**FACE MASK DETECTION**

**A PROJECT REPORT**

###### ***Submitted by***

##### YOGANANDH(17113071)

##### MANOJ B (17113083)

##### J.S.SHIVANANDHAM (17113122)

**Under the guidance of**

DR. P.RANJANA

PROFESSOR, CSE

***In partial fulfillment for the award of the degree***

***Of***

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**



**HINDUSTAN INSTITUTE OF TECHNOLOGY AND SCIENCE**

**CHENNAI - 603 103**

MAY 2020



**BONAFIDE CERTIFICATE**

Certified that this project report **FACE MASK DETECTION** is the bonafide work of **YOGANANDH(17113071) MANOJ B(17113082) J.S.SHIVANANDHAM (17113122)** who carried out the project work under my supervision during the academic year 2**020-2021.**

DR. ANGELINA GEETHA DR. P.RANJANA

**PROF&HEAD SUPERVISOR**

DEPARTMENT OF CSE DEPARTMENT OF CSE

|  |  |
| --- | --- |
| **INTERNAL EXAMINER** | **EXTERNAL EXAMINER** |
| Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Designation: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Designation:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Project Viva - voce conducted on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TITLE** | **PAGE NO** |
| **1** | **Introduction** | **9** |
| **2** | **Literature review** | **12** |
| **3** | **Preprocessing** | **13** |
| **4** | **Face Mask Detection: The Process** | **17** |
|  | **4.1 Introduction** | **17** |
|  | **4.2 Modules** | **18** |
|  | **4.3 Algorithm (SSD)** | **22** |
|  | **4.4 Efficiency & Accuracy** | **24** |
| **5** | **System Requirements** | **25** |
|  | **5.1 Hardware Requirements** | **25** |
|  | **5.2 Software Requirements** | **25** |
| **6** | **Conclusion & Futurework** | **26** |
|  | **References** | **27** |
|  | **Appendix** | **28** |
|  |  |  |
|  |  |  |

**TABLE OF CONTENT**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **TITLE** | **PAGE NO** |
| **3.1** | **Tensorflow Dimensions** | **13** |
| **4.2.** | **Modules** | **16** |
| **4.2.1** | **Training Dataset** | **17** |
| **4.2.2** | **Face and Feature Detection** | **18** |
| **4.2.3** | **Negative Output** | **19** |
| **4.2.3.1** | **Positive Output** | **20** |
| **4.3** | **Architecture of SSD** | **21** |
| **4.4** | **PR Curve** | **23** |
| **4.4.1** | **Rate of Rendering Frames** | **24** |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE NO** | **TITLE** | **PAGE NO** |
| **2.1** | **Literature Review** | **10** |

**ACKNOWLEDGEMENT**

First and foremost, we would like to thank **ALMIGHTY** who has provided us the strength to do justice to our work and contribute our best to it. We wish to express our deep sense of gratitude from the bottom of our heart to our guide **Dr. P.Ranjana, Associate Professor**, **Computer Science and Engineering,** for his motivating discussions, overwhelming suggestions, ingenious encouragement, invaluable supervision, and exemplary guidance throughout this project work.

We would like to extend our heartfelt gratitude to **Dr. Angelina Geetha, Professor& Head, Department of Computer Science and Engineering** for her valuable suggestions and support in successfully completing the project.

We wish to thank our Project Coordinator and Panel members for keeping our project in the right track. We would like to thank all the teaching, technical and non-technical staff of Department of Computer Science and Engineering for their courteous assistance.

We thank the management of **HINDUSTAN INSTITUTE OF TECHNOLOGY AND SCIENCE** for providing us the necessary facilities and support required for the successful completion of the project.

As a final word, we would like to thank each and every individual who have been a source of support and encouragement and helped us to achieve our goal and complete our project work successfully.

**ABSTRACT**

Face mask detection is a tool that is developed using ML which can classify between people that are wearing face masks and those who are not with the feed given to the model. The idea is to provide a solution for a worldwide pandemic safety measure absence.

We are trying to provide a simple but effective solution that can ensure a safety measure to the public at any common place. This proposed Detector is a one-stage detector focused on detecting face masks alone. The use of Tensorflow and Computer vision libraries were inevitable in this project. Numpy was used for the image processing.

