# REAL TIME FACE MASK DETECTION USING CNN

### A PROJECT REPORT

***Submitted by***

**MRITESH M 211418104324**

**VALAPARLA AKHIL TAGORE 211418104330**

**VENGISWARAN A L 211418104332**

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

**BACHELOR OF ENGINEERING**

***in***

**COMPUTER SCIENCE AND ENGINEERING**



## PANIMALAR ENGINEERING COLLEGE

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

**MAY 2022**

# PANIMALAR ENGINEERING COLLEGE

###### (An Autonomous Institution, Affiliated to Anna University, Chennai)

## BONAFIDE CERTIFICATE

Certified that this project report **“REAL TIME FACE MASK DETECTION USING CNN ”** is the bonafide work of “**MRITESH M (211418104324), VALAPARLA AKHIL TAGORE (211418104330) & VENGISWARAN AL (211418104332)”** who carried out the project work under my supervision.

**SIGNATURE SIGNATURE**

**Dr.S.MURUGAVALLI,M.E.,Ph.D., Dr.M.RAJENDIRAN,M.E.,Ph,D.,**

**HEAD OF THE DEPARTMENT PROFESSOR**

DEPARTMENT OF CSE, DEPARTMENT OF CSE,

PANIMALAR ENGINEERING COLLEGE, PANIMALAR ENGINEERING COLLEGE, NASARATHPETTAI, NASARATHPETTAI,

POONAMALLEE, POONAMALLEE,

CHENNAI-600 123. CHENNAI-600 123.

Certified that the above candidate(s) were examined in the Anna University Project Viva-Voce Examination held on...........................

**INTERNAL EXAMINER EXTERNAL EXAMINER**

## DECLARATION BY THE STUDENT

We **MRITESH M (211418104324), VALAPARLA AKHIL TAGORE (211418104330) & VENGISWARAN AL (211418104332)** hereby declare that this project reporttitled “**REAL TIME FACE MASK DETECTION USING CNN**”,under the guidance of **Dr.M.RAJENDIRAN** the orginial work doneby us and we have not plagiarized or submitted to any other degree in any university by us.

**ACKNOWLEDGEMENT**

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHI KUMAR,M.E.,Ph.D** and **Dr.SARANYASREE SAKTHI**

**KUMAR B.E.,M.B.A.,Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr.K.MANI, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. S.MURUGAVALLI,M.E.,Ph.D.,**

for the support extended throughout the project.

We would like to thank my **Project Guide Dr.M.RAJENDIRAN** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

Lastly we take the opportunity to thank all our staff members of the department of computer science and engineering . Regards to our family and friends who offered an unflinching moral support for the completion of this project.

**VALAPARLA AKHIL TAGORE**

**VENGISWARAN AL**

**MRITESH M**

**ABSTRACT**

Effective strategies to restrain the COVID-19 pandemic need high attention to mitigate negatively impacted communal health and the global economy, with the brim-full horizon yet to unfold. It is necessary to build a model that detects people with and without masks in real-time as it works as a simple precautionary measure to prevent the spread of the virus. These models based on deep convolutional neural networks are widely being developed to improve the accuracy and performance of such detection systems. The input is obtained in real-time via a webcam. A “bounding box” that displays whether the user is wearing a mask or not is displayed as the output. We will be using CNN and Deep convolutional layers to efficiently. The existing systems for face mask detection have proved to be less accurate and less efficient. They are not very effective as the dataset available for detecting masks on human faces is relatively small leading to the hard training of the model. We propose a model for face mask detection that leverages CNN to detect whether the user is wearing a face mask or not. It is different from previously existing methods as it possesses a custom GUI. After Image de-Colorization and contouring only, it creates a dataset, hence the dataset is very pure. The advantages of this system are high accuracy, fast processing, high efficiency, and better performance. If deployed correctly, this deep learning-based application helps in simplifying the work of frontline warriors and saving their lives. The real-time applications of the proposed system are that it can be used in healthcare, at crowded places like airports, railway stations as well as in multinational corporations with a workforce of more than 50 people.

**TABLEOFCONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TITLE** | **PAGE NO** |
|  | ABSTRACT | **V** |
|  | LIST OF FIGURES  LIST OF TABLES | **VI**  **IX** |
|  | **INTRODUCTION**   * 1. Introduction   2. Ideology   3. Machine Learning Technologies | **1**  **1**  **2** |
|  | **LITERATURE SURVEY** | **5** |
|  | **SYSTEM ANALYSIS**  3.1 EXISTING SYSTEM  3.2 PROPOSED SYSTEM  3.3 REQUIREMENT SPECIFICATION  3.4 PLATFORM SPECIFICATION– ANACONDA | **10**  **11**  **12**  **13**  **13** |
|  | **SYSTEM DESIGN**  4.1 USE-CASE DIAGRAM  4.2 ACTIVITY DIAGRAM  4.3 SEQUENCE DIAGRAM  4.4 DATA FLOW DIAGRAM 0  4.5 DATA FLOW DIAGRAM 1  4.6 ER DIAGRAM | **18**  **19**  **20**  **22**  **22**  **23**  **24** |
|  | **SYSTEM ARCHITECTURE** | **26** |
|  | **5.1 ARCHITECTURE DIAGRAM**  **5.2 MODULE DESCRIPTION** | **27**  **28** |
|  | 5.2.1 GUI and Datapreprocessing  5.2.2 Training Neural Networks  5.2.3 Classificatio and Output | **28**  **29**  **30** |
|  | **SYSTEM IMPLEMENTATION**   * 1. 6.1 CLIENT SIDE CODING   6.2.SERVER SIDE CODING | **31**  **32**  **42** |
| **7.** | **TESTING**  7.1 TESTING OBJECTIVES  7.2 TYPES OF TESTS | **47**  **48**  **48** |
|  | 7.2.1 Unit Testing  7.2.2 Integeration Testing  7.2.3 Functional Testing  7.2.4 System Testing  7.2.5 Acceptance Testing | **48**  **48**  **49**  **49**  **49** |
|  | 7.3 REPORT OF TEST CASES | **50** |
| **8.** | **CONCLUSION AND FUTURE ENHANCEMENT** | **51** |
| **9.** | **APPENDICES** | **53** |
|  | **BIBLOGRAPHY** | **58** |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **TITLE** | **PAGE NO** |
| **4.1** | USE-CASE DIAGRAM | **19** |
| **4.2** | ACTIVITY DIAGRAM | **20** |
| **4.3** | SEQUENCE DIAGRAM | **22** |
| **4.4** | DATA FLOW DIAGRAM 0 | **22** |
| **4.5** | DATA FLOW DIAGRAM 1 | **23** |
| **4.6** | ER DIAGRAM | **24** |
| **5.1** | ARCHITETURE DIAGRAM | **27** |
| **5.2.2** | NEURAL NETWORK DIAGRAM | **29** |
| **5.2.3** | CLASSIFICATION AND OUTPUT DIAGRAM | **30** |
| **9.1.1** | PROGRAM EXECUTION | **54** |
| **9.1.2** | UI INTERFACE | **54** |
| **9.1.3** | SINGLE FACE MASK DETECTION | **55** |
| **9.1.4** | NO FACE MASK DETECTION | **55** |
| **9.1.5** | FACE MASK DETECTION WITH MULTIPLE FACES | **56** |
| **9.1.6** | DETECTION OF MULTIPLE FACES | **56** |
| **9.1.7** | SHOWING NO MASK COVERING WITH HAND | **57** |
| **9.1.8** | REAL TIME PREDICTION | **57** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | **LIST OF TABLES** |  | |
| **TABLE**  **NO.** | **TABLE NAME** | | | **PAGE**  **NO.** |
| **4.1.1** | USECASE DIAGRAM SYMBOL DESCRIPTION | | | **19** |
| **4.2.1** | ACTIVITY DIAGRAM SYMBOL DESCRIPTION | | | **21** |
| **4.3.1** | ER DIGRAM SYMBOL DESCRIPTION | | | **25** |
| **7.1** | TEST CASES AND POSSIBLE RESULTS | | | **50** |

# INTRODUCTION

**1.INTRODUCTION**

* 1. **Introduction**

According to the World Health Organization (WHO)’s official Situation Report – 205, coronavirus disease 2019 (COVID-19) has globally infected over 20 million people causing over a 0.7million deaths. Individuals with COVID-19 have had a wide scope of symptoms reported – going from mellow manifestations to serious illness. Respiratory problems like shortness of breath or difficulty in breathing is one of them. Elder people having lung disease can possess serious complications from COVID-19 illness as they appear to be at higher risk. Some common human coronaviruses that infect public around the world are 229E, HKU1, OC43, and NL63. Before debilitating individuals, viruses like 2019-nCoV, SARS-CoV, and MERS-CoV infect animals and evolve to human coronaviruses. Persons having respiratory problems can expose anyone (who is in close contact with them) to infective beads. Surroundings of a tainted individual can cause contact transmission as droplets carrying the virus may withal arrive on his adjacent surfaces.

**1.2 Ideology**

COVID-19 had a massive impact on human lives. The pandemic lead to the loss of millions and affected the lives of billions of people. Its negative impact was felt by almost all commercial establishments, education, economy, religion, transport, tourism, employment, entertainment, food security and other industries. According to WHO(World Health Organization), 55.6 million people were infected with Coronavirus and 1.34 million people died because of it as of November 2020. This stands next to black death which almost took the lives of 60 percent of population in Europe in the 14th century. After the person gets

infected, it takes almost fourteen days for the virus to grow in the body of its host and affect them and in the meantime, it spreads to almost everyone who is in contact with that person. So, it is extremely hard to keep the track of the spread of COVID-19.

To curb certain respiratory viral ailments, including COVID-19, wearing a clinical mask is very necessary. The public should be aware of whether to put on the mask for source control or aversion to COVID-19. Potential points of interest in the utilization of masks lie in reducing the vulnerability of risk from a­­­ noxious individual during the "pre-symptomatic" period and stigmatization of discrete persons putting on masks to restrain the spread of the virus. WHO stresses prioritizing medical masks and respirators for health care assistants. Therefore, face mask detection has become a crucial task in a present global society.

Face mask detection involves detecting the location of the face and then determining whether it has a mask on it or not. The issue is proximately cognate to general object detection to detect the classes of objects. Face identification categorically deals with distinguishing a specific group of entities i.e. Face. It has numerous applications, such as autonomous driving, education, surveillance, and so on. This paper presents a simplified approach to serve the above purpose using the basic Machine Learning (ML) packages such as TensorFlow, Keras, OpenCV, and Scikit-Learn.

**1.3 Machine Learning Technologies**

These are used to predict the class/target/labels/categories of a given data point. Classification belongs to the category of supervised learning in which the targets are provided with input data. They are used in many applications like medical diagnosis, spam detection, target marketing etc. They use a mapping function (f) from input variables (X) to discrete output variables(Y).

**OpenCV**

OpenCV is an open-source library that is primarily used for Computer Vision Applications. This contains many functions and algorithms for Motion tracking, Facial recognition, Object Detection, Segmentation and recognition, and many other applications. Images and real-time video streams can be manipulated to suit different needs using this library.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high-resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user communities and an estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups, and by governmental bodies.

