**B.Tech. Project Report**

**ITPE-40**

**On**

**Face Mask Detection**

**By**

**Mayank Yadav**

**Vikash Yadav**

**Kyare Sachin**

**Under the Supervision of**

**[Prof. Minakshi Sharma]**



**DEPARTMENT OF COMPUTER ENGINEERING**

**NATIONAL INSTITUTE OF TECHNOLOGY**

**KURUKSHETRA-136119, HARYANA(INDIA)**

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **Section No.** | **Title** | **Page No.** |
|  | ABSTRACT |  |
|  | OBJECTIVE |  |
|  | LIST OF FIGURES AND TABLE |  |
| 1 | INTRODUCTION |  |
|  | 1.1 PROBLEM DEFINATION |  |
|  | 1.2 OBJECTIVE AND SCOPE |  |
|  | 1.3 SYSTEM REQUIREMENT |  |
| 2 | SRS AND DESIGN |  |
|  | 2.1 INTRODUCTION |  |
|  | 2.1.1 PURPOSE |  |
|  | 2.1.2 SCOPE |  |
|  | 2.1.3 REFERENCES |  |
|  | 2.2 OVERALL DESCRIPTION |  |
|  | 2.2.1 PROJECT PERSPECTIVE |  |
|  | 2.2.2 PROJECT FUNTIONS |  |
|  | 2.3 ASSUMPTIONS AND DEPENDENCIES |  |
|  | 2.4 NON\_FUNTIONAL REQUIREMENTS |  |
|  | 2.4.1 PERFORMANCE REQUIREMENT |  |
| 3 | PROJECT ANALYSIS AND DESIGN |  |
|  | 3.1 METHODOLOGY ACCEPTED |  |
|  | 3.2 ALGORITHM |  |
|  | 3.3 FLOWCHART DIAGRAM |  |
|  | 3.6 ARCHITECTURAL DIAGRAM |  |
| 4 | SYSTEM DESIGN |  |
|  | 4.1 USER OF THE SYSTEM |  |
|  | 4.2 MODULARITY CRITERIA |  |
|  | 4.3 DESIGN METHODOLOGIES |  |
|  | 4.4 USER INTERFACE LAYOUT |  |
| 5 | PROJECT IMPLEMENTATION AND TESTING |  |
|  | 5.1 TOOLS/SCRIPTS FOR IMPLEMENTATION |  |
|  | 5.2 MODULE HIERARCHY |  |
|  | 5.3 TESTING |  |
|  | 5.3.1 TESTING THE MODEL |  |
|  | 5.3.2 MODEL EVALUATION |  |
|  | 5.4 TESTING TYPES |  |
|  | 5.4.1 UNIT TESTING |  |
|  | 5.4.2 INTEGRATION TESTING |  |
|  | 5.4.3 SYSTEM TESTING |  |
|  | 5.5 SNAPSHOT OF UI |  |
|  | 5.5.1 WITHOUT MASK |  |
|  | 5.5.2 WITH MASK |  |
| 6 | FUTURE ENHANCEMENT |  |
| 7 | LIMITATIONS |  |
| 8 | CONCLUSION |  |
| 9 | BIBLIOGRAPHY |  |

**Abstract**

The coronavirus disease has a serious impact on the whole world since 2019. Some of the measures that people should take at this time is wearing masks in public places. An accurate mask detector system is needed to ensure that people follow the COVID-19 guidelines, and in this way it would hopefully help reduce the number of corona cases.

A device called Face Mask Detection will be able to tell whether someone is wearing a mask during a COVID-19 circumstance. Additionally, it recognises where the face should be placed and offers a real-time percentage of how well the mask is being worn. While necessary, this can be utilised in open sections of offices. It will be beneficial when entering stores, airports, train stations, offices, schools, or other public places. In this project, we use image processing to create a mask detector. Using Deep Convolutional Neural Network - MobileNet v2 light architectures, images are segmented and classified.

**Objectives**

* to suggest a technique that will minimise the work required by officials or the authorities to find persons who aren't wearing face masks.
* The implementation of a training model that uses deep learning techniques to learn the face mask features from the dataset is the major goal of the system's development.
* Give a precise split of the number of people who are or are not wearing masks.

**LIST OF FIGURS AND TABLE**

|  |  |  |
| --- | --- | --- |
| **Fig no.** | **Figure name** | **Page no** |
| **3.2.1** | Steps in building the model |  |
| **3.2.1** | Convolutional Neural Network |  |
| **3.3.1** | Flowchart Diagram |  |
| **3.4.1** | Architectural Diagram (Pipe-Filter) |  |
| **5.3.1** | Testing the Model |  |
| **5.3.2** | Model Evaluation |  |
| **5.5.1** | Without Mask |  |
| **5.5.2** | With Mask |  |

**1.Introduction**

The goal of the face mask recognition project is to figure out whether or not someone is wearing a mask and where their face is positioned. The issue has a tight connection between generic object detection, which seeks to identify classes of items, and face detection, which seeks to identify a particular class of objects, the face. Applications for object and face identification can be found in a variety of settings, including security, offices, airports, train stations, and other places.