Basically, a live video feed is processed to find a face after which that face is cut out to be processed and later checked for the existence of a face mask. If positive a green box appears around that face else a red box appears and notification is sent to the appropriate officials about the presence of masks or lack thereof .

This is a simple but effective solution that can be integrated with several other technologies and provides excellent accuracy percentage of outputs in the quickest time.

**CHAPTER 1**

**INTRODUCTION**

**Introduction: -** COVID-19 has been a life changing event is everyone’s life in the entire world. It has caused a global pandemic that has forced people into their homes. This disease has taken a huge toll of human lives around the estimate of 1.4 Million lives. The total number of cases around the world has surged to to 59 million. Although the death percentile is relatively low , the physical effects of this virus are drastic even if it is not fatal. This virus has been found out be respiratory and hence in order to protect themselves and others , the people have been advised to wear face masks at all times. This had made the face mask an essential commodity without which one cannot live healthy. Failing to wear a mask is considered a serious offence punishable by fines and escort out of premises.

There is no efficient face mask detection applications which are now in high demand for transportation means, densely populated areas, residential districts,

large-scale manufacturers and other enterprises to ensure safety.

This system can therefore be used in real-time applications which require face-mask detection for safety purposes due to the outbreak of Covid-19.

This project can be integrated with embedded systems for application in airports, railway stations, offices, schools, and public places to ensure that public safety guidelines are followed.

## ****What**** ****is Face Detection?****

The goal of face detection is to determine if there are any faces in the image or video. If multiple faces are present, each face is enclosed by a bounding box and thus we know the location of the faces

Human faces are difficult to model as there are many variables that can change for example facial expression, orientation, lighting conditions and partial occlusions such as sunglasses, scarf, mask etc. The result of the detection gives the face location parameters and it could be required in various forms, for instance, a rectangle covering the central part of the face, eye centers or landmarks including eyes, nose and mouth corners, eyebrows, nostrils, etc.

## ****Face Detection Methods****

There are two main approaches for Face Detection:

1. Feature Base Approach
2. Image Base Approach

#### ****Feature Base Approach****

Objects are usually recognized by their unique features. There are many features in a human face, which can be recognized between a face and many other objects. It locates faces by extracting structural features like eyes, nose, mouth etc. and then uses them to detect a face. Typically, some sort of statistical classifier qualified then helpful to separate between facial and non-facial regions. In addition, human faces have particular textures which can be used to differentiate between a face and other objects. Moreover, the edge of features can help to detect the objects from the face. In the coming section, we will implement a feature-based approach by using OpenCV.

#### ****Image Base Approach****

In general, Image-based methods rely on techniques from statistical analysis and machine learning to find the relevant characteristics of face and non-face images. The learned characteristics are in the form of distribution models or discriminant functions that is consequently used for face detection. In this method, we use different algorithms such as Neural-networks, HMM, SVM, AdaBoost learning. In the coming section, we will see how we can detect faces with MTCNN or Multi-Task Cascaded Convolutional Neural Network, which is an Image-based approach of face detection

## ****Face Recognition****

Face detection and Face Recognition are often used interchangeably but these are quite different. In fact, Face detection is just part of Face Recognition.

Face recognition is a method of identifying or verifying the identity of an individual using their face. There are various algorithms that can do face recognition but their accuracy might vary. Here I am going to describe how we do face recognition using deep learning