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, and Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV’s deployed uses span the range from stitching Streetview images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detecting of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java, and MATLAB interfaces and supports Windows, Linux, Android, and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCV interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

**LITERATURE SURVEY**

**2.LITERATURE SURVEY**

**Guangzhou, Wei Feng, Jintao Jin, Qujiang Lei Xiuhao Li, Guangchao Gui, Weijun Wang. “**Face Mask Recognition System with YOLOV5 Based on Image Recognition” in IEEE conference location Chengudu, china, 14 DEC 2020.

The rapid development of computer vision makes human-computer interaction possible and has a wide application prospect. Since the discovery of the first case of COVID-19, the global fight against the epidemic has begun. In addition to various studies and findings by medical and health care experts, people's daily behaviors have also become key to combating the epidemic. In China, the government has taken active and effective measures of isolation and closure [1]

**Mohammad Syazwan Mazli Shahar**, **Lucyantie Mazalan.** “Face Identity for Face Mask Recognition System” in  [IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)](https://ieeexplore.ieee.org/xpl/conhome/9431759/proceeding) on April 2021

Facial recognition system analyzes and compares patterns based on the facial information of a person. The technology is widely implemented as a security mechanism in the area such as public surveillance and mobile communication to identify the identity of a person. Today, wearing a face mask is mandatory during COVID-19 pandemic. [2]

**Md. Shahriar Islam, Eimdadul Haque Moon, Md. Ashikujaman Shaikat; Mohammad Jahangir Alam. “**A Novel Approach to Detect Face Mask using CNN” in [2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)](https://ieeexplore.ieee.org/xpl/conhome/9315857/proceeding)

Face detection and recognition will be considered as one of the most intriguing modalities for biometric models. Those researches are mainly required for ensuring security in a most sensitive area. This research paper has proposed a very fast image pre-processing with the mask in the center over the faces. For this system, features extraction and Convolutional Neural Network are used for classification and detection of a masked person.[3]

**Arjya Das, Mohammad Wasif Ansari, Rohini Basak.”**Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV” on [2020 IEEE 17th India Council International Conference (INDICON)](https://ieeexplore.ieee.org/xpl/conhome/9341296/proceeding)

**C**OVID-19 pandemic has rapidly affected our day-to-day life disrupting the world trade and movements. Wearing a protective face mask has become a new normal. In the near future, many public service providers will ask the customers to wear masks correctly to avail of their services. Therefore, face mask detection has become a crucial task to help global society. [4]

**Harish Adusumalli, D. Kalyani; R.Krishna Sri, M. Pratapteja, P.V.R.D Prasada Rao. “**Face Mask Detection Using OpenCV” First published on 31 march,2021

The COVID-19 pandemic is causing a worldwide emergency in healthcare. This virus mainly spreads through droplets which emerge from a person infected with coronavirus and poses a risk to others. The risk of transmission is highest in public places. One of the best ways to stay safe from getting infected is wearing a face mask in open territories as indicated by the World Health Organization (WHO). [5]

**Sammy V. Militante, Nanette V. Dionisio. “**Real-Time Facemask Recognition with Alarm System using Deep Learning” in [2020 11th IEEE Control and System Graduate Research Colloquium (ICSGRC)](https://ieeexplore.ieee.org/xpl/conhome/9223048/proceeding).

In the background of the COVID-19 pandemic, institutions such as the academy suffer a great deal from practically closed globally if the current situation is not going to change. COVID-19 also known as Serious Acute Respiratory Syndrome Corona virus-2 is an infectious disease that is released from an infected sick person who speaks, sneezes, or coughs by respiratory droplets. [6]

**Susanto Susanto, Febri Alwan Putra, Riska Analia, Ika Karlina Laila Nur Suciningtyas. “**The Face Mask Detection For Preventing the Spread of COVID-19 at Politeknik Negeri Batam” in [2020 3rd International Conference on Applied Engineering (ICAE)](https://ieeexplore.ieee.org/xpl/conhome/9350316/proceeding)

After the new Coronavirus disease (COVID-19) case spread rapidly in Wuhan-China in December 2019, World Health Organization (WHO) confirmed that this is a dangerous virus which can be spreading from humans to humans through droplets and airborne. As for the prevention, wearing a face mask is essentials while going outside or meeting to others. [7]

**Bin Xue, Jianpeng Hu, Pengming Zhang. “**Intelligent detection and recognition system for mask wearing based on improved RetinaFace algorithm” in [2020 2nd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)](https://ieeexplore.ieee.org/xpl/conhome/9360928/proceeding).

The COVID-19 can be transmitted through airborne droplets, aerosols and other carriers. In order to better reduce people's risk of infection, individuals need to wear masks to prevent the spread of the virus when going out to public places, seeking medical treatment, and taking public transportation. This paper is based on the improved RETINAFACE algorithm,[8]

**Samuel Ady Sanjaya, Suryo Adi Rakhmawan. “**Face Mask Detection Using MobileNetV2 in The Era of COVID-19 Pandemic” in 2[020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)](https://ieeexplore.ieee.org/xpl/conhome/9325571/proceeding)

Corona Virus Disease (COVID-19) pandemic is causing a health crisis. One of the effective methods against the virus is wearing a face mask. This paper introduces face mask detection that can be used by the authorities to make mitigation, evaluation, prevention, and action planning against COVID-19. The face mask recognition in this study is developed with a machine learning algorithm through the image classification method: MobileNetV2. [9]

**V Nithyashree, S Roopashree, Aparna Duvvuri, L Vanishree, Disha Anand Madival, G Vidyashree. “**A Solution to Covid-19: Detection and Recognition of Faces with Mask” in [2021 International Conference on Intelligent Technologies (CONIT)](https://ieeexplore.ieee.org/xpl/conhome/9497779/proceeding) on 27 june 2021

In this COVID-19 crisis, wearing masks is necessary and no longer an option to the general public. To follow the strict directives given by the government, the businesses have to implement a cost-effective approach to ensure that all its employers wear a face mask and help to control the spread of coronavirus. [10]

**SYSTEM ANALYSIS**

**3.SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM**

Manual monitoring is in general hard to enforce because of the manpower needed to efficiently protect public spaces and ensure that individuals are wearing masks correctly. The current systems are based on a machine learning approach and are highly prone to errors. Although face mask detection results achieved are promising, these traditional approaches are still far from being highly accurate and efficient. The existing systems are simple and effective but are extremely vulnerable to impact. Moreover, state-of-the-art methods can predict outputs pertaining to certain parts of the organ only while some severe conditions may go completely undetected. This could lead practitioners to false assumptions and improper diagnoses and treatments provided to patients.

The existing systems are also not robust in detecting face masks on a human face, as the characteristics of the diseases vary in terms of time due to changes in climate, geographical, and biological aspects. Many of the current techniques are computer vision-based models, which are less accurate and efficient. The discussed limitations have been overcome to enhance the performance of face mask detection models successfully in the presented application. The existing models cause unsatisfactory results & excessive trouble to the general public and authorities.

**Problems in Existing System**

* It doesn’t shows the multiple persons at a time.
* It shows error sometimes.
* It doesn’t shows the percentage value.

**3.2 PROPOSED SYSTEM**

DL allows for analyzing and interpreting massive volumes of data in a fast and accurate way. We propose to use MobileNetV2 architecture to ensure accurate face-mask detection. The proposed approach provides accurate detection of face mask-wearing and whether it is worn in an appropriate way or not in real-time. The first stage is Pre-processing. Images in the training dataset are subjected to Pre-processing to avoid false predictions and to enhance the quality of images. Live capture of multimedia impressions is used for detailed analysis and predictions. To do this, a complete dataset is created after Image de-Colorization and contouring, hence the dataset is very pure.

In addition, the MobileNetV2 is used as a DL architecture for facemask detection. This approach has the advantage of being fast and suited to edge devices, and it provides excellent results for object detection. This approach has the advantage of being fast and suited to edge devices, and it provides excellent results for object detection. The proposed approach achieved 99% accuracy in training and testing and can determine whether a mask is appropriately worn or not in real-time video streams. If deployed correctly, this deep learning technique helps in simplifying the work of frontline warriors and saving their lives. The program can also be linked to the entrance gates, allowing only those who are wearing masks to enter. It can also be used in shopping malls and universities.

The novelty of this paper with regard to exciting works is proposing an efficient and accurate approach for real-time videos. The proposed model works well against train data and test data further this model will provide better results for real-time data. The application is beneficial for conditions where data has to be processed in a short time and results are required instantly. The proposed device might be deployed in high-traffic areas to keep a close eye on people. If we consider the cost estimation for implementing the project, it will be almost of no cost as most metropolitan cities already have cameras installed in public places. The camera; which is the only main requirement of the proposed model, is already available. We propose to use MobileNetV2 architecture to ensure accurate face-mask detection, which has high accuracy. A fast and cost- efficient and quick method.

**3.3 REQUIREMENT SPECIFICATION**

**3.3.1 HARDWARE REQUIREMENTS**

Processor : Intel Core i5 3.29GHz

Hard disk : 1 TB

RAM : 16 GB

Keyboard : 110 keys enhanced

**3.3.2 SOFTWARE REQUIREMENTS**

Operating system : Windows7 (with service pack 1), 8, 8.1 and 10

IDE : Anaconda 2.0

Language : Python

**3.4 PLATFORM SPECIFICATION– ANACONDA**

Anaconda is an open-source package manager for Python and R. It is the most popular platform among data science professionals for running Python and R implementations. There are over 300 libraries in data science, so having a robust distribution system for them is a must for any professional in this field.Anaconda simplifies package deployment and management. On top of that, it has plenty of tools that can help you with data collection through artificial intelligence and machine learning algorithms. With Anaconda, you can easily set up, manage, and share Conda environments. Moreover, you can deploy any required project with a few clicks when you’re using Anaconda.There are many advantages to using Anaconda and the following are the most prominent ones among them:Anaconda is free and open-source. This means you can use it without spending any money. In the data science sector, Anaconda is an industry staple. It is open-source too, which has made it widely popular. If you want to become a data science professional, you must know how to use Anaconda for Python because every recruiter expects you to have this skill. It is a must-have for data science.

It has more than 1500 Python and R data science packages, so you don’t face any compatibility issues while collaborating with others. For example, suppose your colleague sends you a project which requires packages called A and B but you only have package A. Without having package B, you wouldn’t be able to run the project. Anaconda mitigates the chances of such errors. You can easily collaborate on projects without worrying about any compatibility issues.It gives you a seamless environment that simplifies deploying projects. You can deploy any project with just a few clicks and commands while managing the rest. Anaconda has a thriving community of data scientists and machine learning professionals who use it regularly. If you encounter an issue, chances are, the community has already answered the same. On the other hand, you can also ask people in the community about the issues you face there, it’s a very helpful community ready to help new learners. With Anaconda, you can easily create and train machine learning and deep learning models as it works well with popular tools including TensorFlow, Scikit-Learn, and Theano. You can create visualizations by using Bokeh, Holoviews, Matplotlib, and Datashader while using Anaconda.

**How to Use Anaconda for Python**

Now that we have discussed all the basics in our Python Anaconda tutorial, let’s discuss some fundamental commands you can use to start using this package manager.

