VVIT

VASIREDDY VENKATADRI
INSTITUTE OF TECHNOLOGY

(Autonomous)

Face Mask Detection

(Academic Year: 2020-21)

Project Guide

Mr. M. Naga Sri Harsha

Team Members

1.M. Saketh(19BQ1A05B7)

2. K. Sai Teja(19BQ1A0577)

3.K. Shravani(19BQ1A0587)

4. K. Suma Sri(19BQ1A0575)

Department of Computer Science and Engineering

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY, NAMBUR

Table of Content

Problem Statement :- This project is to identify face masks as an object in video surveillance cameras across different places like hospitals, functions, emergency departments etc to provide safety to doctors ,patients and reduce the outbreak of the disease and to provide safety for people.

Introduction:-

We will build a real-time system to detect whether the person on the webcam is wearing a mask or not. We will train the face mask detector model using Keras and OpenCV.

The face mask recognition in this study is developed with a machine learning algorithm through the image classification method: MobileNetv2. MobileNetV2 is a method based on Convolutional Neural Network (CNN) that developed by Google with improved performance and enhancement to be more efficient. This project was done by considering two original datasets. The dataset was taken from the Kaggle dataset and the Real-World Masked Face dataset (RMFD); used for the training, validation, and testing phase so the model can be implemented .

Software Requirements :-

OS :-Linux/Unix/MacOS/Windows(or any other).
Code Editor : Notepad/Visual Studio Code.
Languages used :- Python 3.9.2

Hardware Requirements :-

Webcam
RAM :- 4GB(or above)
Processor :- i5(or i7 recommended)

Implementation:

Modules used :

- **TensorFlow:** The TensorFlow object detection API is the framework for creating a deep learning network that solves object detection problems.
  TensorFlow Graphics aims at making useful graphics functions widely accessible to the community by providing a set of differentiable graphics layers (e.g. cameras, reflectance models, mesh convolutions) and 3D viewer functionalities that can be used in your machine learning models of choice.

- **Keras :** Keras is an Open Source Neural Network library written in Python that runs on top of Theano or Tensorflow. It is designed to be modular, fast and easy to use. Keras is high-level API wrapper for the low-level API, capable of running on top of TensorFlow, CNTK, or Theano. Keras High-Level API handles the way we make models, defining layers, or set up multiple input-output models. In this level, Keras also compiles our model with loss and optimizer functions, training process with fit function. Keras in Python doesn't handle Low-Level API such as making the computational graph, making tensors or other variables because it has been handled by the "backend" engine.

- **Numpy :** NumPy is a module for Python. The name is an acronym for "Numeric Python" or "Numerical Python". It is an extension module for Python, mostly written in C. This makes sure that the precompiled mathematical and numerical functions.
- Furthermore, NumPy enriches the programming language Python with powerful data structures, implementing multi-dimensional arrays and matrices. These data structures guarantee efficient calculations with matrices and arrays. The implementation is even aiming at huge matrices and arrays, better know under the heading of "big data". Besides that the module supplies a large library of high-level mathematical functions to operate on these matrices and arrays.

- **Imutils :** Imutils are a series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, and displaying Matplotlib images easier with OpenCV and both Python 2.7 and Python 3.

- **Matplotlib :** Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack.

- **Open CV :** OpenCV is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection.

- **OS :** The OS module in Python provides functions for interacting with the operating system. OS comes under Python’s standard utility modules. This module provides a portable way of using operating system dependent functionality. The *os* and *os.path* modules include many functions to interact with the file system.

- **Scipy :** SciPy, a scientific library for Python is an open source, BSD-licensed library for mathematics, science and engineering. The SciPy library depends on NumPy, which provides convenient and fast N-dimensional array manipulation. The main reason for building the SciPy library is that, it should work with NumPy arrays.

Note: To install these modules you should type the following command in command prompt.