**CHAPTER 2**

**LITERATURE REVIEW**

****

**Table 2.1 : Literature Review**

**CHAPTER 3**

**PRE-PROCESSING**

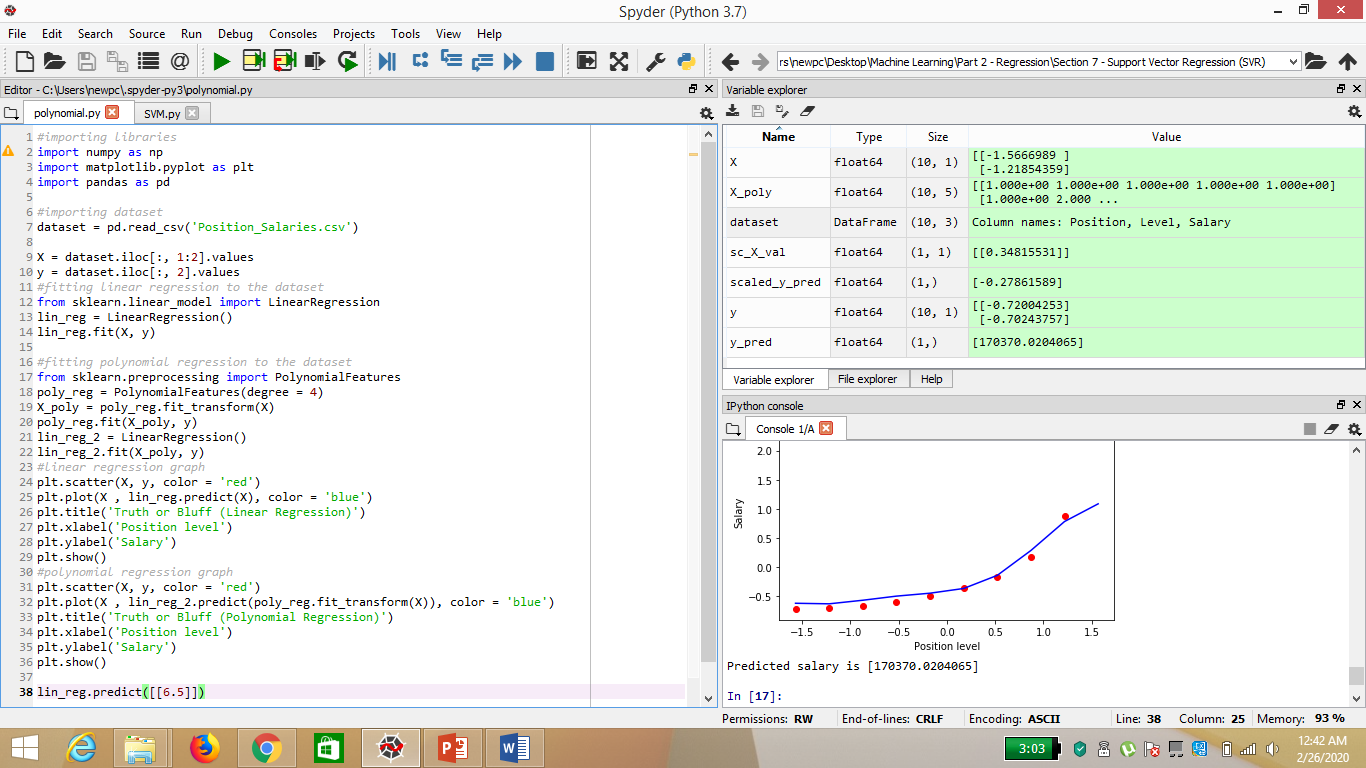
**About the Dataset used:** MAFA (MAsked FAces) is a masked face detection benchmark dataset, of which images are collected from Internet images. MAFA contains 7,811 images and 4,806 masked faces. Faces in the dataset have various various orientations and occlusion degrees, while at least one part of each face is occluded by mask. In the annotation process, each image contains at least one face occluded by various types of masks, while the six main attributes of each masked face, including locations of faces, eyes and masks, face orientation, occlusion degree and mask type.

Out of which 5,023 images were used to train the model and 1,563 were used to test the efficiency of the trained and the remaining 1,225 images were used to validate the model.

**Some of the important libraries used are:-**

**i)** NumPy

Syntax for importing this library is



* NumPy is a type of library function that is present in python language which helps in providing support for various type of matrices and large multi-dimensional arrays.
* NumPy provides various functions to the users which enables the users to perform various tasks and processes on the large multi-dimensional arrays and various types of matrices.
* NumPy makes the mathematical algorithms that run or processes slowly to run faster in python language. It does so by providing various operators and multi-dimensional array that performs smoothly and efficiently on the arrays.
* There are various axes present in NumPy. These axes are known as the dimensions.
* The rank is the total number of axes.
* Ndarray is the array of the NumPy. Ndarray in NumPy can also called as the alias array.
* NumPy can store generic data in a multi-dimensional container and has other scientific uses.
* Arrays can be created in NumPy by different methods and techniques by using important functions.