Additionally, it offers a percentage that indicates how accurately the person is wearing a mask. Webcam video is used as the input source in this case, and OpenCV is used to process the video such that the image is accurately detected, captured from the frame, processed, and output.

Face mask identification in

1.1 Problem Definition

The WHO reported that numerous parties have made attempts to stop the COVID-19 virus's spread since it was classified as a pandemic. Maintaining physical separation and using a mask in public to stop the spread of COVID-19. The procedure of keeping an eye on a sizable populace that practises varied habits in the nation, however, presents some challenges for the authorities. To properly control how the law is applied, authorities require a solution, which begins with quick and precise data availability. In order to distinguish between those who wear masks and those who do not, one method is to employ regionally automated face mask detection.

**1.2 Objective and Scope**

* whether to identify the person behind the mask
* Give a precise breakdown of the number of people wearing masks or not.

**1.3 System Requirements:**

**Hardware Requirements:**

|  |  |  |
| --- | --- | --- |
|  | **Minimum** | **Recommended** |
| **Graphic card** | AMD Radeon R5 M435 | NVIDIA GeForce GTX 1650 |
| **Processor** | Intel Core i5-8265U 160GHz | AMD Ryzen 53550H with Radeon Vega |
| **Disk Capacity** | 1 Terabytes | 1 Terabytes |
| **RAM Capacity** | 8 Gigabytes | 16 Gigabytes |

**Software Requirements:**

|  |  |
| --- | --- |
| Operating System | Microsoft Windows 10 |
| Documentation Tool | Microsoft Word, Web Browser |
| Software | Python 3.8.7, Anaconda3 |
| Coding Languages | Python |
| Library | TensorFlow, Kera |

**Feasibility Study:**

* **Technology Feasibility**

Technical: Python 3.8.7, Anaconda3

Requirements of the Libraries:

1. bsl-py==0.12.0
2. astunparse==1.6.3
3. cachetools==4.2.2
4. certifi==2020.12.5
5. chardet==4.0.0
6. cycler==0.10.0
7. flatbuffers==1.12
8. gast==0.4.0
9. google-auth==1.30.0
10. google-auth-oauthlib==0.4.4
11. google-pasta==0.2.0
12. grpcio==1.34.1
13. h5py==3.1.0
14. idna==2.10
15. imutils==0.5.4
16. joblib==1.0.1
17. keras-nightly==2.5.0.dev2021032900
18. Keras-Preprocessing==1.1.2
19. kiwisolver==1.3.1
20. Markdown==3.3.4
21. matplotlib==3.4.2
22. numpy==1.19.5
23. oauthlib==3.1.0
24. opencv-python==4.5.2.52
25. opt-einsum==3.3.0
26. Pillow==8.2.0
27. protobuf==3.17.0
28. pyasn1==0.4.8
29. pyasn1-modules==0.2.8
30. pyparsing==2.4.7
31. python-dateutil==2.8.1
32. requests==2.25.1
33. requests-oauthlib==1.3.0
34. rsa==4.7.2
35. scikit-learn==0.24.2
36. scipy==1.6.3
37. six==1.15.0
38. sklearn==0.0
39. tensorboard==2.5.0
40. tensorboard-data-server==0.6.1
41. tensorboard-plugin-wit==1.8.0
42. tensorflow==2.5.0
43. tensorflow-estimator==2.5.0
44. termcolor==1.1.0
45. threadpoolctl==2.1.0
46. typing-extensions==3.7.4.3
47. urllib3==1.26.4
48. Werkzeug==2.0.1
49. wrapt==1.12.1

**Operational Resources:**

1 Machine

1 Developer

1 Tester

Laptop AMD Ryzen 5 5600H FHD IPS Gaming Lenovo Ideapad Gaming 3 (8GB/512GB SSD/4GB NVIDIA GTX 1650/120Hz/Win 11)

**SRS and Design**

**2.1 Introduction**

**2.1.1. Purpose**

This document's goal is to give a quick summary of the specifications and requirements for the Face Mask Detection System Using Deep Learning project.

The project's objective is to automatically recognise face masks.

**2.1.2. Scope**

The Face Mask Detection system will allow the authorities to detect the people wearing mask or not who all are not following the guidelines. It will also provide the accuracy percentage of A mask is worn by the individual or not.

**2.1.3. References**

IEEE Standard

* <https://ieeexplore.ieee.org/document/9325631>

**2.2. Overall Description**

**2.2.1 Project perspective**

2019's coronavirus illness has had a significant impact on the entire world. Wearing masks in public places is one of the primary ways to protect individuals. Additionally, a lot of utility companies demand that users only utilise the service when appropriately wearing masks. However, there aren't many studies that use image analysis to detect face masks.