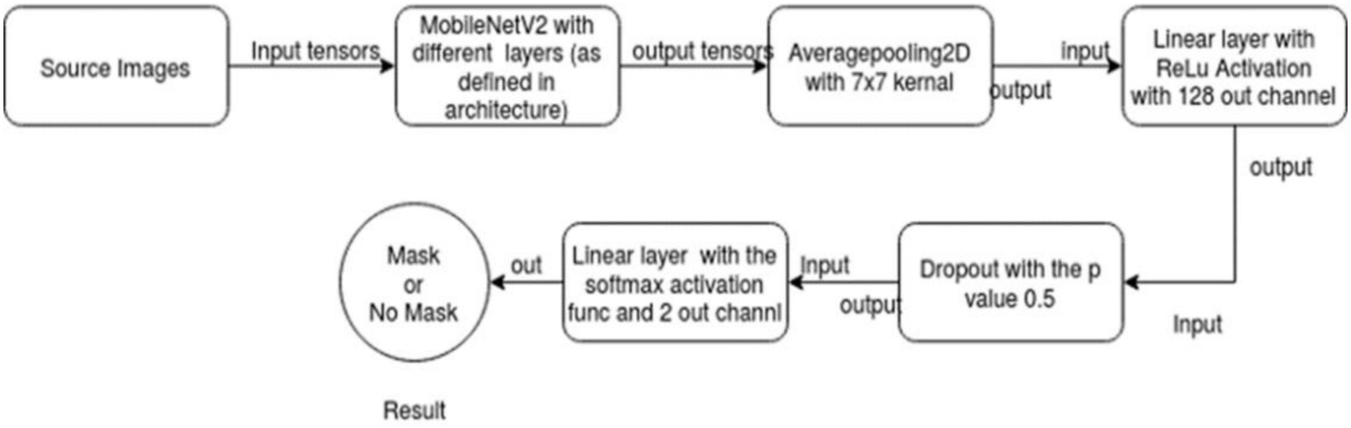
```
pip  install (module name)
```

Example: For installing tensorflow you should type the following command in your command prompt window.

```
pip install tensorflow
```

Flow Chart:

Final Network Model Architecture / Flow


Source Images
Input tensors
MobileNetV2 with different layers (as defined in architecture)
output tensors
Averagepooling2D with 7x7 kernal
input
output
Linear layer with ReLu Activation with 128 out channel
output
Input
Dropout with the p value 0.5
Input
output
Linear layer with the softmax activation func and 2 out channl
out
Mask or No Mask
Result

Process for Face Mask Detection :