**ii)** Tensorflow :

* Syntax :

***import tensorflow as tf***

* TensorFlow is an open source machine learning framework for all developers. It is used for implementing machine learning and deep learning applications. To develop and research on fascinating ideas on artificial intelligence, Google team created TensorFlow. TensorFlow is designed in Python programming language, hence it is considered an easy to understand framework.
* Tensors are used as the basic data structures in TensorFlow language. Tensors represent the connecting edges in any flow diagram called the Data Flow Graph. Tensors are defined as multidimensional array or list.
* Tensors are identified by the following three parameters −

### Rank:

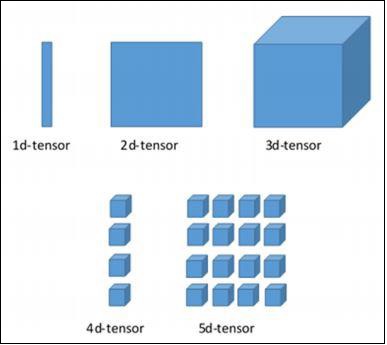
* Unit of dimensionality described within tensor is called rank. It identifies the number of dimensions of the tensor. A rank of a tensor can be described as the order or n-dimensions of a tensor defined.

### Shape:

* The number of rows and columns together define the shape of Tensor.

### Type:

* Type describes the data type assigned to Tensor’s elements.
* A user needs to consider the following activities for building a Tensor −
* Build an n-dimensional array
* Convert the n-dimensional array



3.1 Tensorflow Dimensions

**iii)** **Computer Vision :**

* 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.
* Computer Vision overlaps significantly with the following fields −

1. Image Processing − It focuses on image manipulation.
2. Pattern Recognition − It explains various techniques to classify patterns.
3. Photogrammetry − It is concerned with obtaining accurate measurements from images.

* Syntax for importing CV:

*import cv2 as cv*

**CHAPTER 4**

**Face Mask Detection – The Process**

This chapter provides the information regarding the various process involved in the salary prediction model. This also provides the detailed information regarding the regression models that have been used to predict the salary along with its graph and output for various values.

**4.1** **Introduction :** Our face mask detector didn't use any morphed masked images dataset. The model is accurate, it’s also computationally efficient and thus making it easier to deploy the model to embedded systems (Raspberry Pi, Google Coral, etc.).

This system can therefore be used in real-time applications which require face-mask detection for safety purposes due to the outbreak of Covid-19. This project can be integrated with embedded systems for application in airports, railway stations, offices, schools, and public places to ensure that public safety guidelines are followed.

