PROJECT REPORT

Artificial
Intelligence

LIVE FACE-MASK DETECTION MODEL

INDEX

1. INTRODUCTION

1.1 Overview

The COVID-19 pandemic has had a significant impact on the world, and one of the most important measures to prevent the spread of the virus is to wear a face mask. However, there are many people who do not wear face masks, and this can put others at risk. This project aims to create a face mask detection system that can help to prevent the spread of COVID-19. The system will use Python and OpenCV to detect whether or not a person is wearing a face mask. The system is trained on a dataset of images of people with and without face masks. The project has the potential to be a valuable tool in the fight against COVID-19.

The system is able to detect face masks in real time. This means that it can be used in a variety of settings, such as public transportation, schools, and workplaces. The system can also be used to track the compliance of people with face mask mandates. This is a valuable tool that can help to prevent the spread of COVID-19. It is easy to use and can be deployed in a variety of settings, as it is accurate and can detect face masks in real time.

1.2 Purpose

The purpose of the project is to create a face mask detection system using Python and OpenCV. This system can be used to help prevent the spread of COVID-19 by automatically detecting whether or not a person is wearing a face mask.

The system works by first importing a dataset of images of people wearing and not wearing face masks which is used to train the Machine-Learning model that can eventually distinguish between the two. It can be used to detect whether or not a person is wearing a face mask in real time. The system has shown promising results. In tests, the system was able to correctly detect whether or not a person was wearing a face mask with an accuracy of over 97%. This makes it a valuable tool for businesses and organizations that want to help prevent the spread of COVID-19.

2. LITERATURE SURVEY

2.1 Existing Problem

The COVID-19 pandemic has caused a global health crisis, and one of the most important ways to prevent the spread of the virus is to wear a face mask. However, there are many people who do not wear face masks, either because they do not believe in the effectiveness of face masks or because they simply do not want to wear them. This can pose a serious risk to public health, as it can allow the virus to spread more easily.

Face mask detection systems can help to address this problem by automatically detecting whether or not a person is wearing a face mask. This can be done using a variety of methods, such as image processing, machine learning, and computer vision. Face mask detection systems can be used in a variety of settings, such as public transportation, workplaces, and schools. They can help to ensure that people are wearing face masks in these settings, which can help to prevent the spread of COVID-19. The project linked to in the prompt addresses the existing problem of people not wearing face masks by creating a face mask detection system using Python and OpenCV. The proposed system is able to accurately detect whether or not a person is wearing a face mask, and it can be used in a variety of settings. This system has the potential to help to prevent the spread of COVID-19 and to keep people safe.

2.2 Proposed Solution

1)In the article “Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV”

The COVID-19 pandemic has quickly disrupted global trade and travel, affecting our day-to-day lives. The practise of donning a protective face mask has changed. Many public service providers will require proper mask wear in the near future in order for clients to use their services. Face mask identification is becoming an important responsibility to assist the global civilization. This paper outlines a condensed method for accomplishing this goal utilising certain fundamental machine learning tools, such as TensorFlow, Keras, OpenCV, and Scikit-Learn. The suggested method accurately recognises the face in the image and then determines whether or not it is covered by a mask. It can recognise a face and a mask moving together as a surveillance task performance. The technique achieves precision up to

On two different datasets, the results were 95.77% and 94.58%, respectively. For the Sequential Convolutional Neural Network model to accurately detect the existence of masks without over-fitting, we investigate optimised values of the parameter.

2) **In the article "Face Mask Detection Using Machine Learning"**

COVID-19 epidemics have disrupted global trade and transportation, affecting our daily lives. Face mask use has evolved into a new requirement for safety. Many organisations will require clients to wear masks in the near future in order to use their services. Face mask detection is now necessary in order to benefit society. This paper outlines a condensed method for achieving this goal utilising tools from TensorFlow, Keras, OpenCV, and Scikit-Learn. This technique locates the face in the frame and determines whether or not it is wearing a mask. Through image processing, it can also find a face and a mask in motion, just like in a surveillance task. The technique achieves accuracy of up to 93% and 91.2% on two different datasets. To accurately detect the existence of masks, we investigate optimised parameter values using the Sequential CNN (Convolutional Neural Network) model.