**2.2.2 Project Functions**

• Pre-processing

• Training the Images

• Face Mask Detection

**2.3 Assumptions and Dependencies**

**Assumptions**

It is assumed that the datasets we will utilise for the system's initial entry during training are the right inputs. The training data is obtained from web databases like Kaggle.

**Dependencies**

• Python

• Tensorflow – keras

**2.4 Nonfunctional Requirements**

**2.4.1. Performance Requirements**

1. The user must not move his/her face out of camera’s sight in order to get correct results.

2. The background must not be too bright or too dark while detecting the face mask.

3. The system will be implemented in Python script with an accuracy of the model of over 90%.

**3.Project Analysis and Design**

**3.1. Methodology Accepted**

The best suitable model for this project is Iterative model.

**Iterative Model:**

In this project, we go through training and testing phases during development.

Each iteration in an iterative development process develops, alters, and expands. The software architecture remains steady because each iteration improves on the one before it.

**3.2. Algorithm**

The system's main contributions are:

Make a training model using MobileNetV2 and machine learning techniques that learns the dataset's face mask attributes.

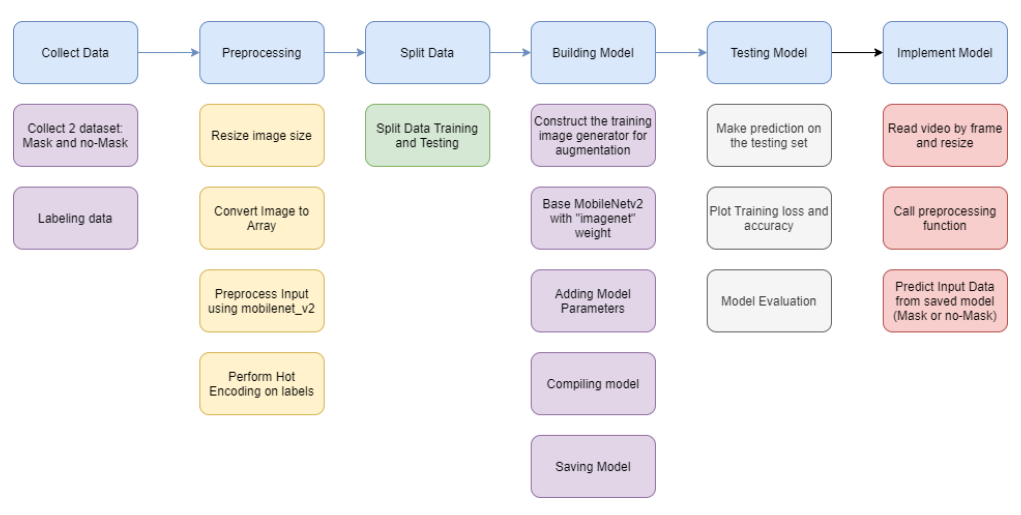
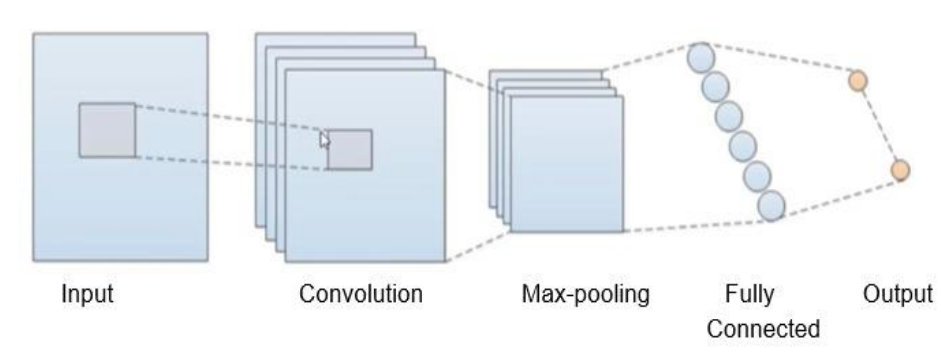