**4.2 Modules :**

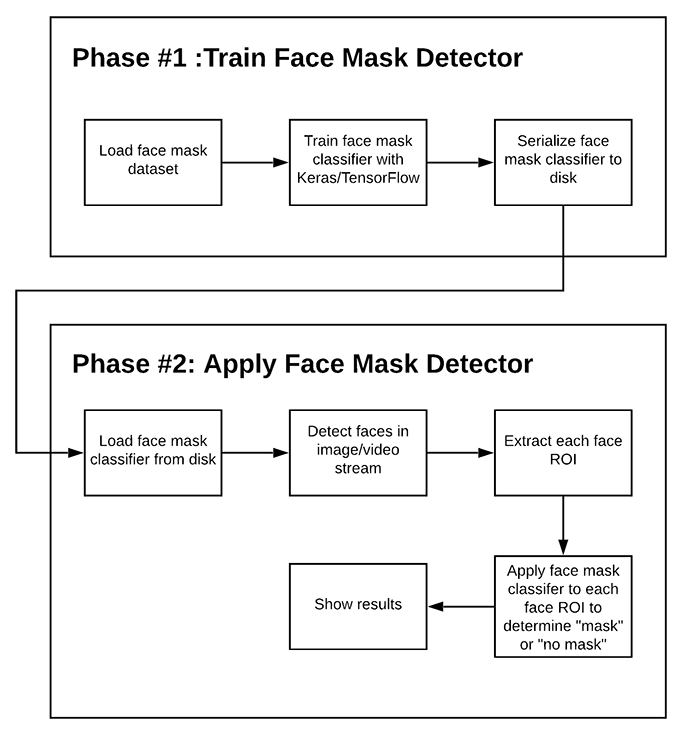


Fig 4.2. Modules of Face Mask Detection

* **Training**
* **Face Detection & Object Detection ( Face mask)**
* **Output**

**4.2.1 Training :**



Fig 4.2.1 Training Dataset

Here we’ll focus on loading our face mask detection dataset from disk, training a model (using TensorFlow) on this dataset, and then serializing the face mask detector to disk.

**4.2.2 Face and Mask Detection :**

* Facial landmarks allow us to automatically infer the location of facial structures, including:

1. Eyes
2. Eyebrows
3. Nose
4. Mouth
5. Jawline



Fig 4.2.2 Face Detection

&

Feature Detection



* If a face is detected then the model proceeds onto mask detection :

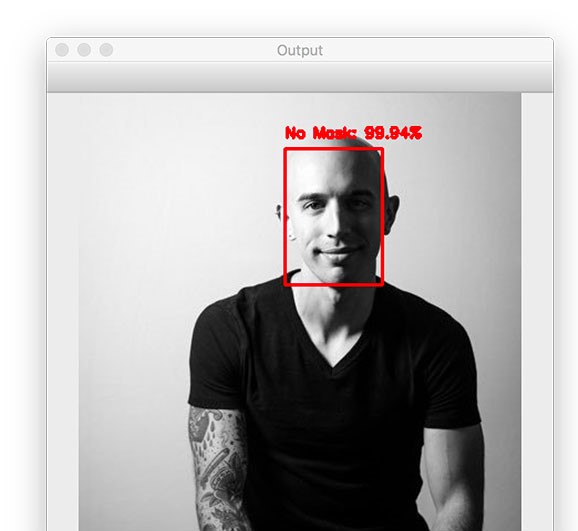
In the last step, we use the OpenCV library to run an infinite loop to use our web camera in which we detect the face using the Cascade Classifier. The code webcam = cv2.VideoCapture(0)denotes the video feed.

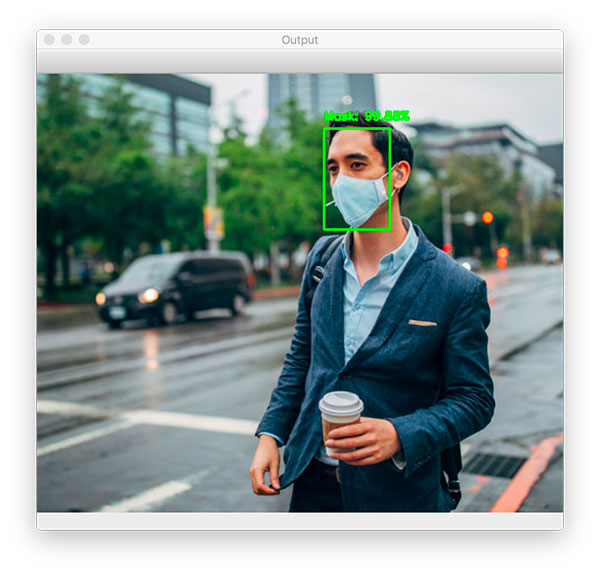
The model will predict the possibility of each of the two classes ([without\_mask, with\_mask]). Based on which probability is higher, the label will be chosen and displayed around our faces.

**4.2.3 Output of Deployment:**

* This final module gives the output which denotes if the person in the video fed to this system is wearing a mask or not. If the former is true a green box appears around that face whereas a red box appears if the case is the latter

Fig 4.2.3 Negative output

* .