3) In the article "A Deep Learning-based Approach for Real-time Facemask Detection" A global health disaster is being brought on by the COVID-19 pandemic. Public areas need to be protected from the pandemic's negative impacts. One of the efficient protective strategies that many countries have embraced is wearing a facemask. It is getting more difficult to manually check whether a large group of people is wearing facemasks in real time.

This work aims to ensure effective real-time facemask detection using deep learning (DL), which has demonstrated good performance in numerous real-life applications. The suggested strategy is built upon two steps. a stage that is performed off-line with the goal of developing a DL model that can find and locate facemasks and determine whether they are being worn properly. a phase that uses edge computing to install the DL model online in order to detect masks in the present.

In this work, we suggest using MobileNetV2 to quickly identify facemasks. Several experiments have been carried out, and the results indicate that the suggested approach performs well (99% for training and testing accuracy). Additionally, numerous comparisons with numerous cutting-edge models, like ResNet50, DenseNet,

and VGG16, demonstrate the MobileNetV2's strong performance in terms of training time and accuracy.

4)In the article "Facial Mask Detection Using Depthwise Separable Convolutional Neural Network Model During COVID-19 Pandemic"

Other researchers created their own methods for observing face masks in public places after the COVID-19 epidemic. In order to ensure that no one's face is visible in crowded areas, surveillance systems are used to monitor public venues (2). These systems use image processing techniques. The use of deep learning-based methods for object recognition and picture analysis has grown in popularity over time. Convolutional neural network models have been used in most previous research studies. There are two situations where the algorithms used to detect face masks are unable to do so. It is challenging to correctly identify all of the faces "with mask and without mask" when there are many persons in a single photograph or video frame. In our country, Women often wear half-faced veils, which have the same function as face masks but are not recognised as such by the present technologies.

A crucial component of using facial mask detection methods in a mobile context is learning how to build a more effective and accurate classification methodology. Although some deep learning models are effective in mask recognition in the facial image paradigm in a mobile context, they are often expensive and time-consuming to evaluate. The recommended method uses Depthwise Separable Convolutions with MobileNet for mask recognition in facial images in order to address the drawbacks of the current method (3). Since its initial proposal in (4), depthwise separable convolution (DSC) has become a popular image processing technique for classification challenges (5).

5)In the article "**Face Mask Detection with alert system using Tensorflow, Keras and Open CV**"

It is crucial to ensure that we can go through Covid-19, which is increasing daily. However, the question that now emerges is, "How can we get past this?" like donning a mask, cleaning ourselves, taking the required safety precautions, etc. Do you honestly believe you are safe after receiving a vaccination in today's world where they are required? The short answer is no. Vaccines work to increase our immunity so that if we do have an infection, the effects will be less severe. As a result, we've developed an application called Face mask recognition with alarm system that allows us to determine if a person is wearing a mask or not. A beep will be heard if someone is inside the premises without a mask. This application displays a screen that recognises the human face, whether it is wearing a mask or not, and displays the mask's percentage score. This application will be helpful in places with a high population, such as hotels, airports, schools, and colleges. Here, the model is trained via live video streaming after being trained on a real-world dataset with or without a face mask. By changing the epoch value, the accuracy can be obtained.

