# PROJECT REPORT

(Project Term August-December 2021)

## Face Mask Recognition

Submitted by


**Ravinshu Makkar** **Registration Number: 11908152**
**Divyanshu Tiwari** **Registration Number: 11908110**

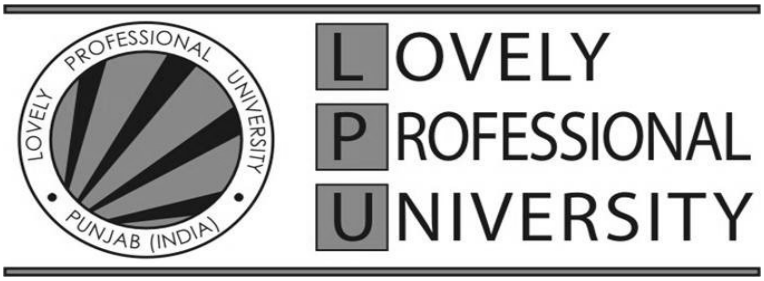

**Project Group Number 2295255**

**Course Code: INT 246**

Under the Guidance of

**Dr. Sagar Pande**

## School of Computer Science and Engineering



# DECLARATION

We hereby declare that the project work entitled “Face Mask Recognition” is an authentic record of our own work carried out as requirements of Project for the award of B. Tech degree in Computer Science & Engineering from Lovely Professional University, Phagwara, under the guidance of Dr. Sagar Pande, during August to November 2021. All the information furnished in this project report is based on our own intensive work and is genuine.

Project Group Number: 2295255

Name of Student 1: Ravinshu Makkar
Registration Number: 11908253

Name of Student 2: Divyanshu Tiwari
Registration Number: **11908110**

Date: 19/11/2021

Date: 19/11/2021

# CERTIFICATE

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. They have completed this Project under my guidance and supervision. The present work is the result of their original investigation, effort, and study. No part of the work has ever been submitted for any other degree at any University. The Project is fit for the submission and partial fulfillment of the conditions for the award of B. Tech degree in Computer Science & Engineering from Lovely Professional University, Phagwara.

**Signature and Name of the Mentor**

**Designation**

**School of Computer Science and Engineering,**
Lovely Professional University,
Phagwara, Punjab.

Date :

# ACKNOWLEDGEMENT

We would like to express my special thanks of gratitude to my teacher Dr. Sagar Pande who gave us the golden opportunity to do this wonderful project on the topic Face Mask Recognition, which also helped me in doing a lot of research and I come to know about so many things. I am thankful to them.

It helped me increase my knowledge and skills. Thanks again to all who supported.

# TABLE OF CONTENTS

Title Page……(i)
Declaration…...……..(ii)
Certificate……..…….……(iii)
Acknowledgement……...…….....(iv)
Table of Contents……...……(v)

1 Introduction **6**
1.1 Setup **6**
1.1.1 Environment **6**
1.1.2 Dataset **6**
1.2 Training **6**
1.2.1 Import the package **7**
1.2.2 Mask Loop **7**
1.2.3 Image Loop **8**
1.2.4 Model **8**
1.2.5 Accuracy **9**
1.3 Detection **9**
1.3.1 Webcam **9**
1.3.2 The Photo **10**
Links **11**

# 1. INTRODUCTION

Machine learning is a subfield of artificial intelligence (AI). The goal of machine learning generally is to understand the structure of data and fit that data into models that can be understood and utilized by people. The aim of the project is to apply Machine Leering methods in order to improve the performance of the model.

## 1.1. Setup

To accomplish this task, we have used the **Google Colab.** Colab is virtual machine that can be used to work with machine learning projects. TensorFlow, sklearn, mobilenets and many more pip are pre-installed in system of colab.

### 1.1.1. Environment

Firstly sign-in with google id in colab and open a new notebook in it. This will create environment for the project.

### 1.1.2. Dataset

Now its time for the dataset. With massive number of nearly 1900 pics with mask and 1900 pics without mask we created a folder of the dataset. We used open-source projects to accomplish this task. Website like Kaggle and google photos helped a lot in this project. When we finalised the pic for the model, we made two categories and placed pics in the right one named mask and without mask. With nearly 4000 pics we created our model.

