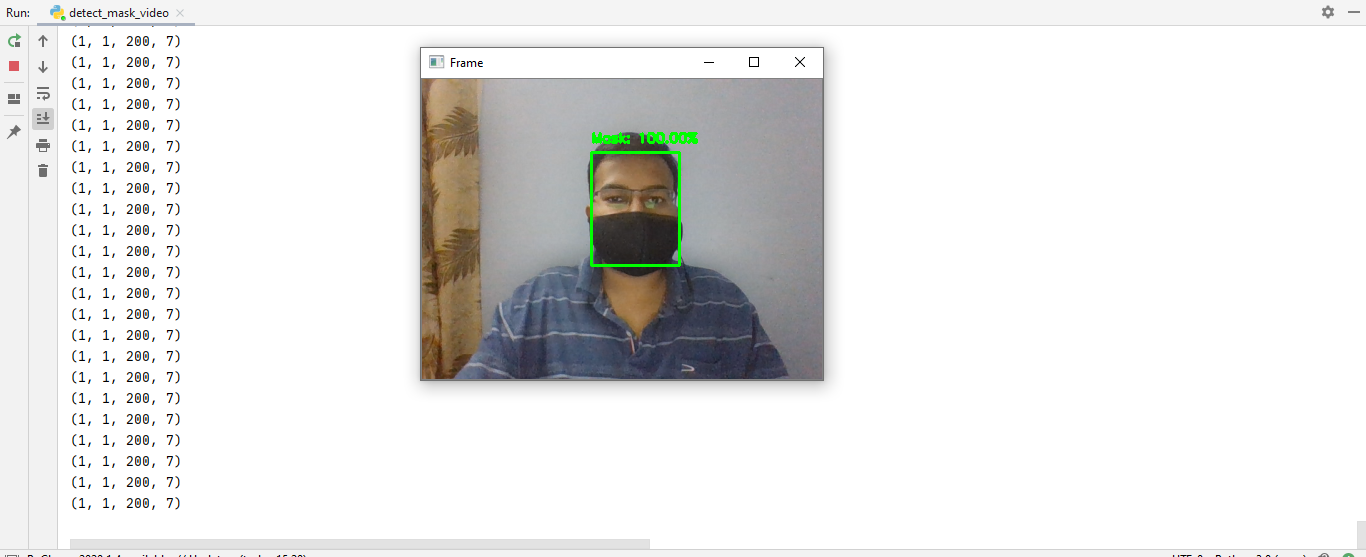
**Output:**

****



The above images depict the epochs during the training of our face mask detection model. Our model accuracy is around 99%

**No Mask**

**Mask**

The last two images show the output when we run our face detection program. If the person has mask it shows in green level with the confidence and Mask tag. Otherwise it shows a red square with the confidence and NoMask tag.

**Application of your project:**

We have developed a face mask detector on images. And in excess we also developed a model which can be applied to Live Video Streams to detect if the person in the frame are wearing a mask or not.

With some additional improvements and additions our neural network model can be integrated with CCTV to detect and identify people without masks.

**Conclusion:**

Our neural network model is developed using libraries like Keras, TensorFlow, Mobile Net and OpenCV. It uses AveragePooling2D, Flatten, Dense and Dropout as the layers in the network. It can detect and identify people without masks. Our model is accurate, and since the MobileNetV2 classifier is used, it is also computationally efficient and thus making it easier to deploy the model to embedded systems (Arduino, RaspberryPi, etc.).

So, it can also be used in real-life applications which require face mask detection for safety purposes due to Covid-19.

This model can be integrated with embedded systems in public places like Airports, Railway stations, Bus stations, Offices, Schools, Hospitals, etc.