Fig3.2.1  **steps in building the model**

**Convolutional Neural Network:**

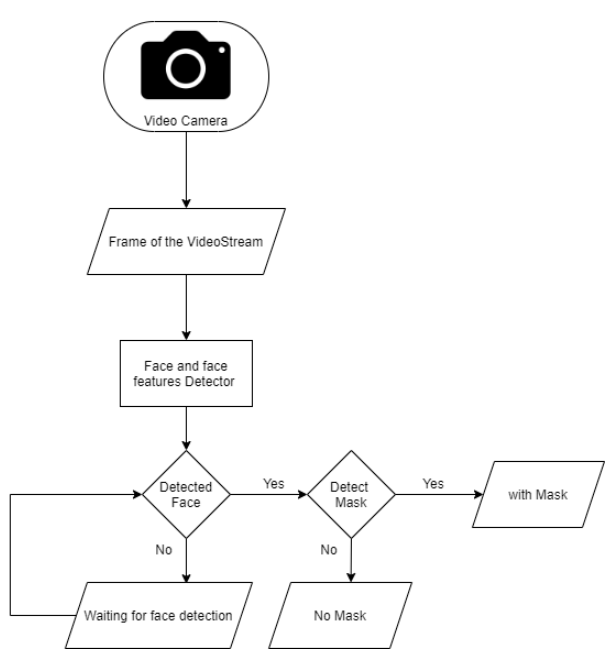


**fig3.2.2 CNN**

An input layer, hidden layers, and an output layer make up a convolutional neural network. Any intermediate layers in a feedforward neural network are referred to as hidden layers since the activation function and final convolution hide their inputs and outputs. The layers that carry out convolutions are among the hidden layers of a convolutional neural network. The convolution procedure produces a feature map when the convolution kernel moves across the input matrix for a layer, and this feature map then becomes part of the input for the following layer. Following this are further layers like normalising, pooling, and fully linked layers.

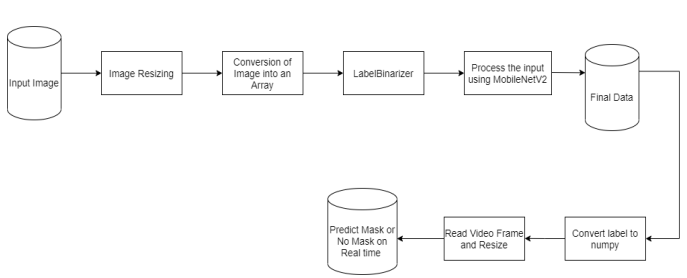
A class of neural networks known as convolutional neural networks (CNN or ConvNet) is particularly adept in processing data with a grid-like architecture, such as a

**3.3. Flowchart Diagram**



**Fig 3.3.1 Flowchart of Face Mask Detection**

**3.4. Architectural Diagram (Pipe-Filter)**

**fig 3.4.1 Pipe-Filter Architectural Diagram for Face Mask Detection**

**4.System Design**

**4.1 User of the System**

User: The individual using the system.

**4.2 Modularity criteria**

The proposed system has following modules:

➢ data collecting

➢ pre-processing

➢ split the data

➢ building the model

➢ testing the model

➢ implement the model

**4.3 Design Methodologies**

**Convolutional Neural Network**

Convolutional Neural Network (ConvNet/CNN) is a deep learning algorithm that can

input image, assigning importance (learnable weights and biases) to different aspects/objects in the image

and be able to distinguish one from the other.

➢ Availability of GPUs that speed up the model training process

➢ Availability of the data file. For the task of image captioning, CNN is widely used because of the success of CNN when used in image annotation problems. CNN has successfully solved image annotation problems with high accuracy.

**4.4 User Interface Layouts**

User interface (UI) design, or user interface engineering, is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with a focus on maximizing usability and user experience. The goal of user interface design is to simplify and streamline user interaction as much as possible in terms of achieving user goals (user-centered design).

Good user interface design makes it easy to complete a task without drawing unnecessary attention to itself. Graphic design and typography are used to support its usability, influence how the user performs certain interactions, and enhance the aesthetic appearance of the design; Design aesthetics can enhance or detract from users' ability to use interface features. The design process must balance technical functionality and visual elements (eg, a mental model) to create a system that is not only operational, but also usable and adaptable to changing user needs.

**5. Implementation and Testing**

**5.1 Tools/Scripts for Implementation**

**➢ Anaconda**

Anaconda is a free open-source software and it supports free GPU. we can improve your Python programming language coding skills. Develop machine learning and deep learning applications using popular libraries such as Keras , TensorFlow and OpenCV. We use Anaconda for developing the training model

**➢ With TensorFlow**

A complete machine learning open source platform is called TensorFlow. It is a vast and adaptable ecosystem of resources, frameworks, and tools that offers processes with high-level APIs. The framework has many conceptual levels, allowing you to pick the one you need to develop and implement machine learning models. For instance, the Distribution Strategy API allows you to do distributed hardware setups for big machine learning workloads, and TensorFlow Extended provides all you need for a complete production machine learning pipeline (TFX).

**➢ Keras**

On the other hand, TensorFlow, CNTK, and Theano are built upon by the high-level neural network library known as Keras. Deep learning can be easily and quickly prototyped using Keras, and it runs smoothly on CPU and GPU. This framework's code is developed in Python, which is simple to debug and extensible.

**➢ The OpenCV**

A free and open-source software library for computer vision and machine learning is called OpenCV. A standard infrastructure for computer vision applications was created with OpenCV in order to speed up the incorporation of artificial intelligence into products.

**5.2 Module hierarchy**

**➢ Dataset Collectio**n

**COCO Dataset, which provides free and vivid pictures, was used to compile this dataset.**

**➢ Preprocessing**

To turn picture objects into RGB fields, image preprocessing is used. The array size then becomes (299, 299, 3). Pre-processing is used in the caption to convert phrases that were previously in word form into a series of tokens based on a certain dictionary word index. The model has two inputs during training. The output layer of the pre-trained Inception-v3 model is deleted, and the extracted image features are output as the first input. A preprocessed description that has been turned into a sequential token index serves as the second input.