Listing All Environments

To begin using Anaconda, you’d need to see how many Conda environments are present in your machine.

conda env list

It will list all the available Conda environments in your machine.

Creating a New Environment

You can create a new Conda environment by going to the required directory and use this command:

conda create -n <your\_environment\_name>

You can replace <your\_environment\_name> with the name of your environment. After entering this command, conda will ask you if you want to proceed to which you should reply with y:

proceed ([y])/n)?

On the other hand, if you want to create an environment with a particular version of Python, you should use the following command:

conda create -n <your\_environment\_name> python=3.6

Similarly, if you want to create an environment with a particular package, you can use the following command:

conda create -n <your\_environment\_name>pack\_name

Here, you can replace pack\_name with the name of the package you want to use.

If you have a .yml file, you can use the following command to create a new Conda environment based on that file:

conda env create -n <your\_environment\_name> -f <file\_name>.yml

We have also discussed how you can export an existing Conda environment to a .yml file later in this article.

**Activating an Environment**

You can activate a Conda environment by using the following command:

conda activate <environment\_name>

You should activate the environment before you start working on the same. Also, replace the term <environment\_name> with the environment name you want to activate. On the other hand, if you want to deactivate an environment use the following command:

conda deactivate

**Installing Packages in an Environment**

Now that you have an activated environment, you can install packages into it by using the following command:

conda install <pack\_name>

Replace the term <pack\_name> with the name of the package you want to install in your Conda environment while using this command.

**Updating Packages in an Environment**

If you want to update the packages present in a particular Conda environment, you should use the following command:

conda update

The above command will update all the packages present in the environment. However, if you want to update a package to a certain version, you will need to use the following command:

conda install <package\_name>=<version>

**Exporting an Environment Configuration**

Suppose you want to share your project with someone else (colleague, friend, etc.). While you can share the directory on Github, it would have many Python packages, making the transfer process very challenging. Instead of that, you can create an environment configuration .yml file and share it with that person. Now, they can create an environment like your one by using the .yml file.

For exporting the environment to the .yml file, you’ll first have to activate the same and run the following command:

conda env export ><file\_name>.yml

The person you want to share the environment with only has to use the exported file by using the ‘Creating a New Environment’ command we shared before.

**Removing a Package from an Environment**

If you want to uninstall a package from a specific Conda environment, use the following command:

conda remove -n <env\_name><package\_name>

On the other hand, if you want to uninstall a package from an activated environment, you’d have to use the following command:

conda remove <package\_name>

**Deleting an Environment**

Sometimes, you don’t need to add a new environment but remove one. In such cases, you must know how to delete a Conda environment, which you can do so by using the following command:

conda env remove –name <env\_name>

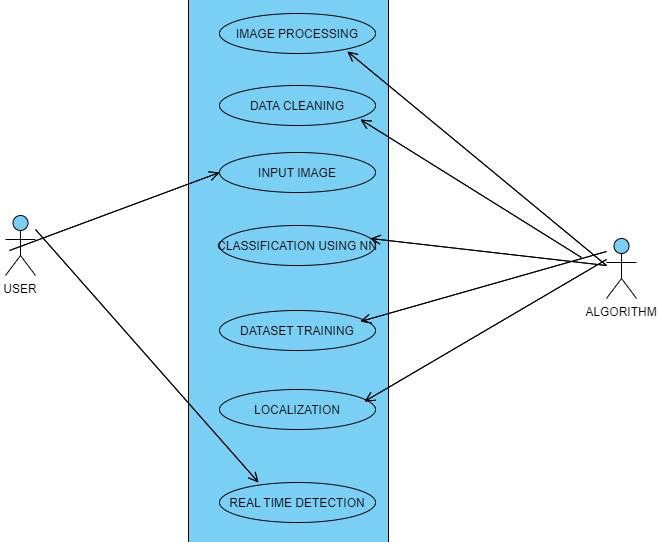
The above command would delete the Conda environment right away.

**SYSTEM DESIGN**

**4.SYSTEM DESIGN**

**4.1 USE-CASE DIAGRAM**

A use case diagram is a standard diagram that shows all interactions between the user, dataset, and algorithm used. It is developed in the early stages of the process.



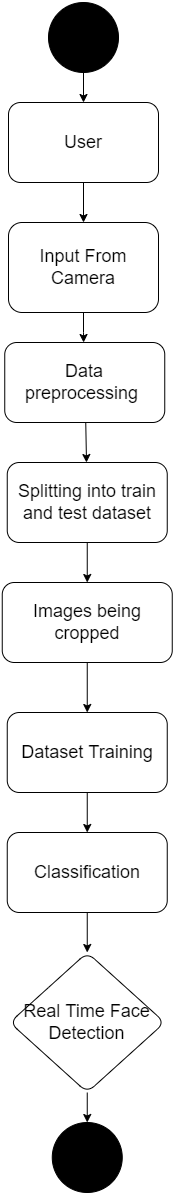
**Fig 4.1 – Use-Case Diagram**

|  |  |  |
| --- | --- | --- |
| Symbol Name | Symbol | Description |
| Entity |  | An entity is represented by a rectangle which contains the entity’s name. |
| Attribute |  | In the Chen notation, each attribute is represented by an oval containing attribute’s name |
| One or More |  | It represents One or More |

**Table 4.1.1 Use Case diagram description**

**4.2 ACTIVITY DIAGRAM**

In simple terms, a diagram that represents the order of all activities is called the activity diagram. It shows the workflow between different activities that take place in the whole process. However, these are not exactly flowcharts but are similar.

****

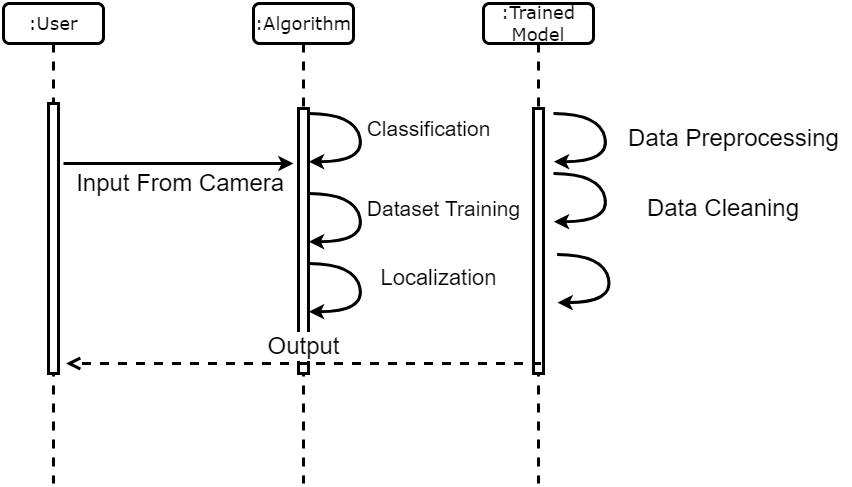
**Fig 4.2 – Activity Diagram**

|  |  |  |
| --- | --- | --- |
| **Symbol Name** | **Symbol** | **Description** |
| Start/Initial State |  | A small filled circle followed by an arrow represents start point for  any activity diagram. |
| Activity State |  | An action state represents the non-interruptible action of objects. |
|  |  | A diamond represents a |
|  | decision with alternate |
|  | paths. The outgoing |
| Decisions and Branching | alternates should be labelled with a condition |
|  | or guard expression. You |
|  | can also label one of the |
|  | paths "else." |
| Final State |  | An arrow pointing to a filled circle nested inside another circle represents the final action state. |

**Table 4.2.1 Activity diagram description**

**4.3 SEQUENCE DIAGRAM**

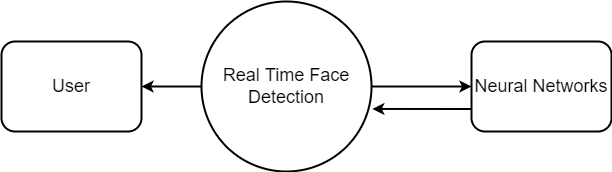
These are other kinds of interaction-based diagrams that show how all the operations are carried out. They capture the context of collaborations between objects and processes.



**Fig 4.3 – Sequence Diagram**

**4.4 DATA FLOW DIAGRAM LEVEL 0**

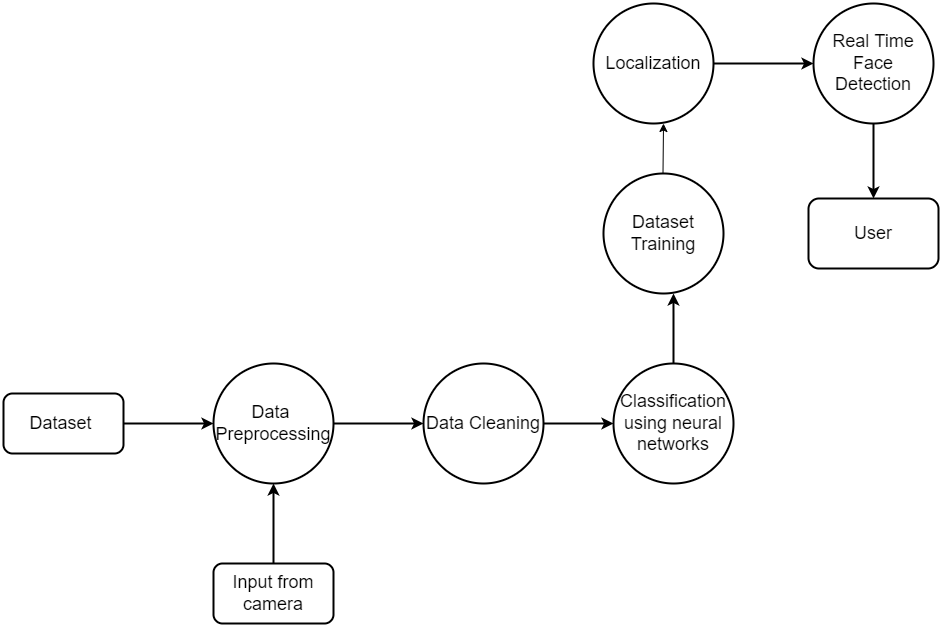
This is basically a contextual diagram, also referred to as a “context diagram”. It only represents the top level or the 0 Level in the whole process.it gives an abstraction kind of view and shows the whole system as a single process and its relationship to externalities.

****

**Fig 4.4 – Data Flow Diagram 0**

**4.5 DATA FLOW DIAGRAM LEVEL 1**

Level DFDs represent the complete system as a single process. it notates every process and sub-process that comes together in a sequence to form the complete system. This along with ) and 2-level data flow diagrams comprise the “fundamental system model”.