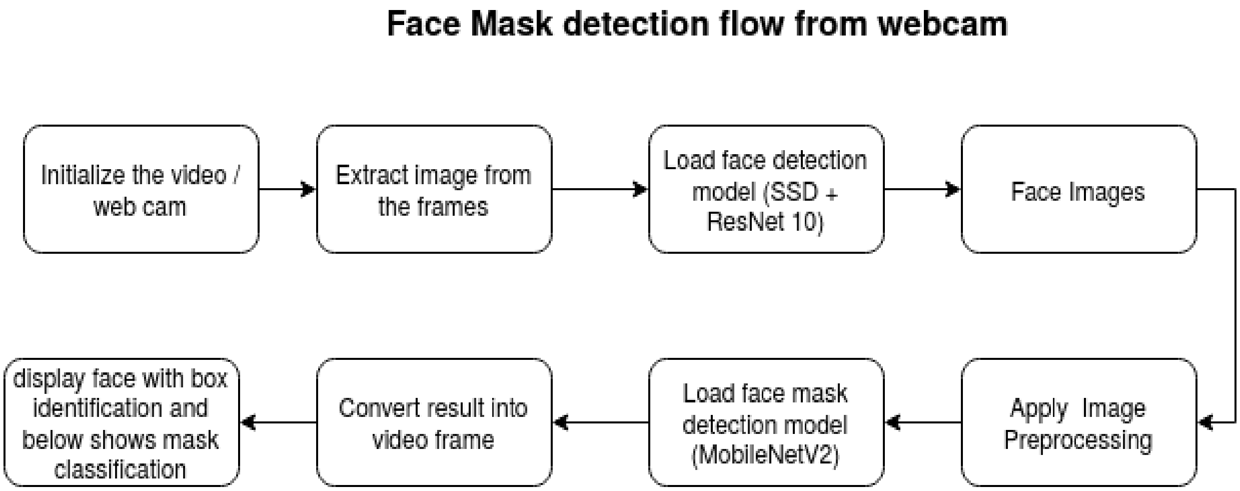
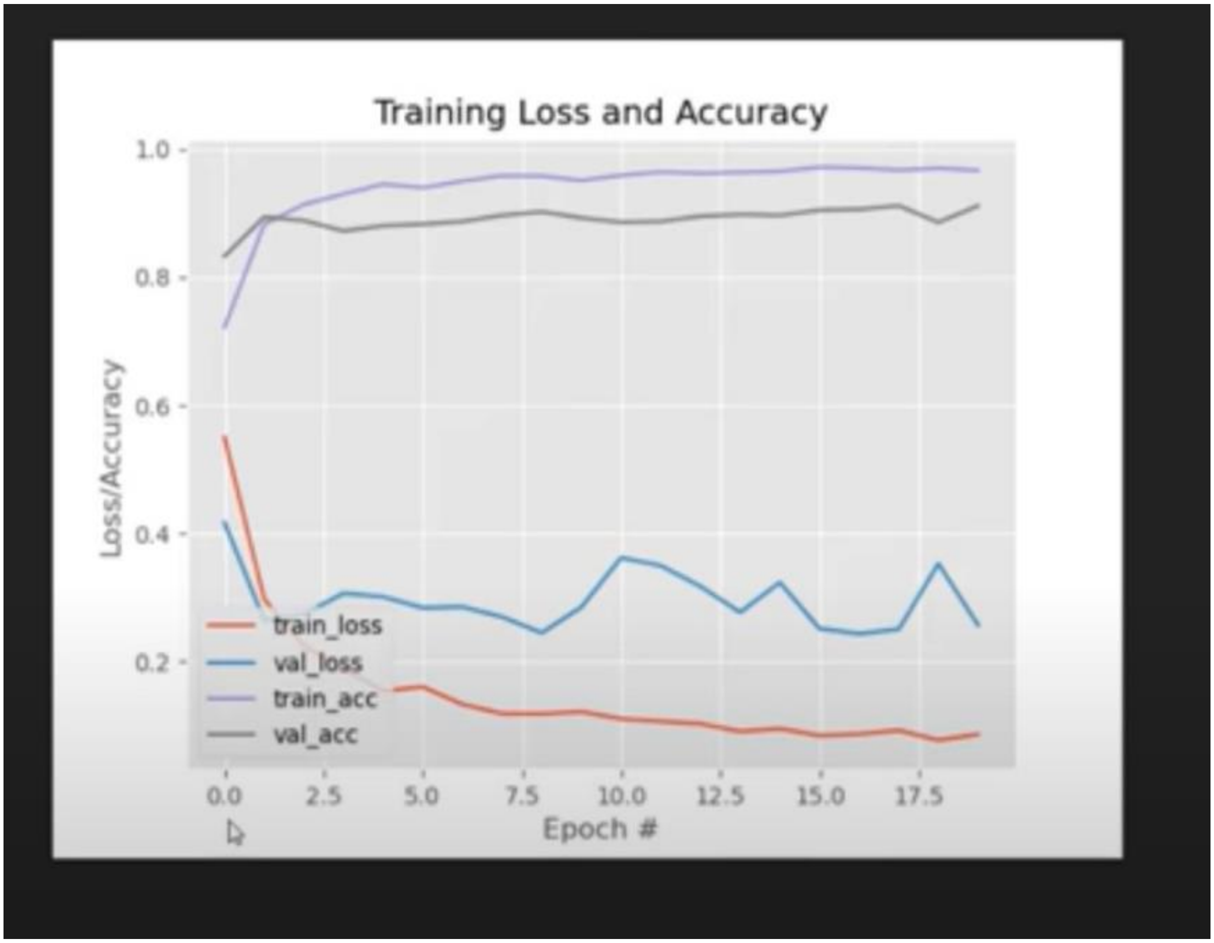