## 1.2. Training

After the dataset the main part is to train our model and use it in our notebook. For that we created a python file named train_mask_detector.py. The process and instructions are stated below.

### 1.2.1. Import the package

First step was to import all the necessary packages in our file required to train our model. Firstly, we imported TensorFlow packages required to preprocess the images then we imported sklearn packages and mobilenets to process the image. Mobilenets package is normally accurate to process the image but it is fast so we decided to use it. Packages are shown in image 1.1.

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.layers import AveragePooling2D
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Input
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.utils import to_categorical
from sklearn.preprocessing import LabelBinarizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from imutils import paths
import matplotlib.pyplot as plt
import numpy as np
import os
```

### 1.2.2. Mask Loop

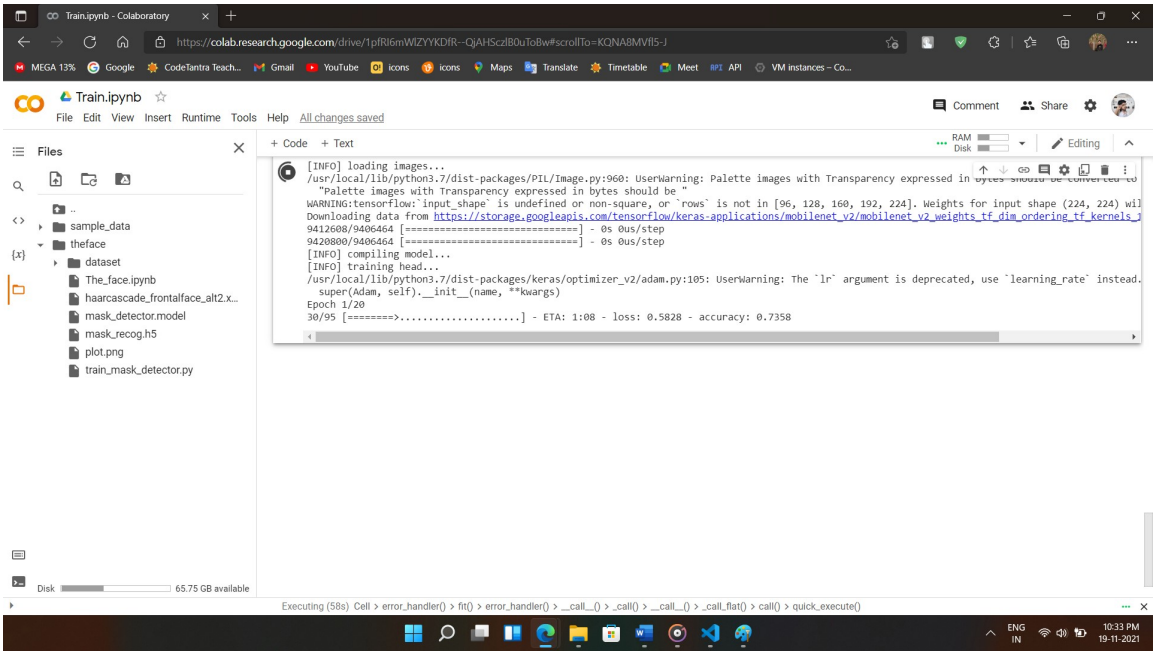
It's time to loop through each image in our dataset to create a model. For this we used the join option os.join option fulfill this requirement. Then img_to_array function was used to create all the image as array to preprocess them.

### 1.2.3. Image loop

We have many images but what if a person is walking upside down we need to train our model for more accurate results so we went with Image_Data_Genrator function to create and build different image with all the config and give more accurate results.

### 1.2.4. Model

After such hard task it took around 40-45min to create our trained model. Image 1.2 shows the starting of model creating.



### 1.2.5. Accuracy

After model was created the accuracy and loss in model are depicted as below


### 1.3. Detection

Now the code is required to use webcam and capture the pic and detect if mask is there or not. For this we use the below steps.

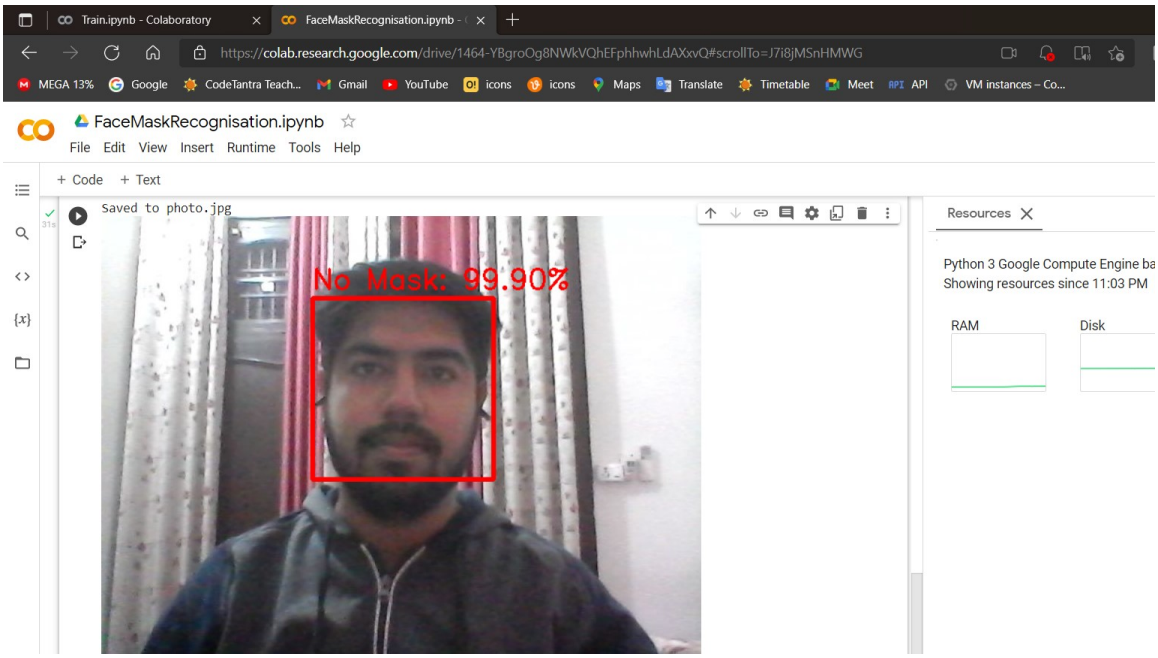
### 1.3.1. Webcam

Connecting a webcam in a virtual machine was not an easy task. But a snippet code was available to use in colab to connect to webcam of device with the virtual machine. We used the same java script with modifications to control the flow of the webcam and use frames of ours. Below image shows the javascript.