**➢ Training the Images**

**The target picture and the training images are encoded using the inception V3 model to extract CNN features. The suggested weighted likelihood goal is used to identify the CNN features of the training pictures. The trained GRU serves as a decoder during the generation step, using the CNN characteristics of the target picture as input to provide image identification.**

**➢ Face Mask Detection**

We use an object detector framework to create a backbone, neck, and heads detection network in order to effectively develop a face mask detection network. Backbone is a broad term for a feature extractor that creates feature maps from information extracted from pictures using convolutional neural networks.

**5.3 Testing**

**5.3.1 Testing the Model**

There are procedures for testing the model to ensure that it can predict data accurately. Making predictions about the testing set is the first stage. The following graphic displays the results for 20 iterations of assessing the accuracy and loss during model training:

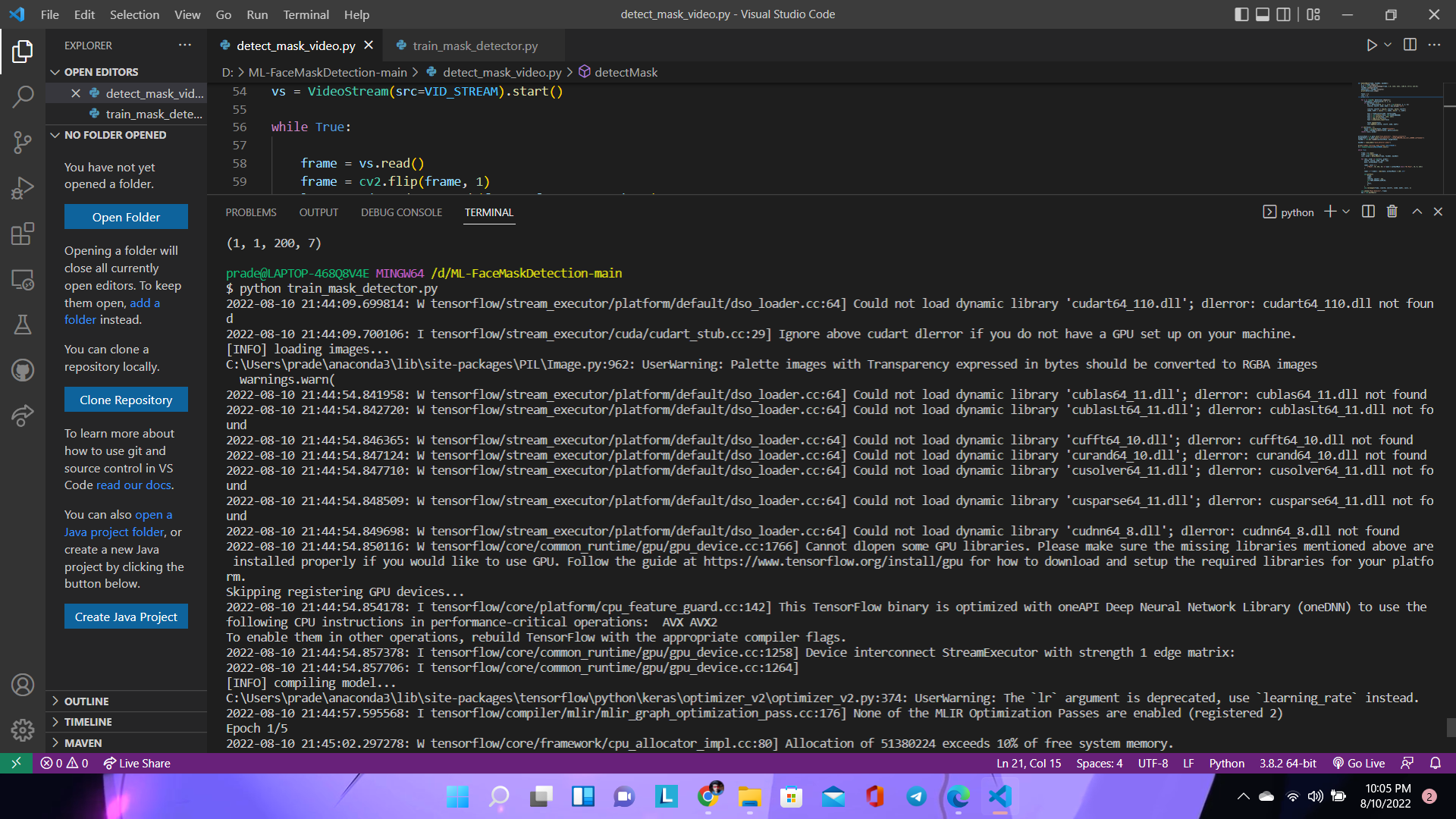
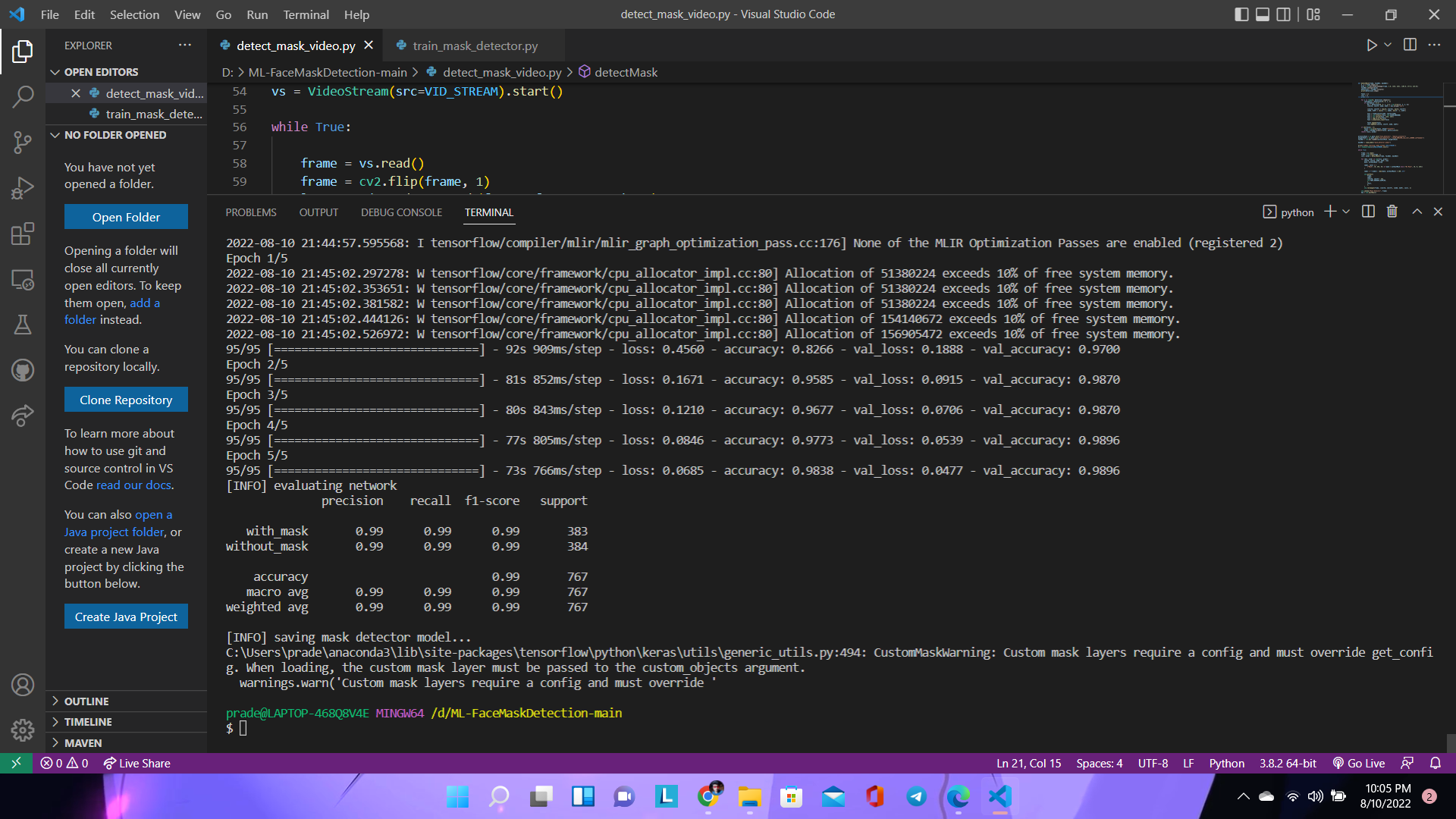


Fig 5.3.1

 Fig 5.3.2 Testing the Model

**5.3.2 Model Evaluation**

There is no longer a need for more iterations to improve the model's accuracy when the accuracy line is stable. The next stage is to evaluate the model, as shown in the image below.