Face Mask detection flow from webcam
Initialize the video / web cam
Extract image from the frames
Load face detection model (SSD + ResNet 10)
Face Images
Apply Image Preprocessing
Load face mask detection model (MobileNetV2)
Convert result into video frame
display face with box identification and below shows mask classification

Image Classification :

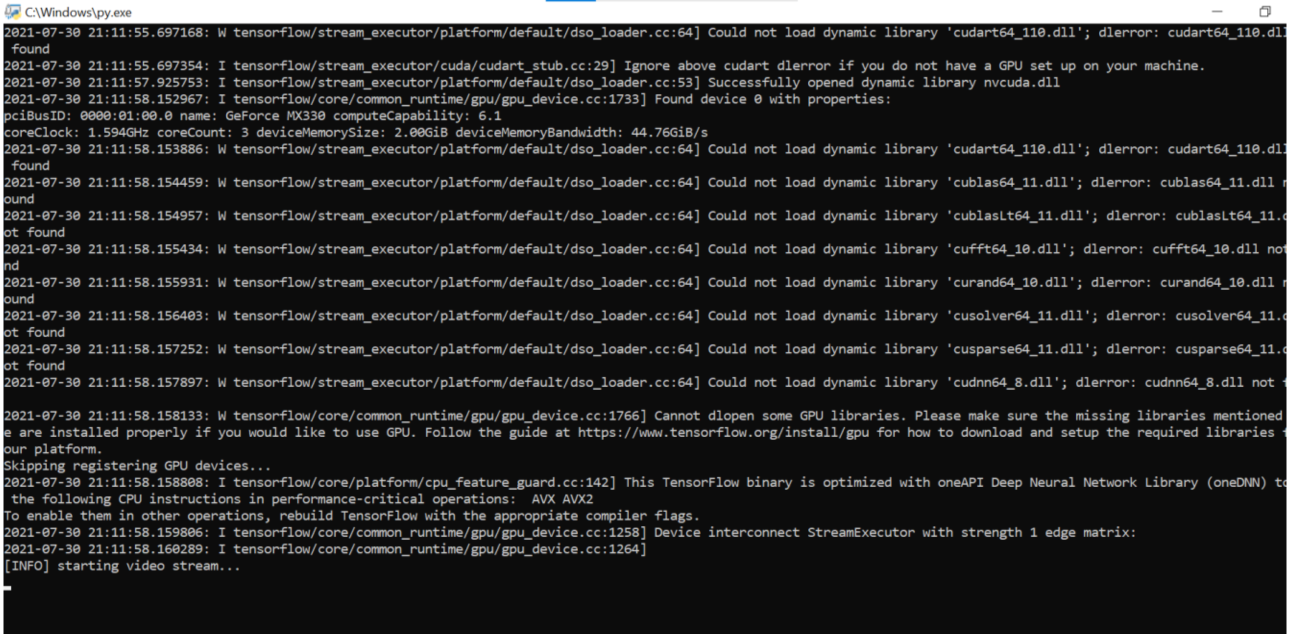
Training and Accuracy loss Graph:



The above graph shows how efficiently the model works. The graph is plotted once after the training process is finished.

After training and testing the model the graph is plotted using numpy module in python.