****

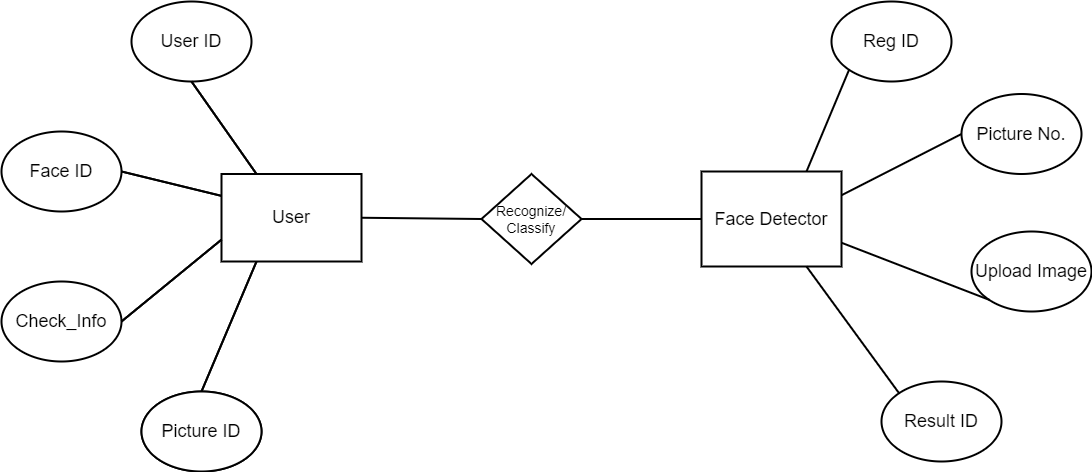
**Fig 4.5 – Data Flow Diagram**

|  |  |  |
| --- | --- | --- |
| Symbol Name | Symbol | Description |
| Entity |  | An entity is represented by a rectangle which contains the entity’s name. |
| Attribute |  | In the Chen notation, each attribute is represented by an oval containing attribute’s name |
| One or More |  | It represents One or More |

**Table 4.2.1 data flow diagram description**

**4.6 ER DIAGRAM**

ER stands for Entity Relationship. These diagrams display the relationship of entities that are used and stored in the database. They explain the structure of the whole process. these diagrams can be made using three basic concepts, attributed, relationships, and entities.



**Fig 4.6 – ER Diagram**

|  |  |  |
| --- | --- | --- |
| **Symbol Name** | **Symbol** | **Description** |
| Start/Initial State |  | A small filled circle followed by an arrow represents start point for  any activity diagram. |
| Activity State |  | An action state represents the non-interruptible action of objects. |
|  |  | A diamond represents a |
|  | decision with alternate |
|  | paths. The outgoing |
| Decisions and Branching | alternates should be labelled with a condition |
|  | or guard expression. You |
|  | can also label one of the |
|  | paths "else." |
| Final State |  | An arrow pointing to a filled circle nested inside another circle represents the final action state. |

**4.6.1 ER diagram description**

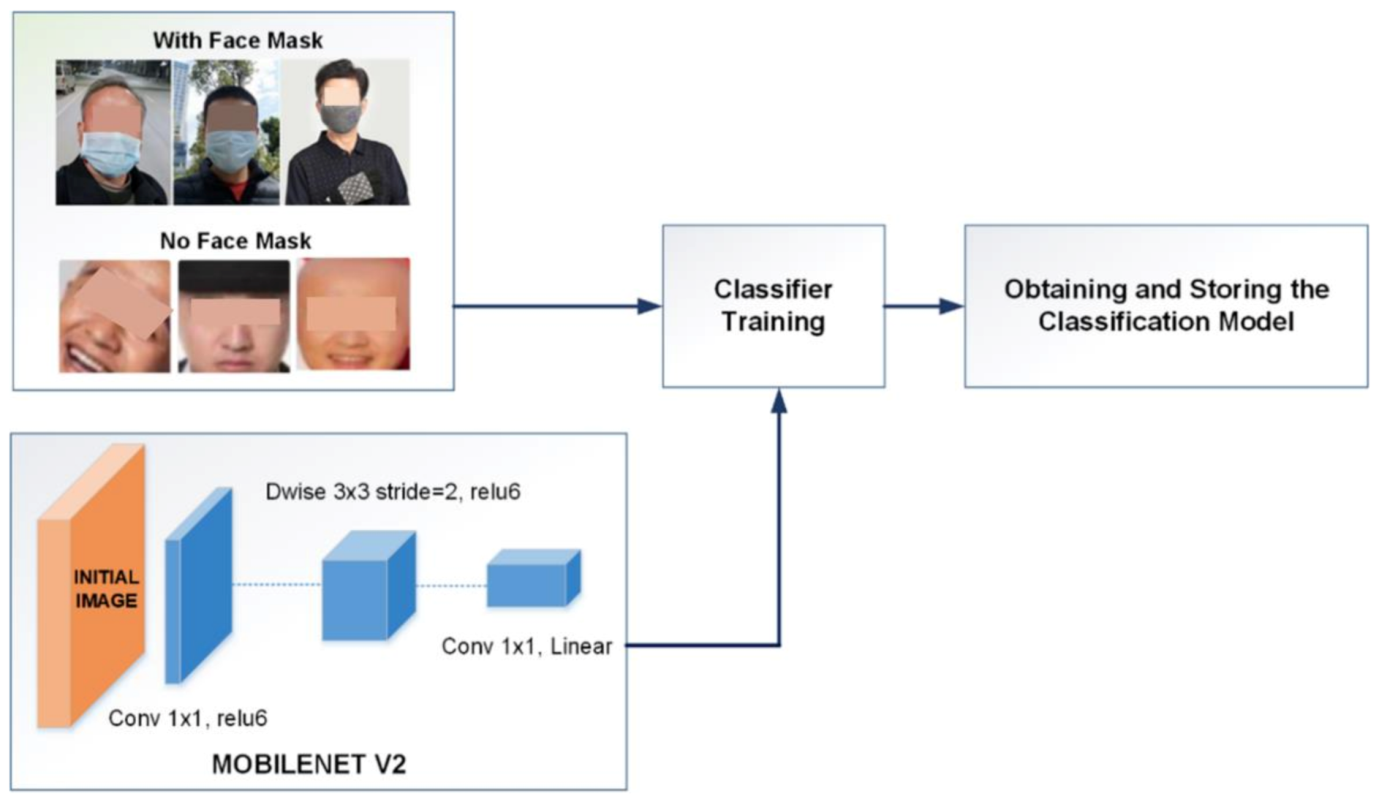
**SYSTEM ARCHITECTURE**

**5.SYSTEM ARCHITECTURE**

**5.1 Architecture Diagram**

The system get an input image from the user. After getting the image the system pre-processes the input image using medium filter, After that feature Extraction and selection Now the database image is taken for CNN algorithm.

In the CNN classification, we use google net (image processing) to train the system and will classify whether the image is normal or tuberculosis image. If the system confirms a normal image then it gives the result as normal or if the system confirms that tuberculosis is present in that image then output image is show in a new window and the result of that image (pothole is present) is displayed.



**5.1 Architecture Diagram**

**5.2 MODULE DESCRIPTION**

**5.2.1 MODULE 1: GUI AND DATA PREPROCESSING**

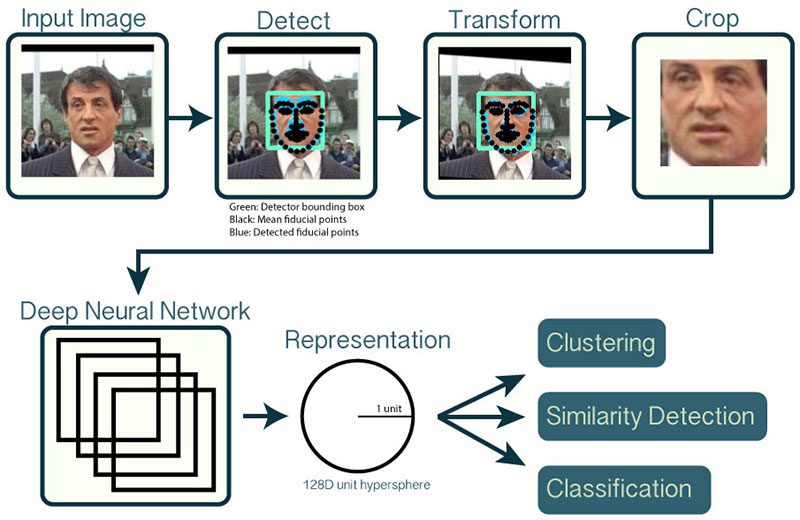
The gathering of datasets for face mask detection was done through continuous image de-colouration and contouring, hence the dataset is very pure. In many cases, the main module is the input module where the data are implemented, in this module the data can be directly added via GUI and through an endpoint connection which can be able to produce the output. The GUI application is built in python. Face mask detection refers to detecting faces in the image and then classifying each face as with a mask or without a mask. The process begins with preprocessing of the given data from a large dataset. The data is cleaned and pre-processed at this stage, where missing and null value records are dropped. In our dataset, we cleaned all the null values and checked whether all the data types are valid. The main purpose of preprocessing is to identify and drop or substitute the missing values in the dataset which occupy a very small part of the whole data, to ensure an accurate result.

**5.2.2 MODULE 2: TRAINING NEURAL NETWORKS**

MobileNetV2 is an architecture of bottleneck depth separable convolution building of basic blocks with residuals. We propose to use MobileNetV2 architecture to ensure accurate face-mask detection, the proposed CNN architecture. We choose to use MobileNetV2 since it provides several advantages such as:

* it is a light-weight DL suited to edge devices,
* it provides excellent results for object detection, and
* it can efficiently tradeoff between accuracy and latency using simple global hyperparameters.

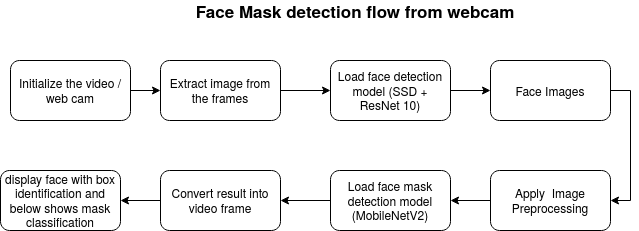
The most amazing thing about this approach is that it actually works remarkably well. They tend to work best when all the features are at least, well, relevant, because the number of features selected for a particular tree is small.



**5.2.2 Neural networks model in face mask detection**

**5.2.3 MODULE 3: CLASSIFICATION AND OUTPUT**

The MobilenetV2 architecture is used as a foundation for the classifier to perform real-time mask identification. Since MobileNet v2 outperforms other state of art methods in predicting outcomes accurately, it is used efficiently in this model. Then the proposed CNN classifies faces with and without masks as the output layer of the proposed CNN architecture. It saves the best model with the highest accuracy. Finally, the saved model can be used in the real world. The name of this phase is model evaluation. Overfitting when any classification parameter like with mask and without mask needs to be adjusted, it is necessary to have a training dataset apart from training and testing datasets. Web and desktop apps can use the live video feed for detection. The program can also be linked to the entrance gates, allowing only those who are wearing masks to enter. It can also be used in shopping malls and universities.