```
def take_photo(filename='photo.jpg', quality=0.8):
  js = Javascript('''
    async function takePhoto(quality) {
      const div = document.createElement('div');
      const capture = document.createElement('button');
      capture.textContent = 'Capture';
      div.appendChild(capture);

      const video = document.createElement('video');
      video.style.display = 'block';
      const stream = await navigator.mediaDevices.getUserMedia({video: true});

      document.body.appendChild(div);
      div.appendChild(video);
      video.srcObject = stream;
      await video.play();

      google.colab.output.setIframeHeight(document.documentElement.scrollHeight, true);

      await new Promise((resolve) => capture.onclick = resolve);

      const canvas = document.createElement('canvas');
      canvas.width = video.videoWidth;
      canvas.height = video.videoHeight;
      canvas.getContext('2d').drawImage(video, 0, 0);
      stream.getVideoTracks()[0].stop();
      div.remove();
      return canvas.toDataURL('image/jpeg', quality);
    }
    ''')
```

### 1.3.2. The Photo

After the webcam process we had a new task ahead of us. Now we have a photo a model and the process of detection is left to be accomplished. Now we passed the photo from open cv face detection xml and got the face detected and next we passed the photo to our trained model and fit it in the right category label with the accuracy.




Project Link: - [mravinshu/theface (github.com)]

Lab link: - [FaceMaskRecognisation.ipynb - Colaboratory (google.com)]