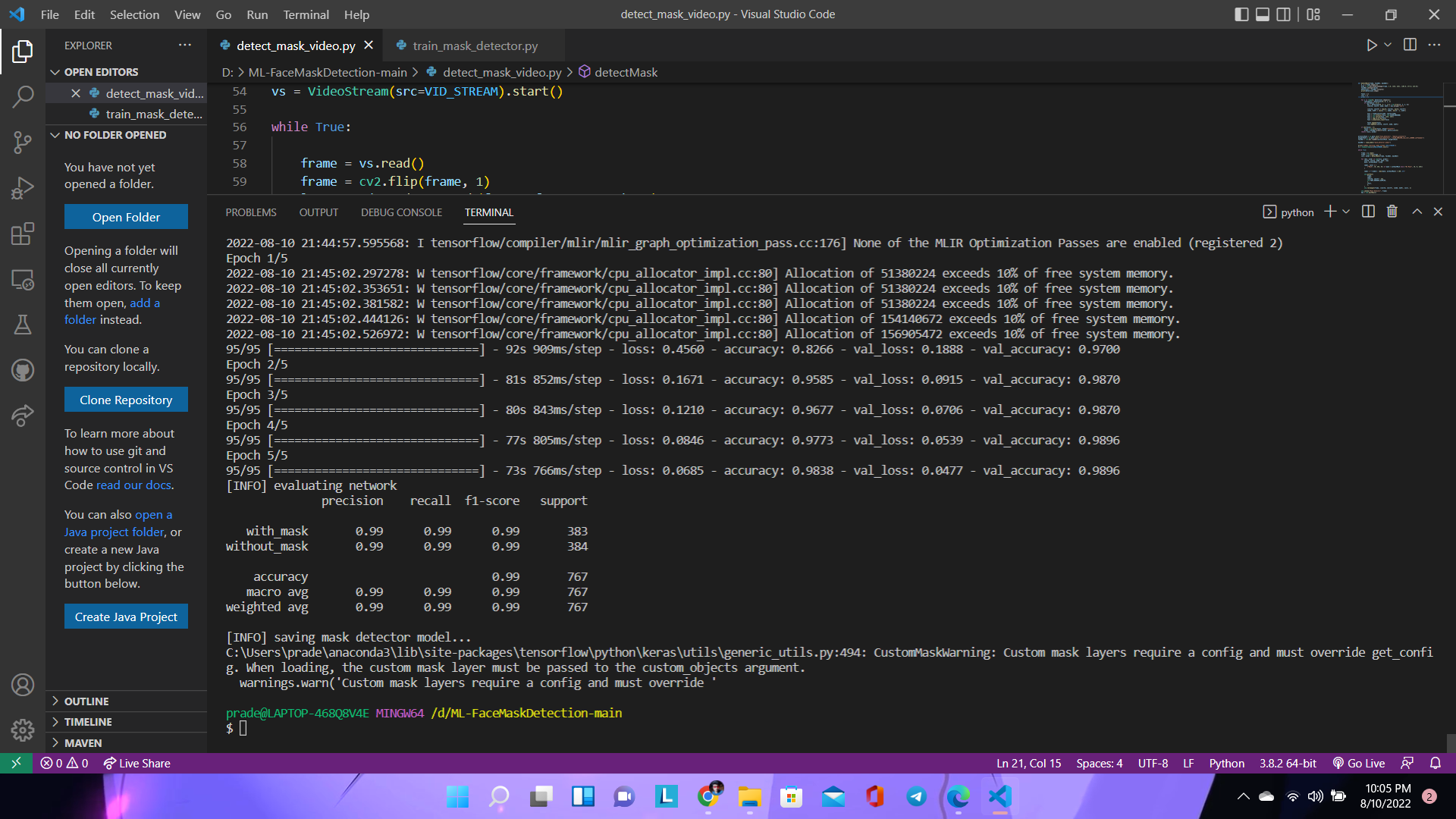


Fig5.3.2 Model Evaluation

**5.4 Testing Types**

**5.4.1 Unit testing**

Unit testing is a type of software testing used in computer programming to evaluate the suitability of individual pieces of source code, groups of one or more computer programme modules, and the related control information, use guidelines, and operating procedures. the system

➢ Test the proper functioning of preprocessing module preprocess data set

➢ Test if training images work correctly.

➢ Check to see if the model correctly detects the face mask.

**5.4.2 Integration testing**

Software testing's integration testing step, often known as integration and testing, involves combining and testing several software components together. takes place before verification testing and after unit testing. The goal of integration testing is to produce an integrated system that is suitable for system testing by taking as input modules that have undergone unit testing, grouping them into bigger aggregates, and applying the tests outlined in the integration test plan to these aggregates.

* Verify the model successfully catches the input picture
* Verify that the specified image's characteristics are rendered by the system.
* Verify that the model accurately detects the face mask.

**5.4.3 System testing**

A whole integrated system is tested using software or hardware to see whether it complies with predetermined requirements.