**5.3.3 classification and result output**

**SYSTEM IMPLEMENTATION**

**6.1 Client side coding**

**Gui.py**

# import the necessary packages

from tensorflow.keras.applications.mobilenet\_v2 import preprocess\_input

from tensorflow.keras.preprocessing.image import img\_to\_array

from tensorflow.keras.models import load\_model

from imutils.video import VideoStream

import numpy as np

import imutils

import time

import cv2

import os

import winsound

frequency=2500

duration=50

def detect\_and\_predict\_mask(frame, faceNet, maskNet):

(h, w) = frame.shape[:2]

blob = cv2.dnn.blobFromImage(frame, 1.0, (224, 224),

(104.0, 177.0, 123.0))

faceNet.setInput(blob)

detections = faceNet.forward()

print(detections.shape)

faces = []

locs = []

preds = []

# loop over the detections

for i in range(0, detections.shape[2]):

confidence = detections[0, 0, i, 2]

if confidence > 0.5:

box = detections[0, 0, i, 3:7] \* np.array([w, h, w, h])

(startX, startY, endX, endY) = box.astype("int")

(startX, startY) = (max(0, startX), max(0, startY))

(endX, endY) = (min(w - 1, endX), min(h - 1, endY))

face = frame[startY:endY, startX:endX]

face = cv2.cvtColor(face, cv2.COLOR\_BGR2RGB)

face = cv2.resize(face, (224, 224))

face = img\_to\_array(face)

face = preprocess\_input(face)

# add the face and bounding boxes to their respective

# lists

faces.append(face)

locs.append((startX, startY, endX, endY))

# only make a predictions if at least one face was detected

if len(faces) > 0:

faces = np.array(faces, dtype="float32")

preds = maskNet.predict(faces, batch\_size=32)

# return a 2-tuple of the face locations and their corresponding

# locations

return (locs, preds)

# load our serialized face detector model from disk

prototxtPath = r"face\_detector\deploy.prototxt"

weightsPath = r"face\_detector\res10\_300x300\_ssd\_iter\_140000.caffemodel"

faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)

# load the face mask detector model from disk

maskNet = load\_model("mask\_detector.model")

# initialize the video stream

print("[INFO] starting video stream...")

vs = VideoStream(src=1).start()

# loop over the frames from the video stream

while True:

frame = vs.read()

frame = imutils.resize(frame, width=400)

(locs, preds) = detect\_and\_predict\_mask(frame, faceNet, maskNet)

for (box, pred) in zip(locs, preds):

# unpack the bounding box and predictions

(startX, startY, endX, endY) = box

(mask, withoutMask) = pred

label = "Mask" if mask > withoutMask else "No Mask"

color = (0, 255, 0) if label == "Mask" else (0, 0, 255)

if label!="Mask":

winsound.Beep(frequency,duration)

# include the probability in the label

label = "{}: {:.2f}%".format(label, max(mask, withoutMask) \* 100)

cv2.putText(frame, label, (startX, startY - 10),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.45, color, 2)

cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)

# show the output frame

cv2.imshow("Frame", frame)

key = cv2.waitKey(1) & 0xFF

if key == ord("q"):

break

# do a bit of cleanup

cv2.destroyAllWindows()

vs.stop()

import tkinter as tk

from tensorflow.keras.applications.mobilenet\_v2 import preprocess\_input

from tensorflow.keras.preprocessing.image import img\_to\_array

from tensorflow.keras.models import load\_model

from imutils.video import VideoStream

import numpy as np

import imutils

import time

import cv2

import os

frequency=2500

duration=50

import glob

import matplotlib.pyplot as plt

from tkinter import filedialog

from tkinter import \*

from PIL import ImageTk, Image

from keras import models

from tensorflow.keras import preprocessing, layers, models, callbacks

import cv2

from tensorflow.keras.models import load\_model

from keras import models

global loadedModel

import numpy

import imutils

import numpy as np

import glob

import imutils

import numpy as np

from keras import models

from tensorflow.keras import preprocessing, layers, models, callbacks

from sklearn.metrics import pairwise

import time

import os # Operating system functionality

import random # Random number generator

import pandas as pd # Data analysis & manipulation

import numpy as np # Array-processing

import seaborn as sns # Data visualization

import matplotlib.pyplot as plt # Data visualization

from tensorflow.keras import preprocessing, layers, models, callbacks # Neural networks

from sklearn import metrics # Model evaluation

global loadedModel

size = 30

from PIL import Image, ImageTk

from tkinter import \*

import numpy as np

import cv2

from keras.preprocessing import image

from tensorflow.keras.preprocessing.image import img\_to\_array

import numpy as np

#from keras import models

from tensorflow.keras.models import load\_model

# loading Python Imaging Library

from PIL import ImageTk, Image

# To get the dialog box to open when required

from tkinter import filedialog

#initialise GUI

top=tk.Tk()

top.geometry('800x600')

top.title('Detector')

top.configure(background='#CDCDCD')

label=Label(top,background='#CDCDCD', font=('arial',15,'bold'))

def yes():

# load our serialized face detector model from disk

prototxtPath = 'face\_detector/deploy.prototxt'

weightsPath = 'face\_detector/res10\_300x300\_ssd\_iter\_140000.caffemodel'

faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)

# load the face mask detector model from disk

maskNet = load\_model("mask\_detector.model")

# initialize the video stream

print("[INFO] starting video stream...")

camera = cv2.VideoCapture(0)

# loop over the frames from the video stream

while True:

(\_,frame) = camera.read()

frame = imutils.resize(frame, width=400)

(locs, preds) = detect\_and\_predict\_mask(frame, faceNet, maskNet)

for (box, pred) in zip(locs, preds):

(startX, startY, endX, endY) = box

(mask, withoutMask) = pred

label = "Mask" if mask > withoutMask else "No Mask"

color = (0, 255, 0) if label == "Mask" else (0, 0, 255)

if label!="Mask":

#winsound.Beep(frequency,duration)

label = "{}: {:.2f}%".format(label, max(mask, withoutMask) \* 100)

cv2.putText(frame, label, (startX, startY - 10),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.45, color, 2)

cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)

cv2.imshow("Frame", frame)

key = cv2.waitKey(1) & 0xFF

if key == ord("q"):

break

cv2.destroyAllWindows()

vs.stop()

def detect\_and\_predict\_mask(frame, faceNet, maskNet):

(h, w) = frame.shape[:2]

blob = cv2.dnn.blobFromImage(frame, 1.0, (224, 224),

(104.0, 177.0, 123.0))

faceNet.setInput(blob)

detections = faceNet.forward()

faces = []

locs = []

preds = []

# loop over the detections

for i in range(0, detections.shape[2]):

confidence = detections[0, 0, i, 2]

if confidence > 0.5:

box = detections[0, 0, i, 3:7] \* np.array([w, h, w, h])

(startX, startY, endX, endY) = box.astype("int")

(startX, startY) = (max(0, startX), max(0, startY))

(endX, endY) = (min(w - 1, endX), min(h - 1, endY))

face = frame[startY:endY, startX:endX]

face = cv2.cvtColor(face, cv2.COLOR\_BGR2RGB)

face = cv2.resize(face, (224, 224))

face = img\_to\_array(face)

face = preprocess\_input(face)

faces.append(face)

locs.append((startX, startY, endX, endY))

if len(faces) > 0:

faces = np.array(faces, dtype="float32")

preds = maskNet.predict(faces, batch\_size=32)

return (locs, preds)

btn\_real = Button(top, text ='REAL\_TIME', bg='#0052cc', fg='#ffffff',width=10, height=2,command = yes).place(

x = 500, y= 200)

btn\_exit = Button(top, text ='EXIT', bg='#0052cc', fg='#ffffff',width=10, height=2,command = top.destroy).place(

x = 200, y= 200)

label.pack(side=BOTTOM,expand=True)

label.pack(side=BOTTOM,expand=True)

heading = Label(top, text="Mask Detector",pady=20, font=('arial',30,'bold'))

heading.configure(background='#CDCDCD',foreground='#364156')

heading.pack()

top.mainloop()

tensorflow>=1.15.2

keras==2.3.1

imutils==0.5.3

numpy==1.18.2

opencv-python==4.2.0.\*

matplotlib==3.2.1

scipy==1.4.1

# import the necessary packages

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

INIT\_LR = 1e-4

EPOCHS = 20

BS = 32

DIRECTORY = r"/dataset"

CATEGORIES = ["with\_mask", "without\_mask"]

print("[INFO] loading images...")

data = []

labels = []

for category in CATEGORIES:

path = os.path.join(DIRECTORY, category)

for img in os.listdir(path):

img\_path = os.path.join(path, img)

image = load\_img(img\_path, target\_size=(224, 224))

image = img\_to\_array(image)

image = preprocess\_input(image)

data.append(image)

labels.append(category)

lb = LabelBinarizer()

labels = lb.fit\_transform(labels)

labels = to\_categorical(labels)

data = np.array(data, dtype="float32")

labels = np.array(labels)

(trainX, testX, trainY, testY) = train\_test\_split(data, labels,

test\_size=0.20, stratify=labels, random\_state=42)

# construct the training image generator for data augmentation

aug = ImageDataGenerator(

rotation\_range=20,

zoom\_range=0.15,

width\_shift\_range=0.2,

height\_shift\_range=0.2,

shear\_range=0.15,

horizontal\_flip=True,

fill\_mode="nearest")

# load the MobileNetV2 network, ensuring the head FC layer sets are

# left off

baseModel = MobileNetV2(weights="imagenet", include\_top=False,

input\_tensor=Input(shape=(224, 224, 3)))

# construct the head of the model that will be placed on top of the

# the base model

headModel = baseModel.output

headModel = AveragePooling2D(pool\_size=(7, 7))(headModel)

headModel = Flatten(name="flatten")(headModel)

headModel = Dense(128, activation="relu")(headModel)

headModel = Dropout(0.5)(headModel)

headModel = Dense(2, activation="softmax")(headModel)

model = Model(inputs=baseModel.input, outputs=headModel)

for layer in baseModel.layers:

layer.trainable = False

# compile our model

print("[INFO] compiling model...")

opt = Adam(lr=INIT\_LR, decay=INIT\_LR / EPOCHS)

model.compile(loss="binary\_crossentropy", optimizer=opt,

metrics=["accuracy"])

# train the head of the network

print("[INFO] training head...")