****

4.2.3.1 Positive output

**4.3 Algorithm (SSD) :**

Single Shot Detection (SSD), discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. At prediction time, the network generates scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape. Additionally, the network combines predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes. Our SSD model is simple relative to methods that require object proposals because it completely eliminates proposal generation and subsequent pixel or feature resampling stage and encapsulates all computation in a single network. This makes SSD easy to train and straightforward to integrate into systems that require a detection component

## Why sliding window approach wouldn't work?

It's natural to think of building an object detection model on the top of an image classification model. Once we have a good image classifier, a simple way to detect objects is to slide a 'window' across the image and classify whether the image in that window (cropped out region of the image) is of the desired type. Sounds simple! Well, there are at least two problems:

1. How do you know the **size of the window** so that it always contains the object? Different types of objects (palm tree and swimming pool), even the same type of objects (e.g. a small building and a large buidling) can be of varying sizes as well.
2. **Aspect ratio** (the ratio of height to width of a bounding box). A lot of objects can be present in various shapes like a building footprint will have a different aspect ratio than a palm tree.

Hence the use of SSD is the best choice.

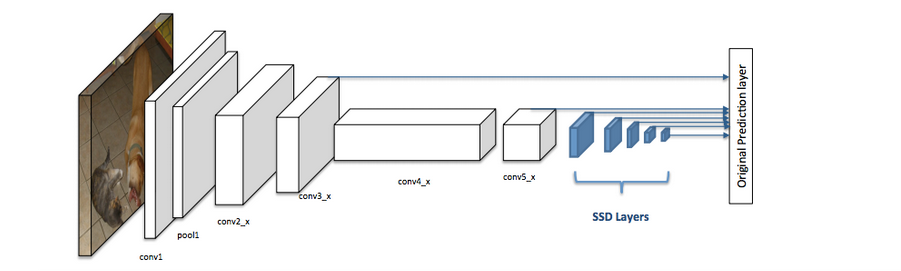


Fig 4.3 The Architecture of SSD detector

**4.4 Efficiency and Accuracy of Model :**

* A precision-recall curve (or PR Curve) is a plot of the precision (y-axis) and the recall (x-axis) for different probability thresholds.

1. **PR Curve**: Plot of Recall (x) vs Precision (y).

* A model with perfect skill is depicted as a point at a coordinate of (1,1). A skillful model is represented by a curve that bows towards a coordinate of (1,1). A no-skill classifier will be a horizontal line on the plot with a precision that is proportional to the number of positive examples in the dataset. For a balanced dataset this will be 0.5.
* The focus of the PR curve on the minority class makes it an effective diagnostic for imbalanced binary classification models.
* Calculated as: 100 % – specificity. Positive predictive value – fraction of persons with a positive test result who do have the disease. Calculated as **TP / (TP + FP).** Precision-recall curve.

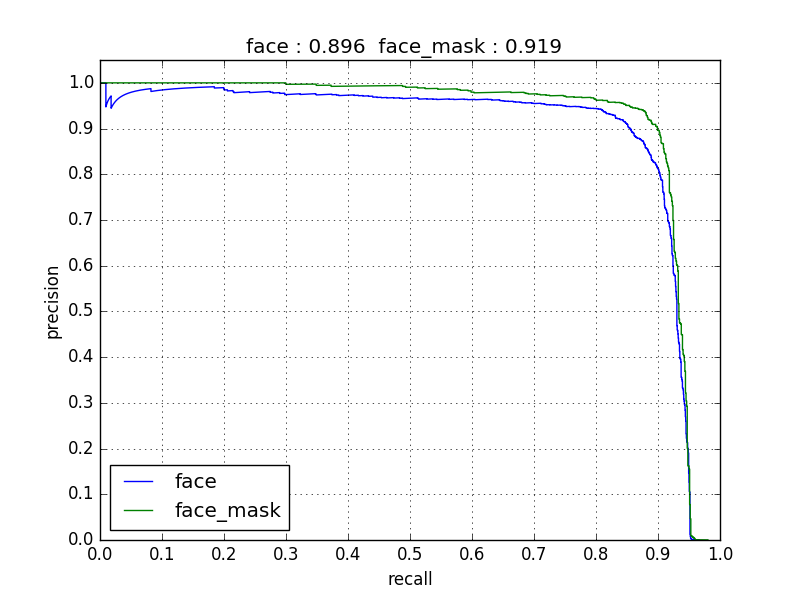


Fig 4.4 PR curve of Face Mask Detection Model

* The efficiency of detecting a face in the proposed model is at 89%
* The efficiency of detecting a face-mask in the proposed model is at 91.9%