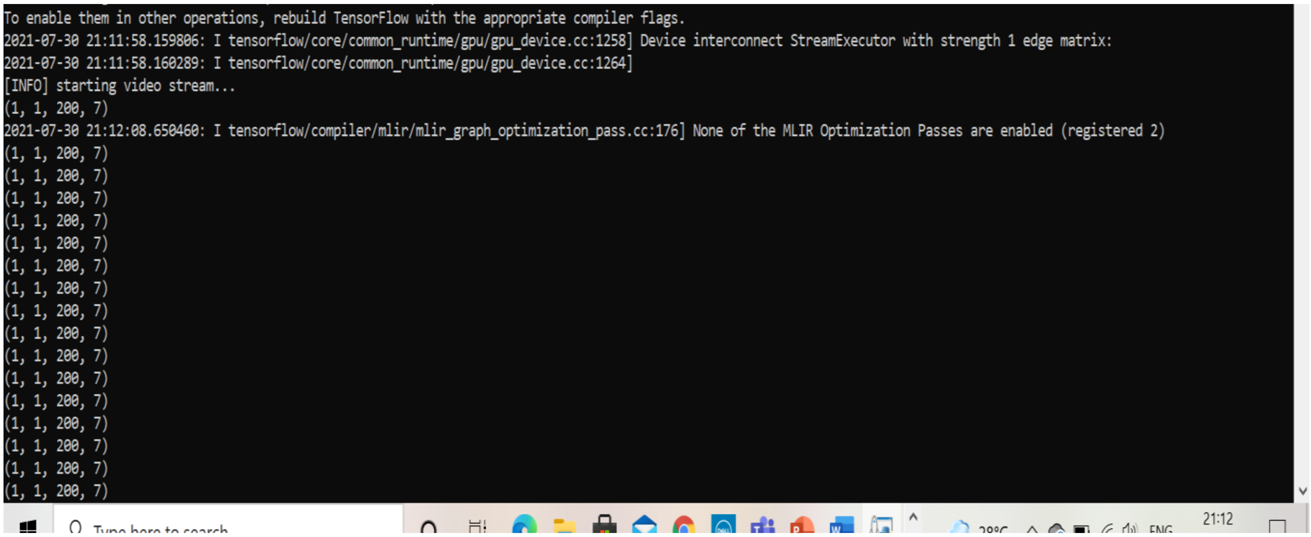
Result: After executing our file in the command prompt for detect mask video the execution process is as shown below:



```
2021-07-30 21:11:55.697168: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll
 found
2021-07-30 21:11:55.697354: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2021-07-30 21:11:57.925753: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library nvcuda.dll
2021-07-30 21:11:58.152967: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1733] Found device 0 with properties:
pciBusID: 0000:01:00.0 name: GeForce MX330 computeCapability: 6.1
coreClock: 1.594GHz coreCount: 3 deviceMemorySize: 2.00GiB deviceMemoryBandwidth: 44.76GiB/s
2021-07-30 21:11:58.153886: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll
 found
2021-07-30 21:11:58.154459: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cublas64_11.dll'; dlerror: cublas64_11.dll n
ound
2021-07-30 21:11:58.154957: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cublasLt64_11.dll'; dlerror: cublasLt64_11.d
ot found
2021-07-30 21:11:58.155434: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cufft64_10.dll'; dlerror: cufft64_10.dll not
nd
2021-07-30 21:11:58.155931: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'curand64_10.dll'; dlerror: curand64_10.dll n
ound
2021-07-30 21:11:58.156403: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cusolver64_11.dll'; dlerror: cusolver64_11.d
ot found
2021-07-30 21:11:58.157252: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cusparse64_11.dll'; dlerror: cusparse64_11.d
ot found
2021-07-30 21:11:58.157897: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudnn64_8.dll'; dlerror: cudnn64_8.dll not f

2021-07-30 21:11:58.158133: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1766] Cannot dlopen some GPU libraries. Please make sure the missing libraries mentioned
e are installed properly if you would like to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how to download and setup the required libraries f
our platform.
Skipping registering GPU devices...
2021-07-30 21:11:58.158808: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to
 the following CPU instructions in performance-critical operations:  AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2021-07-30 21:11:58.159806: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1258] Device interconnect StreamExecutor with strength 1 edge matrix:
2021-07-30 21:11:58.160289: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1264]
[INFO] starting video stream...
```