H = model.fit(

aug.flow(trainX, trainY, batch\_size=BS),

steps\_per\_epoch=len(trainX) // BS,

validation\_data=(testX, testY),

validation\_steps=len(testX) // BS,

epochs=EPOCHS)

print("[INFO] evaluating network...")

predIdxs = model.predict(testX, batch\_size=BS)

predIdxs = np.argmax(predIdxs, axis=1)

print(classification\_report(testY.argmax(axis=1), predIdxs,

target\_names=lb.classes\_))

print("[INFO] saving mask detector model...")

model.save("mask\_detector.model", save\_format="h5")

# plot the training loss and accuracy

N = EPOCHS

plt.style.use("ggplot")

plt.figure()

plt.plot(np.arange(0, N), H.history["loss"], label="train\_loss")

plt.plot(np.arange(0, N), H.history["val\_loss"], label="val\_loss")

plt.plot(np.arange(0, N), H.history["accuracy"], label="train\_acc")

plt.plot(np.arange(0, N), H.history["val\_accuracy"], label="val\_acc")

plt.title("Training Loss and Accuracy")

plt.xlabel("Epoch #")

plt.ylabel("Loss/Accuracy")

plt.legend(loc="lower left")

plt.savefig("plot.png")

**6.2** **Server side coding**

**Deploy.txt**

input: "data"

input\_shape {

dim: 1

dim: 3

dim: 300

dim: 300

}

layer {

name: "data\_bn"

type: "BatchNorm"

bottom: "data"

top: "data\_bn"

param {

lr\_mult: 0.0

}

param {

lr\_mult: 0.0

}

param {

lr\_mult: 0.0

}

}

layer {

name: "data\_scale"

type: "Scale"

bottom: "data\_bn"

top: "data\_bn"

param {

lr\_mult: 1.0

decay\_mult: 1.0

}

param {

lr\_mult: 2.0

decay\_mult: 1.0

}

scale\_param {

bias\_term: true

}

}

layer {

name: "conv1\_h"

type: "Convolution"

bottom: "data\_bn"

top: "conv1\_h"

param {

lr\_mult: 1.0

decay\_mult: 1.0

}

param {

lr\_mult: 2.0

decay\_mult: 1.0

}

convolution\_param {

num\_output: 32

pad: 3

kernel\_size: 7

stride: 2

weight\_filler {

type: "msra"

variance\_norm: FAN\_OUT

}

bias\_filler {

type: "constant"

value: 0.0

}

}

}

layer {

name: "conv1\_bn\_h"

type: "BatchNorm"

bottom: "conv1\_h"

top: "conv1\_h"

param {

lr\_mult: 0.0

}

param {

lr\_mult: 0.0

}

param {

lr\_mult: 0.0

}

}

layer {

name: "conv1\_scale\_h"

type: "Scale"

bottom: "conv1\_h"

top: "conv1\_h"

param {

lr\_mult: 1.0

decay\_mult: 1.0

}

param {

lr\_mult: 2.0

decay\_mult: 1.0

}

scale\_param {

bias\_term: true

}

}

layer {

name: "conv1\_relu"

type: "ReLU"

bottom: "conv1\_h"

top: "conv1\_h"

}

layer {

name: "conv1\_pool"

type: "Pooling"

bottom: "conv1\_h"

top: "conv1\_pool"

pooling\_param {

kernel\_size: 3

stride: 2

}

}

layer {

name: "layer\_64\_1\_conv1\_h"

type: "Convolution"

bottom: "conv1\_pool"

top: "layer\_64\_1\_conv1\_h"

param {

lr\_mult: 1.0

decay\_mult: 1.0

}

convolution\_param {

num\_output: 32

bias\_term: false

pad: 1

kernel\_size: 3

stride: 1

weight\_filler {

type: "msra"

}

bias\_filler {

type: "constant"

value: 0.0

}

}

}

**TESTING**

**7. TESTING**

**7.1 TESTING OBJECTIVES**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**7.2 TYPES OF TESTS**

**7.2.1 Unit Testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration.

**7.2.2 Integration Testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent.

**7.2.3 Functional testing**

Functional testing provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centred on the following items:

**Valid Input**: identified classes of valid input must be accepted.

**Invalid Input**: identified classes of invalid input must be rejected.

**Functions:** identified functions must be exercised.

**Output**: identified classes of application outputs must be exercised.

## 

## 7.2.4 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing predriven process links and integration points.

## 7.2.5 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirement.

**7.3 TEST CASES & REPORTS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S No** | **Test Cases** | **Expected Output** | **Actual Output** | **Status** |
| 1. | Loading dataset for  Training | The dataset should be trained | Dataset is trained | PASS |
| 2. | Loading the CNN model | The model should be loaded for training the dataset | The model has been loaded successfully | PASS |
| 3. | Training the model | The model should be trained as per the conditions | The model has been trained | PASS |
| 4. | Inserting the input image | The input image should be successfully uploaded for further process | The input image has been uploaded | PASS |
| 5. | Pre-processing the input image | Pre-process the image to reduce distortion and enhance the image for further processing | Displays the pre-processed image | PASS |
| 6. | Classification -  Normal | It should display the message as “Mask with percent” | Displays the message as “mask with percent” | PASS |
| 7. | Classification -  Tuberculosis | It should display the message as “no mask” and display the Segmented image | Displays the message as “no mask” and segmented image | PASS |

**Table 7.1 Test Cases and Possible Results**

**CONCLUSION**

**8.CONCLUSION AND FUTURE ENHANCEMENT**

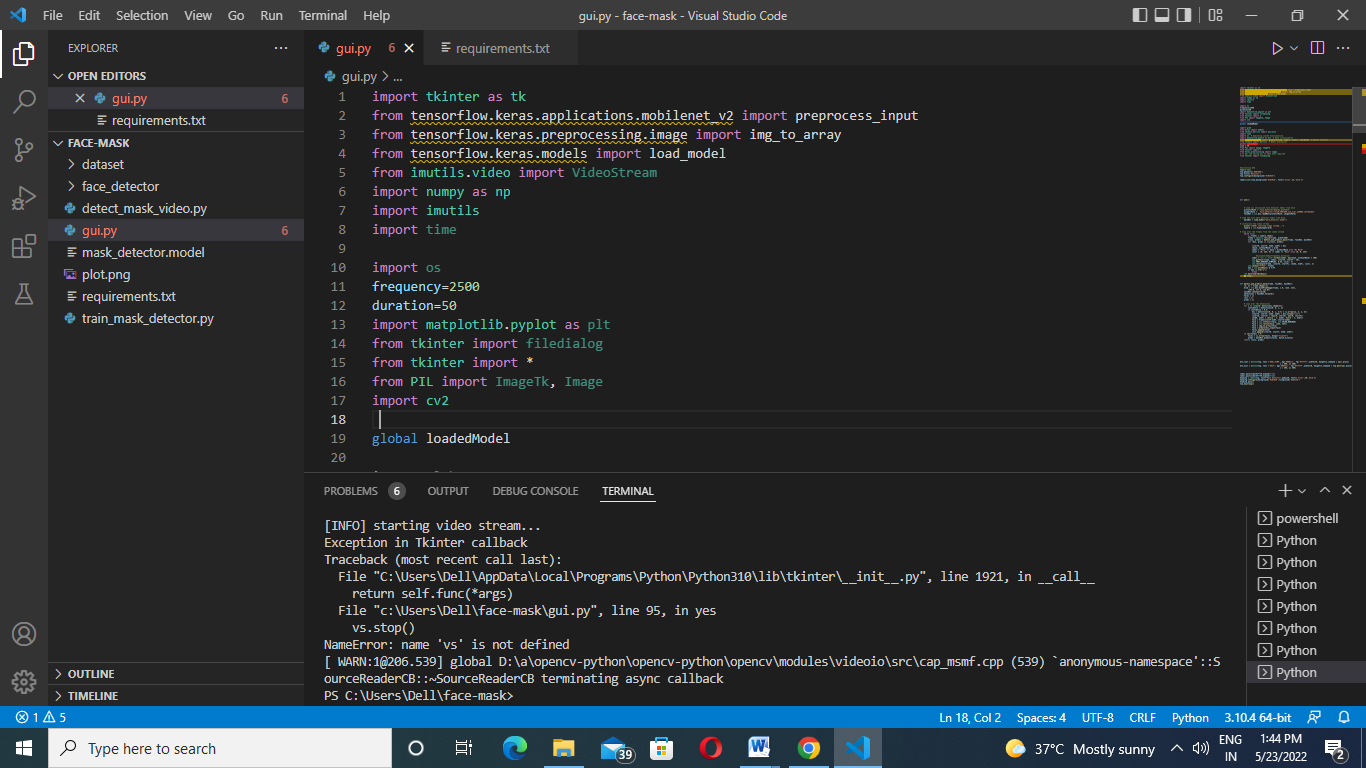
Human faces are different for different people on the basis of various parameters. The uniqueness and measurement of the different parameters help us to recognize the persons. In summary, the state-of-the-art methods for human face recognition have already achieved a high accuracy rate which led to its practical applications.However, it cannot be denied that using the existing method poses many challenges, mainly slow functioning, and inefficiency.Due to the vital importance of developing an intelligent surveillance system to detect and identify human faces, we presented a fast and robust approach that can immediately

The proposed framework is not only much faster than the previous work but also maintains competitive accuracy with the state-of-the-art human detection system.It is independent of the user's physical interaction. It has very precise measurements and permits for high deployment and authentication.Our proposed model for face detection systems is beneficial to the world for advanced applications such as access and security, payments, and criminal identifications. The proposed system has shown excellent performance in the face recognition systems with a high accuracy rate and a much higher speed up rate as compared to the previously used state-of-the-art methods such as CNN.However, some occlusion-based challenges are still inevitable and require more attention and research. Despite some limitations and challenges of the face recognition algorithm, there is scope for us to improvise these frameworks in the near future. For future works, more comprehensive datasets and novel algorithms may reduce the occlusion-based problems. This would enable wider application scenarios.During the Covid-19 era, wearing masks is inevitable and may continue to be so in the coming years for human safety. Thus, we would also like to develop novel ways to detect the faces of people wearing masks, and amp up security for every scenario.