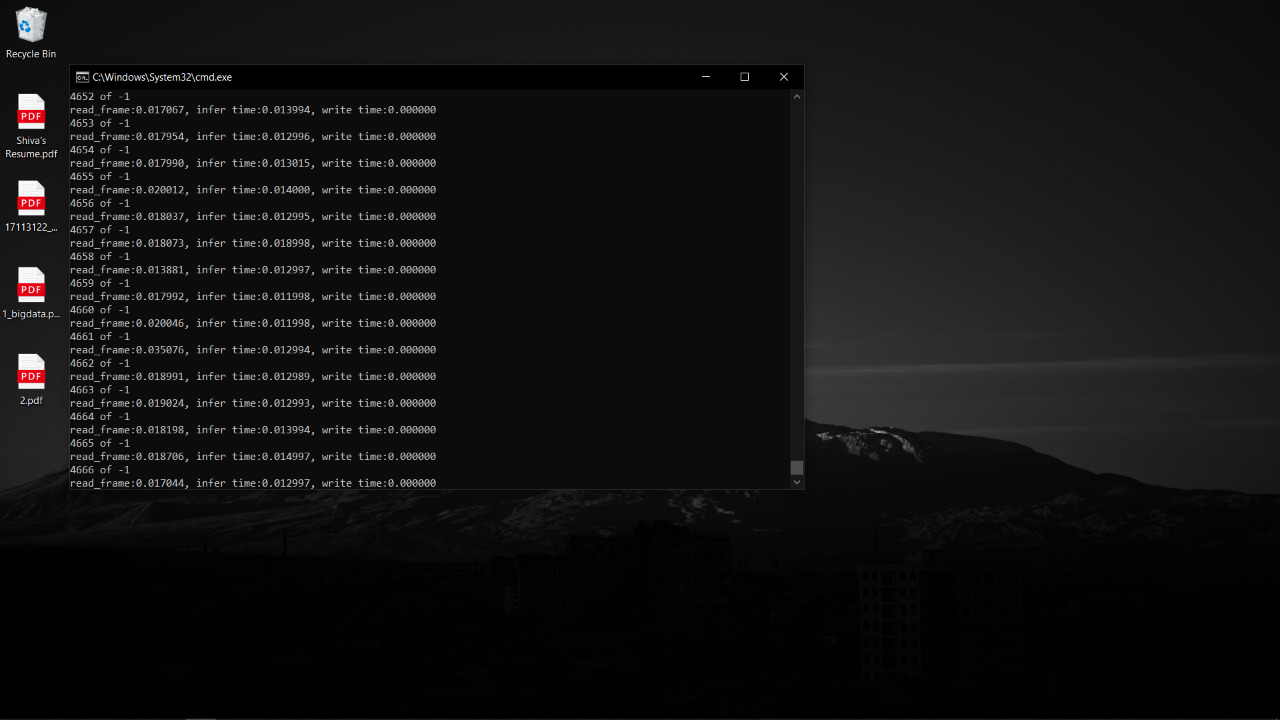


Fig 4.4.1 The speed at which frames are rendered.

* As it is evident from the figure above , the quickness of the proposed model is pretty quick. We render our frames at an average of .013 sec

**CHAPTER 5**

**SYSTEM REQUIREMENTS**

* 1. **Hardware Requirements**
* Processor : 1GHZ Dual core or more
* RAM : 4GB or more
* Internal Storage : Minimum 500MB Free Space Required
* Hardware : 1. I5/R3 or above

Radeon 570 or Nvidia 950 or above

**5.2 Software Requirements**

* Operating System : Windows, Mac OS, Linux
* Any code editor.
* Support for the python libraries mentioned.

**CHAPTER 6**

**CONCLUSION & FUTURE WORK**

**6.1 Conclusion :**

A new Face Mask Detection system has been designed and implemented in this project work. The exercise was carried out based on the need for a monitoring system to ensure the usage of face masks in a global pandemic.

The proposed system creates opportunity for:

* Reliable unmanned system to carry out a tedious task.
* Have faith in the management to ensure safety.
* Detain or send back those who fail to wear masks.
* Authorities to enforce masks or fine people who fail to.