**5.5 Snapshot of UI**

5.5.1. Without mask:

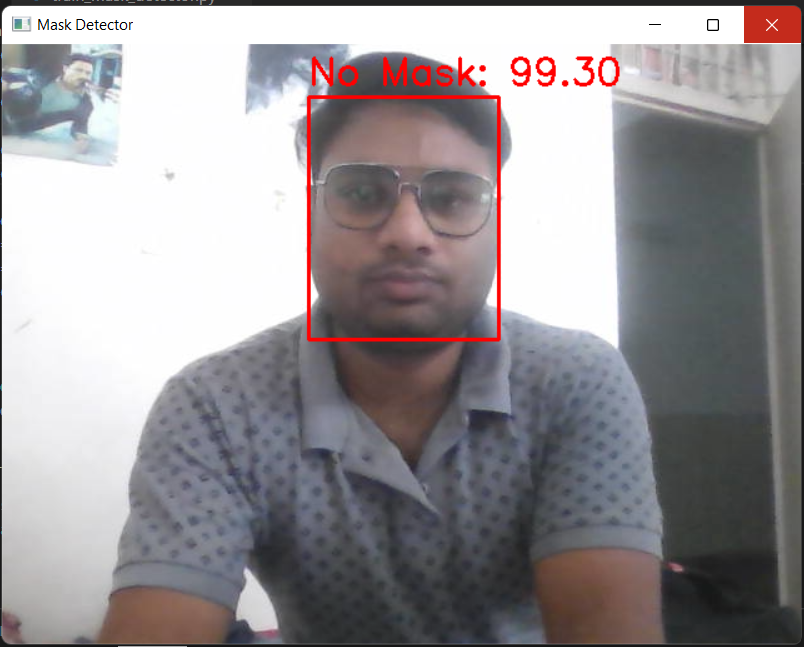


fig 5.5.1

5.5.2. With Mask:

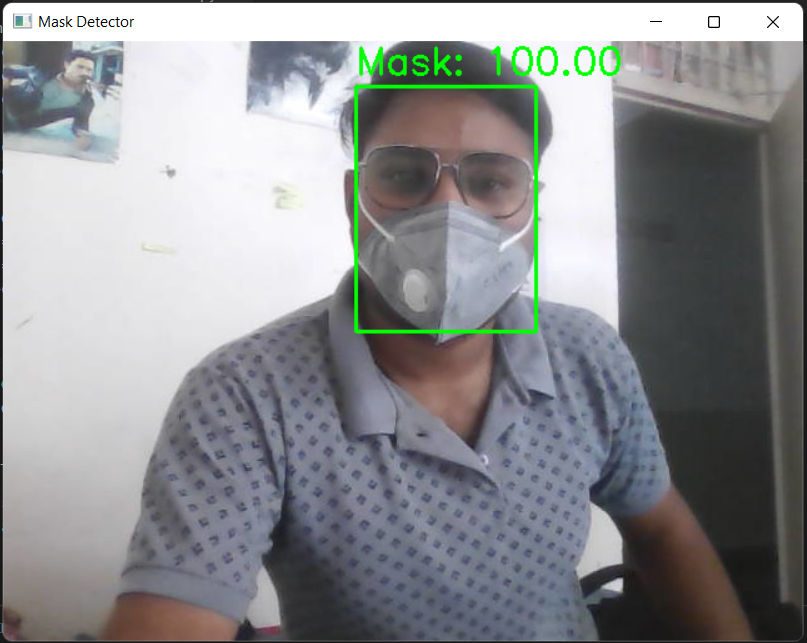


Fig 5.5.2

**6. Future Enhancement**

• We may utilise Internet of Things (IoT) technology, such as Arduino, to detect temperature, mask wear to keep track of the number of people, or an alarm system if a person is not wearing a mask. Mask detection is used to determine whether or not persons nearby the camera are wearing a mask.

• Adding more images will improve this model's accuracy. The accuracy of the system may be increased by training and the use of effective cameras.

**7. Limitations**

• Detection is a weak point. Face detection is more reliable than manual identification methods, but it is also more susceptible to errors due to changes in appearance or camera angles.

• Significant data storage burden. Face detection ML technology needs robust data storage, which may not be available to all users.

• Potential privacy violation. There are many advantages to using face recognition to assist law enforcement in finding criminals, but the same monitoring may also be used to spy on ordinary residents. To make sure that technology is utilised fairly and in compliance with people's right to privacy, stringent restrictions must be put in place.

**8. Conclusion**

This work concludes by presenting a machine learning-based face mask identification algorithm. The model can offer the proportion of persons using a face mask in several cities with good accuracy following the training, validation, and testing phases.

This study can be a simple first step for authorities to leverage unstructured data sources for more data-driven mitigation and assessment, which might be important for a statistical organisation that must act swiftly to adapt and apply machine learning and new digital data sources. Planned prevention and response to COVID-19

**9. Bibliography**

1. <https://www.kaggle.com/andrewmvd/face-mask-detection>

2. <https://ieeexplore.ieee.org/document/9325631>

3. <https://www.pyimagesearch.com/>

4. <https://stackoverflow.com/>

5. <https://scikit-learn.org/>

6.<https://keras.io/api/layers/pooling_layers/average_pooling2d/>

7. <https://keras.io/api/layers/reshaping_layers/flatten/>

8. <https://keras.io/api/layers/core_layers/dense/>

9. <https://keras.io/api/layers/activations/>

10. <https://keras.io/api/applications/mobilenet/>

11. <https://www.geeksforgeeks.org/matplotlib-pyplot-figure-in-python/>