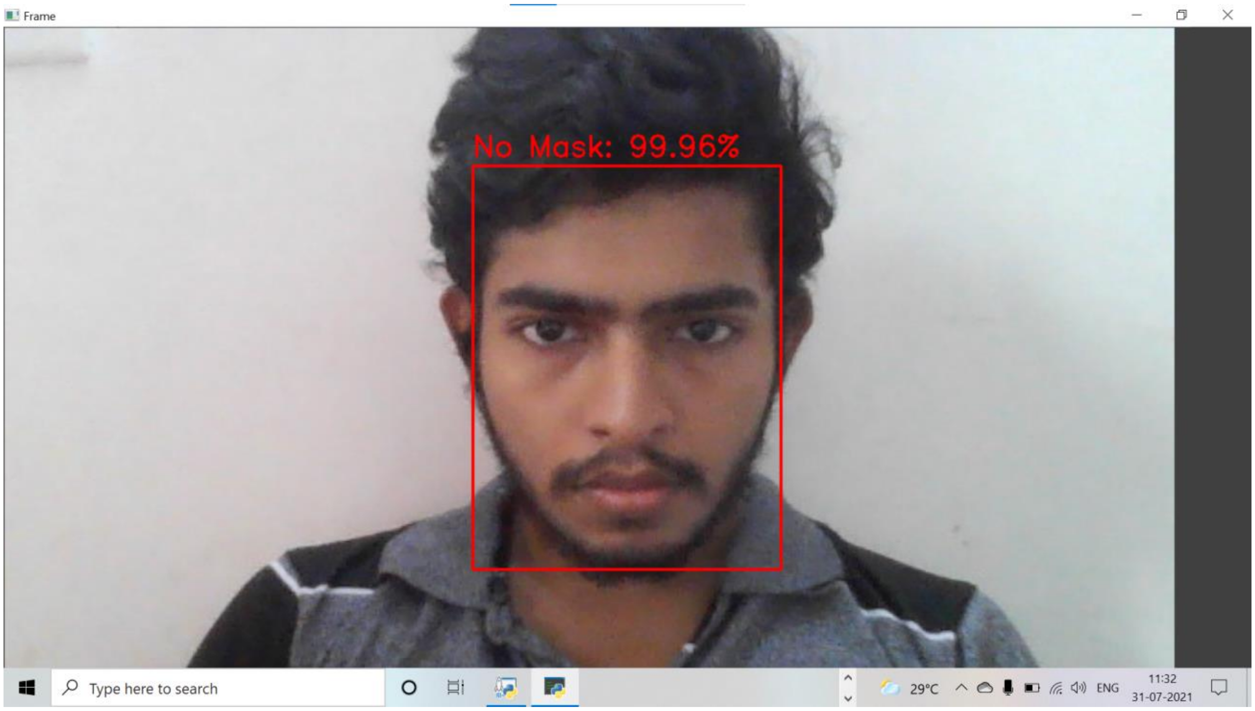
Before starting the video stream the model checks whether the processor is suitable for execution or not. After all requirements are satisfied then it will redirect to a frame then our webcam will on automatically showing faces along with the tags contained in rectangular boxes with or without mask.



```
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2021-07-30 21:11:58.159806: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1258] Device interconnect StreamExecutor with strength 1 edge matrix:
2021-07-30 21:11:58.160289: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1264]
[INFO] starting video stream...
(1, 1, 200, 7)
2021-07-30 21:12:08.650460: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:176] None of the MLIR Optimization Passes are enabled (registered 2)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
```