**6.2 Future Work**

1. **Gather images of faces that may “confuse” our classifier into thinking the person is wearing a mask when in fact they are not** — potential examples include shirts wrapped around faces, bandana over the mouth, etc.
2. Improve efficiency of recognizing Faces and masks
3. Embedding the proposed system to various technologies for a wider application
4. Improve the rate at which detection is processed.

**.**

**REFERENCES**

1. Face Detection : https://www.eff.org/pages/face-detection

2 Object recognition :https://paperswithcode.com/task/object-recognition

3. Face recognition: https://realpython.com/face-recognition-with-python/

# 4. Real-time Face detection | Detection using OpenCV

https://www.mygreatlearning.com/blog/real-time-face-detection/

5.Object Detection with deep learning a review : by Q.Zhao & P.Cheng Research gate publication July 2018.

# 6. Facial Mask Detection using Semantic Segmentation by [Toshanlal Meenpal](https://ieeexplore.ieee.org/author/38232827900); [Ashutosh Balakrishnan](https://ieeexplore.ieee.org/author/37087102528); [Amit Verma](https://ieeexplore.ieee.org/author/37086841535)

# 7. How Accurate are Facial Recognition Systems – and Why Does It Matter?

https://www.csis.org/blogs/technology-policy-blog/how-accurate-are-facial-recognition-systems-%E2%80%93-and-why-does-it-matter

**APPENDIX A**

**SAMPLE SCREEN**

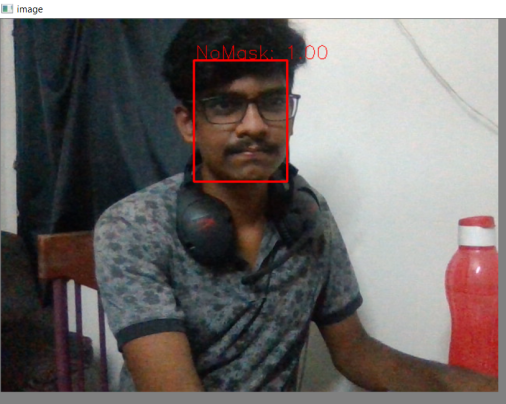
****

Fig.The output when the face is maskless in the input feed

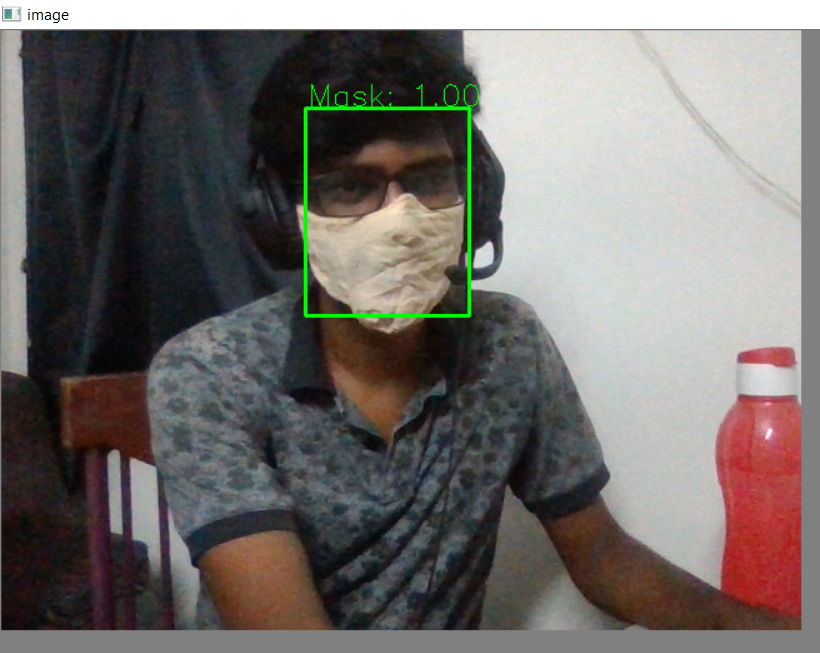
****

Fig. The output when the face is wearing a mask in the input feed