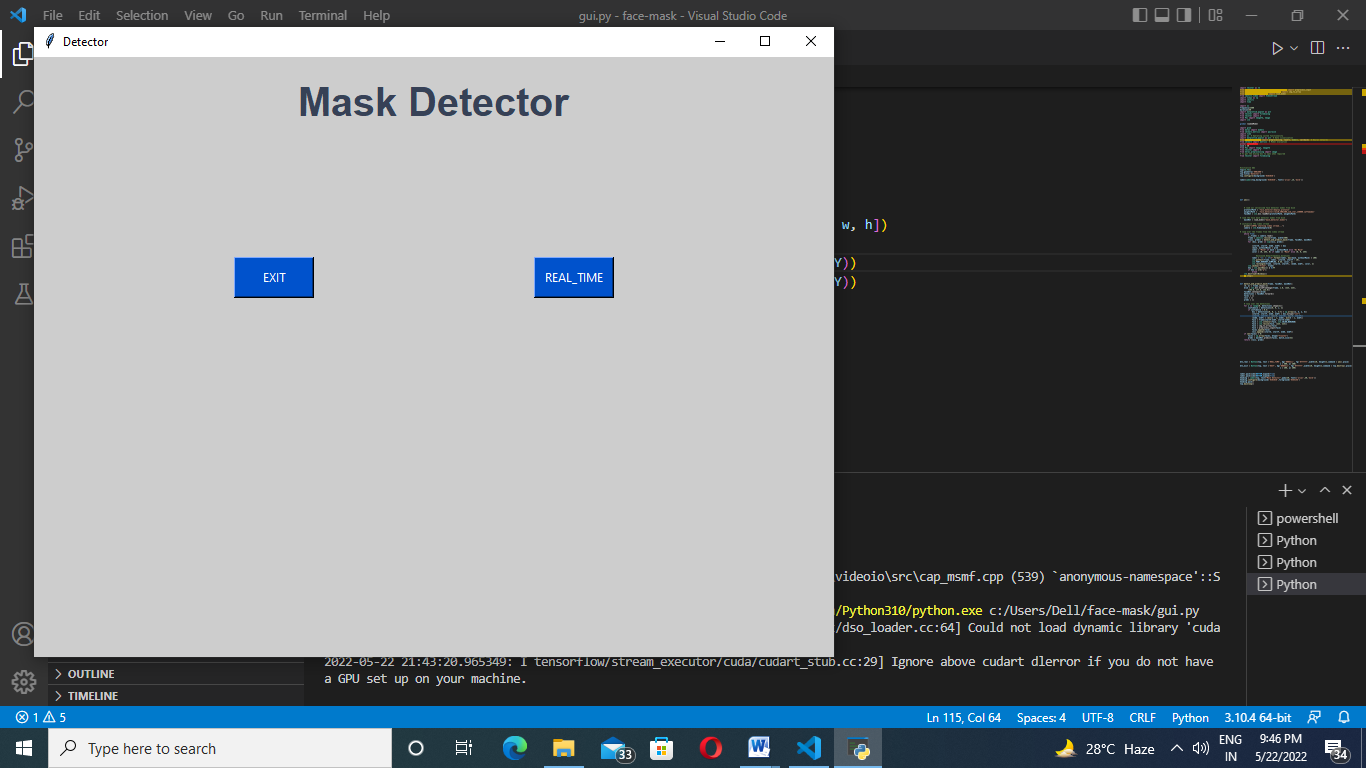
**APPENDICES**

**9. APPENDICES**

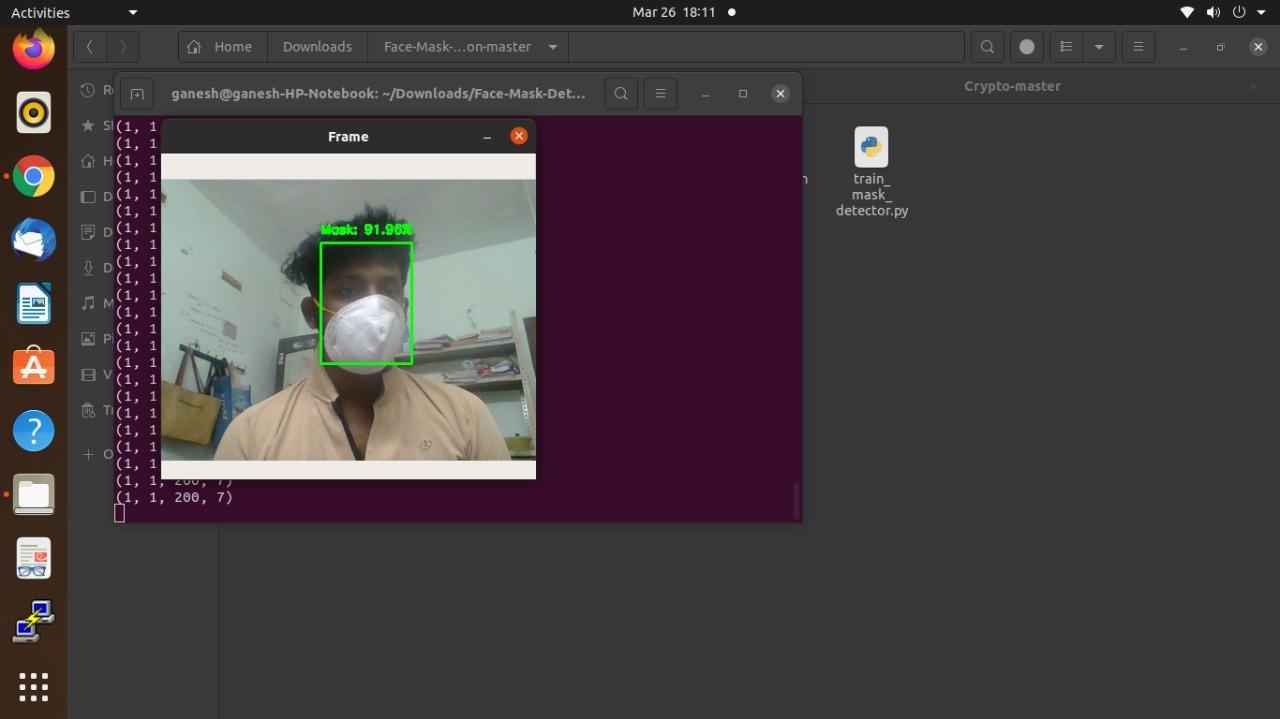
**9.1 SAMPLE SCREENSHOTS**

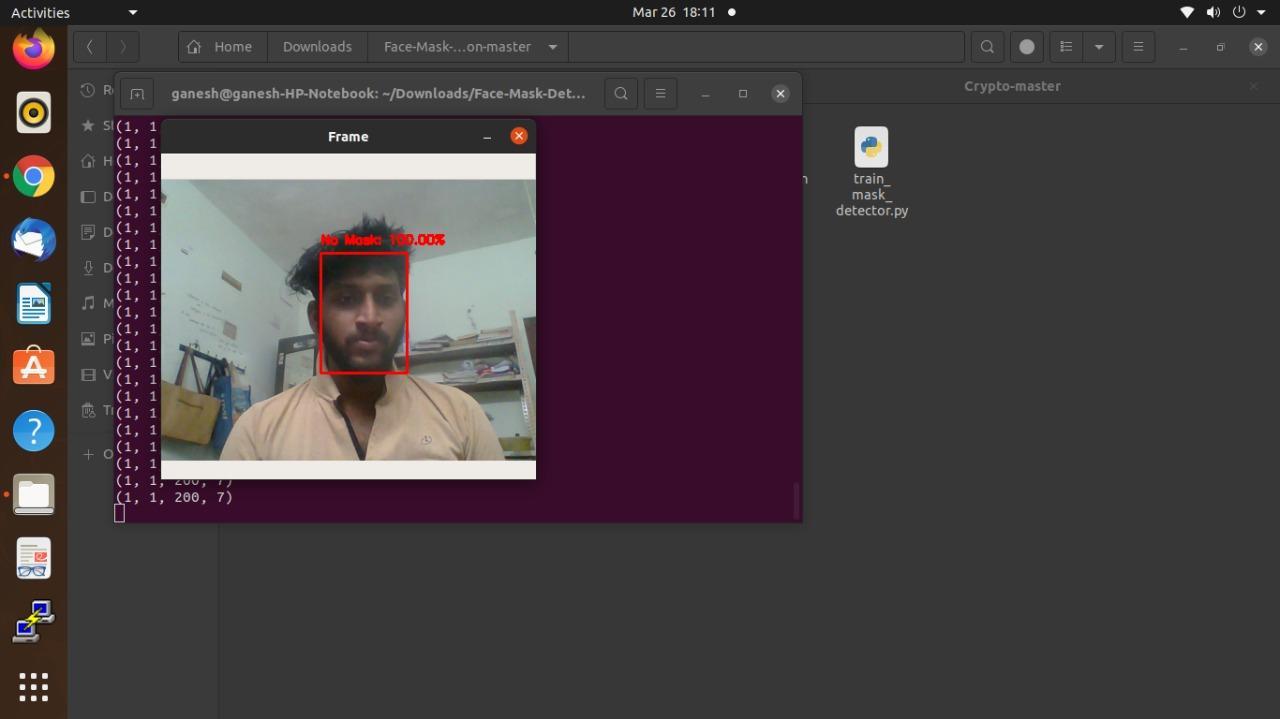
****

**Fig 9.1.1 coding is running on the visual studio**

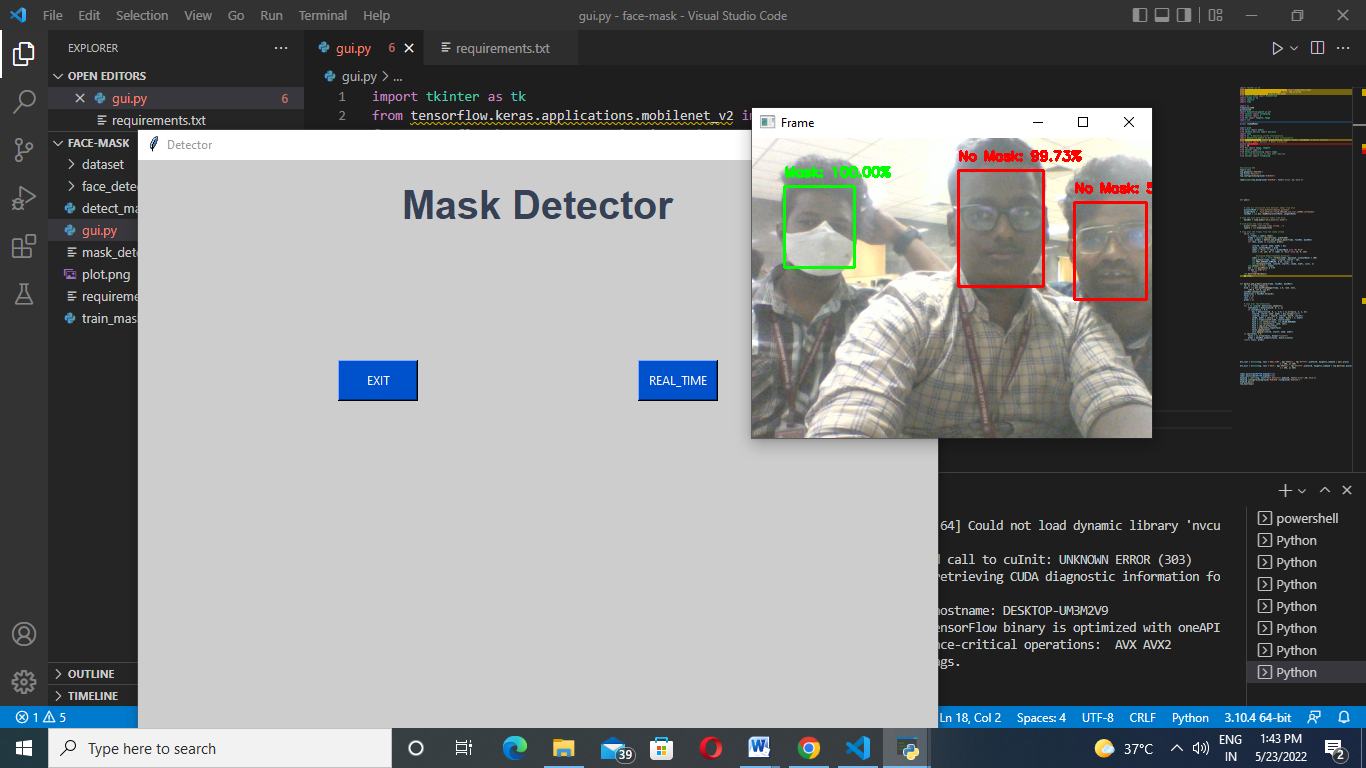
****

**Fig 9.1.2 UI interface of face mask detection click real time**

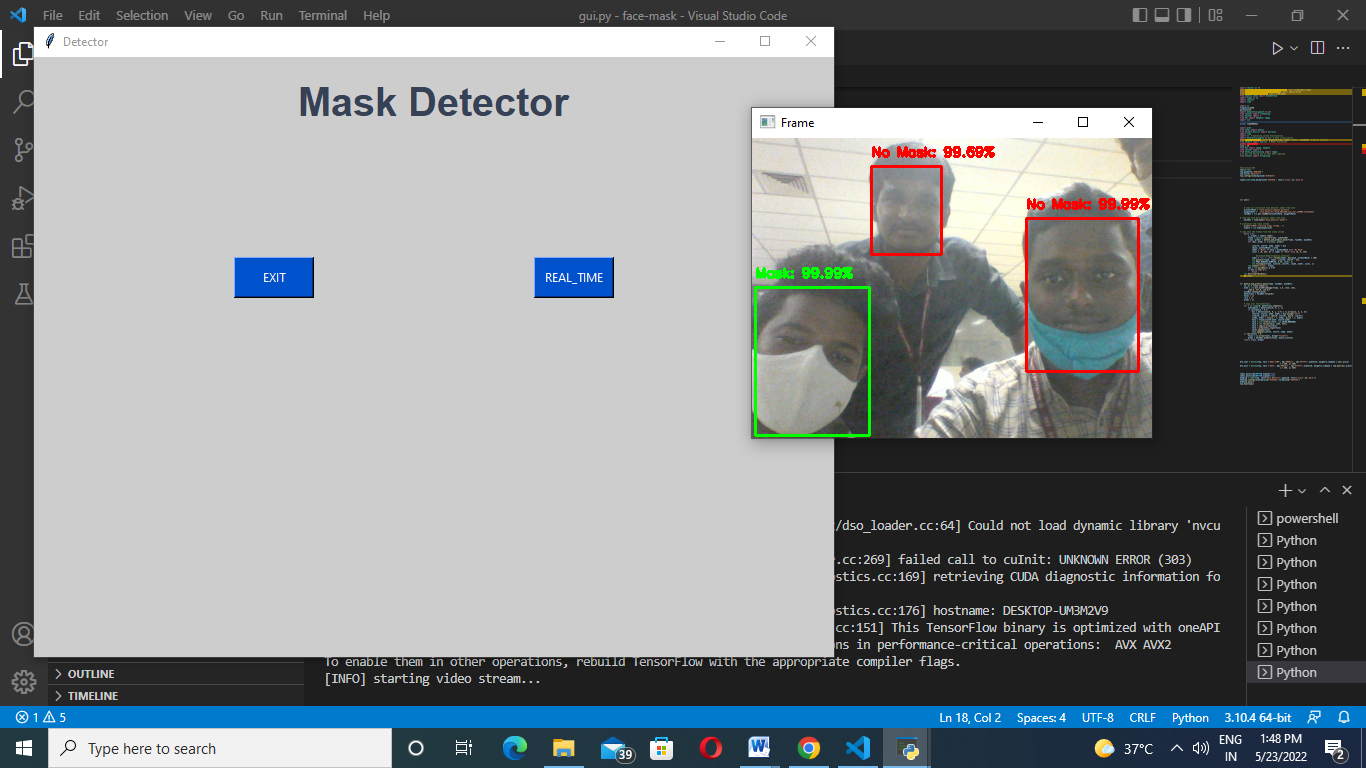
**Fig 9.1.3 Single mask prediction**

****

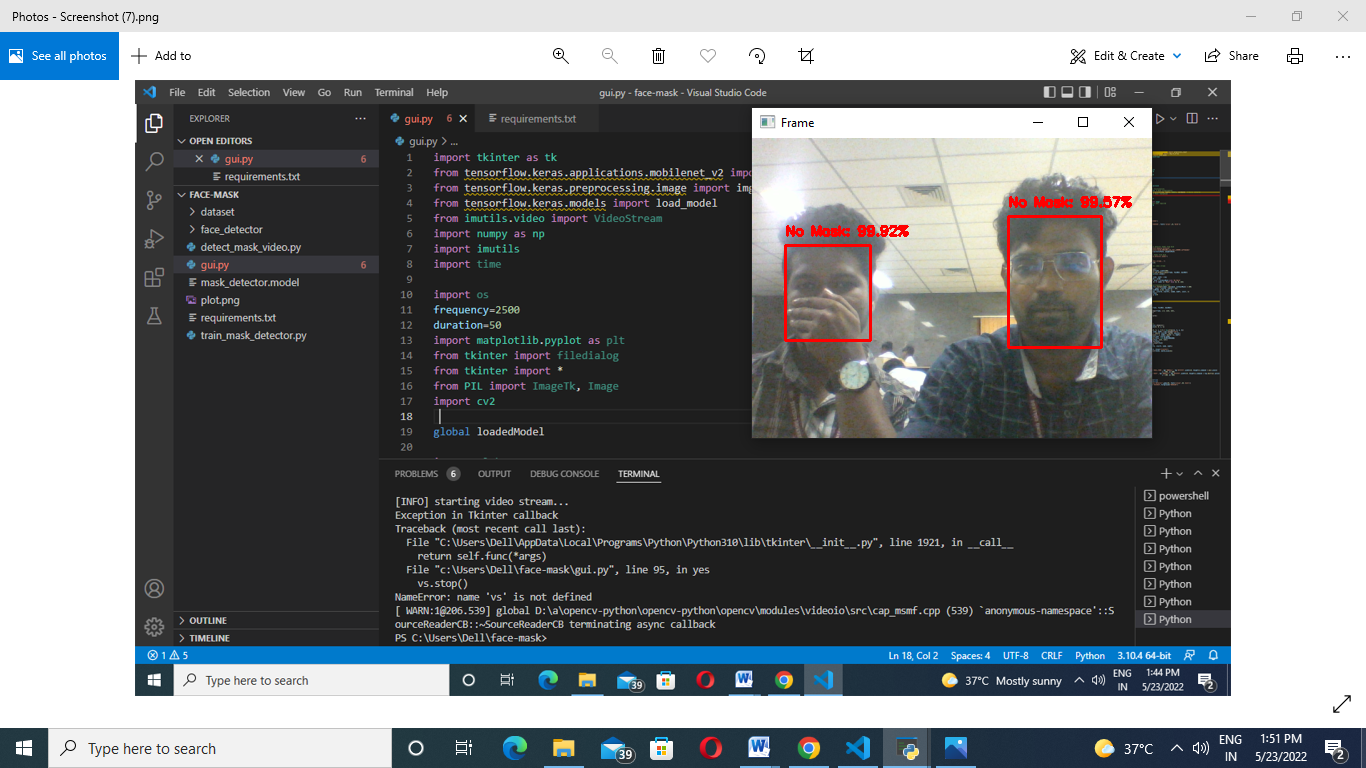
**Fig 9.1.4 Single no mask prediction**

****

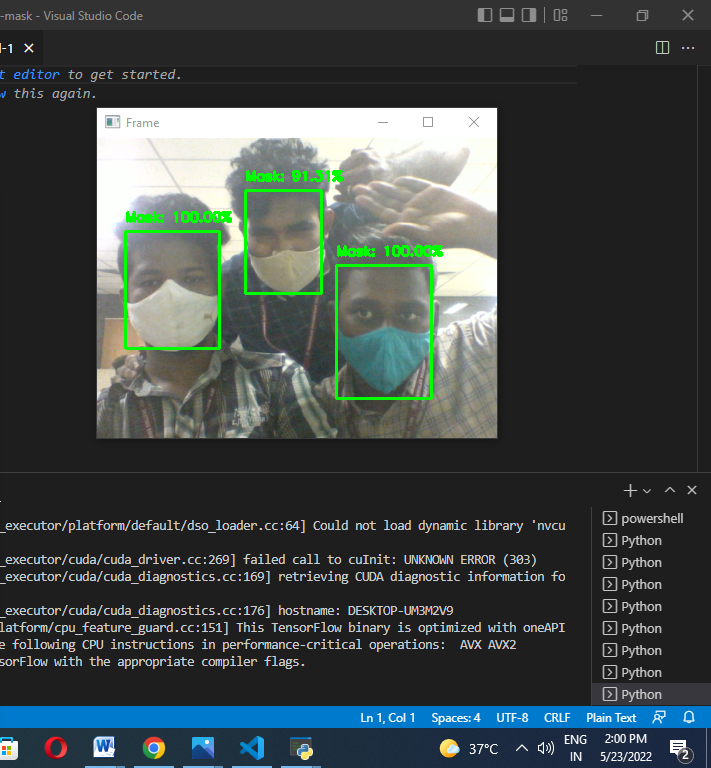
**Fig 9.1.5 It shows mask prediction of multiple persons**

****

**Fig 9.1.6 It shows mask prediction of multiple persons**

****

**Fig 9.1.7 It shows no mask while face is covered with hand**

****

**Fig 9.1.8 It shows wearing mask while everyone wears the mask**

**BIBLOGRAPHY**

**REFERENCES**

[1] Guangzhou, Wei Feng, Jintao Jin, Qujiang Lei Xiuhao Li, Guangchao Gui, Weijun Wang. “Face Mask Recognition System with YOLOV5 Based on Image Recognition” in IEEE conference location Chengudu, china, 14 DEC 2020.

[2] Bin Xue, Jianpeng Hu, Pengming Zhang. “Intelligent detection and recognition system for mask wearing based on improved RetinaFace algorithm” in [2020 2nd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)](https://ieeexplore.ieee.org/xpl/conhome/9360928/proceeding).

[3] Samuel Ady Sanjaya, Suryo Adi Rakhmawan. “Face Mask Detection Using MobileNetV2 in The Era of COVID-19 Pandemic” in 2[020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)](https://ieeexplore.ieee.org/xpl/conhome/9325571/proceeding)