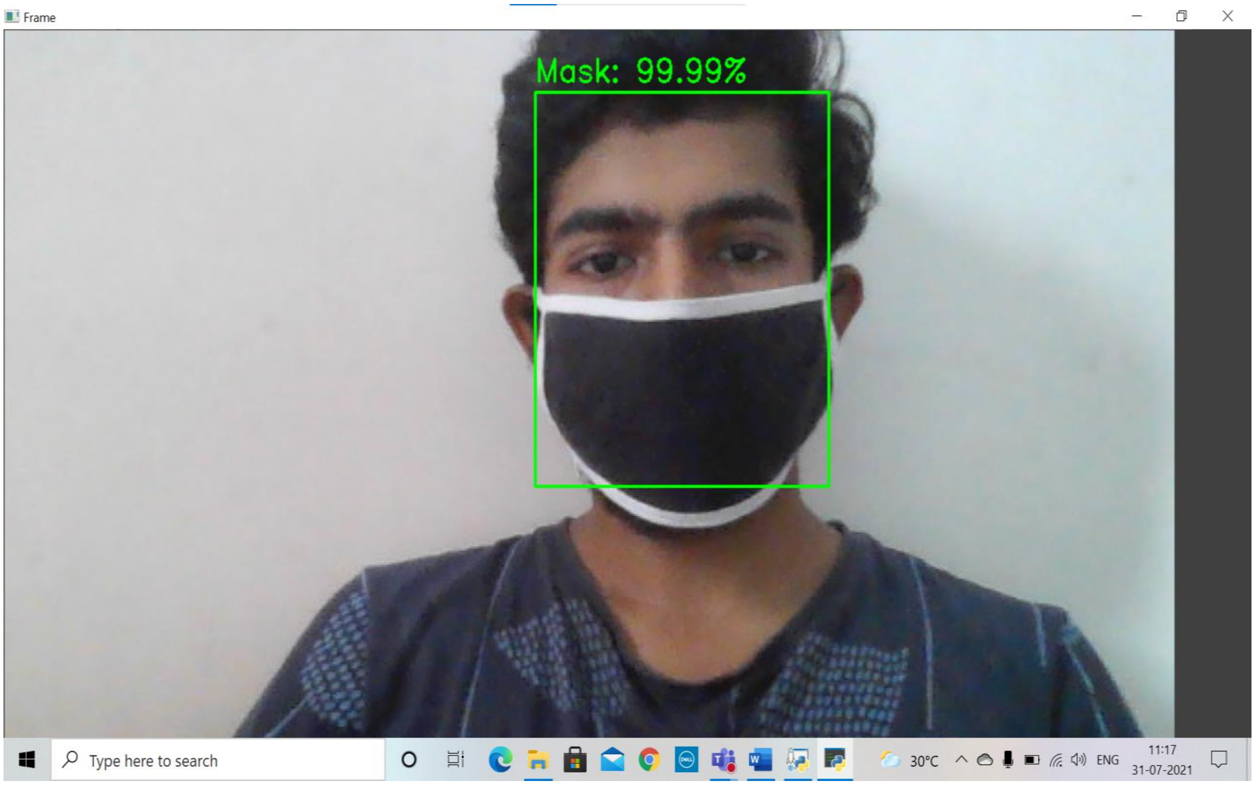
The frame is shown in two cases:

Without Mask:



- The webcam captures all the faces recorded in the frame and the model draws bounding boxes around all the faces and using MobilenetV2 algorithm the faces are classified.
- As shown above the model predicts the accuracy of the person wearing a mask.
- If he is not wearing a mask then the bounding box will be red in color with a label “No Mask” along with the percentage of accuracy.

With Mask:



- **When the person wears a mask then the bounding box around the face will be green in color and displays a label “Mask” along with the percentage of accuracy.**