[4] Arjya Das, Mohammad Wasif Ansari, Rohini Basak.”Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV” on  [2020 IEEE 17th India Council International Conference (INDICON)](https://ieeexplore.ieee.org/xpl/conhome/9341296/proceeding)

[5] Harish Adusumalli, D. Kalyani; R.Krishna Sri, M. Pratapteja, P.V.R.D Prasada Rao. “Face Mask Detection Using OpenCV” First published on 31 march,2021

[6] Sammy V. Militante, Nanette V. Dionisio. “Real-Time Facemask Recognition with Alarm System using Deep Learning” in [2020 11th IEEE Control and System Graduate Research Colloquium (ICSGRC)](https://ieeexplore.ieee.org/xpl/conhome/9223048/proceeding).

[7] Bin Xue, Jianpeng Hu, Pengming Zhang. “Intelligent detection and recognition system for mask wearing based on improved RetinaFace algorithm” in [2020 2nd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)](https://ieeexplore.ieee.org/xpl/conhome/9360928/proceeding).

[8] Susanto Susanto, Febri Alwan Putra, Riska Analia, Ika Karlina Laila Nur Suciningtyas. “The Face Mask Detection For Preventing the Spread of COVID-19 at Politeknik Negeri Batam” in  [2020 3rd International Conference on Applied Engineering (ICAE)](https://ieeexplore.ieee.org/xpl/conhome/9350316/proceeding)

[9] Samuel Ady Sanjaya, Suryo Adi Rakhmawan. “Face Mask Detection Using MobileNetV2 in The Era of COVID-19 Pandemic” in 2[020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)](https://ieeexplore.ieee.org/xpl/conhome/9325571/proceeding)

[10] V Nithyashree, S Roopashree, Aparna Duvvuri, L Vanishree, Disha Anand Madival, G Vidyashree. “A Solution to Covid-19: Detection and Recognition of Faces with Mask” in [2021 International Conference on Intelligent Technologies (CONIT)](https://ieeexplore.ieee.org/xpl/conhome/9497779/proceeding) on 27 june 2021