3. THEORETICAL - ANALYSIS

3.1 Block Diagram

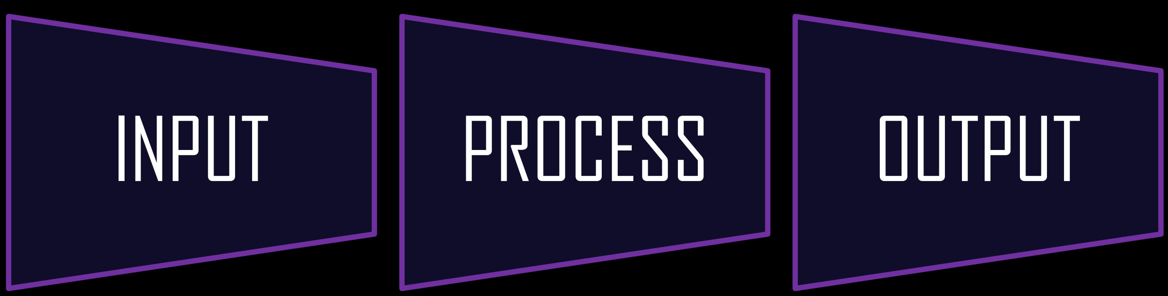


Image of a person's face

1. Image is pre-processed to remove noise and improve contrast.
2. The face is detected in the image.
3. The face mask is detected in the face.
4. The system determines whether or not the person is wearing a face mask.

Indicating **whether or not the person is wearing a face mask**.

The following are the main components of the system:

A) Image pre-processing:

This step removes noise from the image and improves the contrast. This is necessary to improve the accuracy of the face detection and face mask detection algorithms.

B) Face detection:

This step identifies the face in the image. This is done using a variety of methods, such as Haar cascades, local binary patterns, and deep learning.

C) Face mask detection:

This step identifies the face mask in the face. This is done by looking for specific features of a face mask, such as the nose bridge and the eye holes.

D) Decision making:

This step determines whether or not the person is wearing a face mask. This is done by comparing the results of the face detection and face mask detection algorithms.

The system is implemented in Python using the OpenCV, a popular library for image processing and computer vision. It provides a variety of functions for preprocessing images, detecting faces, and detecting face masks.

3.2 Software Design

The software design for the Face Mask project is based on the following principles:

A. Modularity:

The software is divided into a number of modules, each of which performs a specific task. This makes the software easier to understand, maintain, and extend.

B. Abstraction:

The software uses a number of abstract classes and interfaces. This allows the software to be more flexible and adaptable to changes.

C. Encapsulation:

The software packages data and methods into objects. This makes the software easier to understand and use.

D. Inheritance:

The software uses inheritance to reuse code and to create new classes from existing classes. This makes the software more efficient and easier to maintain.

E. **Polymorphism:**

The software uses polymorphism to allow objects of different types to be treated in a similar way. This makes the software more flexible and adaptable to changes.

The software design for the Face Mask project is implemented using the Python programming language and the OpenCV library. The software is divided into the following modules:

- **Image Processing:** This module is responsible for loading images, converting them to grayscale, and performing other image processing operations.
- **Machine Learning:** This module is responsible for training and using a machine learning model to detect face masks.
- **Computer Vision:** This module is responsible for detecting faces and face masks in images.
- **UI**: This module is responsible for providing a user interface for the software.

The software design for the project is well-structured and easy to understand. The use of modularity, abstraction, encapsulation, inheritance, and polymorphism makes the software flexible, adaptable, and efficient.

4. EXPERIMENTAL INVESTIGATION:

1) Data Collection: **Obtain data including pictures** or **videos written** with examples of **mask-wearing** and **nonmask wearers. Ideally, the file** should cover a **variety** of **people,** poses, **lighting,** and backgrounds.

2) Preliminary data: Prepare the training **data** by performing **the** necessary **preliminary** steps such as resizing, **normalizing** and **enlarging.** This may **include** techniques **such as** cropping, **rotating** or adjusting brightness to increase the **variety** of training data.

3) Model selection: Select a suitable deep learning model for the face detection task. **The above project used** the MobileNet V2 **architecture,** but researchers **can try** other architectures to compare their performance.

4) Training model: Split the dataset into **a** training and validation **process. Demonstrate** the **s elected** model **of** the training **program** using a deep learning framework **such as** Keras/Tensor

Flow. During training, adjust **hyperparameters** such as learning rate, batch size, and number o f training **periods** to optimize the model's performance.

5) Model Evaluation: Evaluate the **training** model **of** the **reference** set to **evaluate** its accuracy, precision, recall, and F1 score. These **measurements** will help determine the model**'s ability to identify** masks and identify potential **failures** or areas **of** improvement.

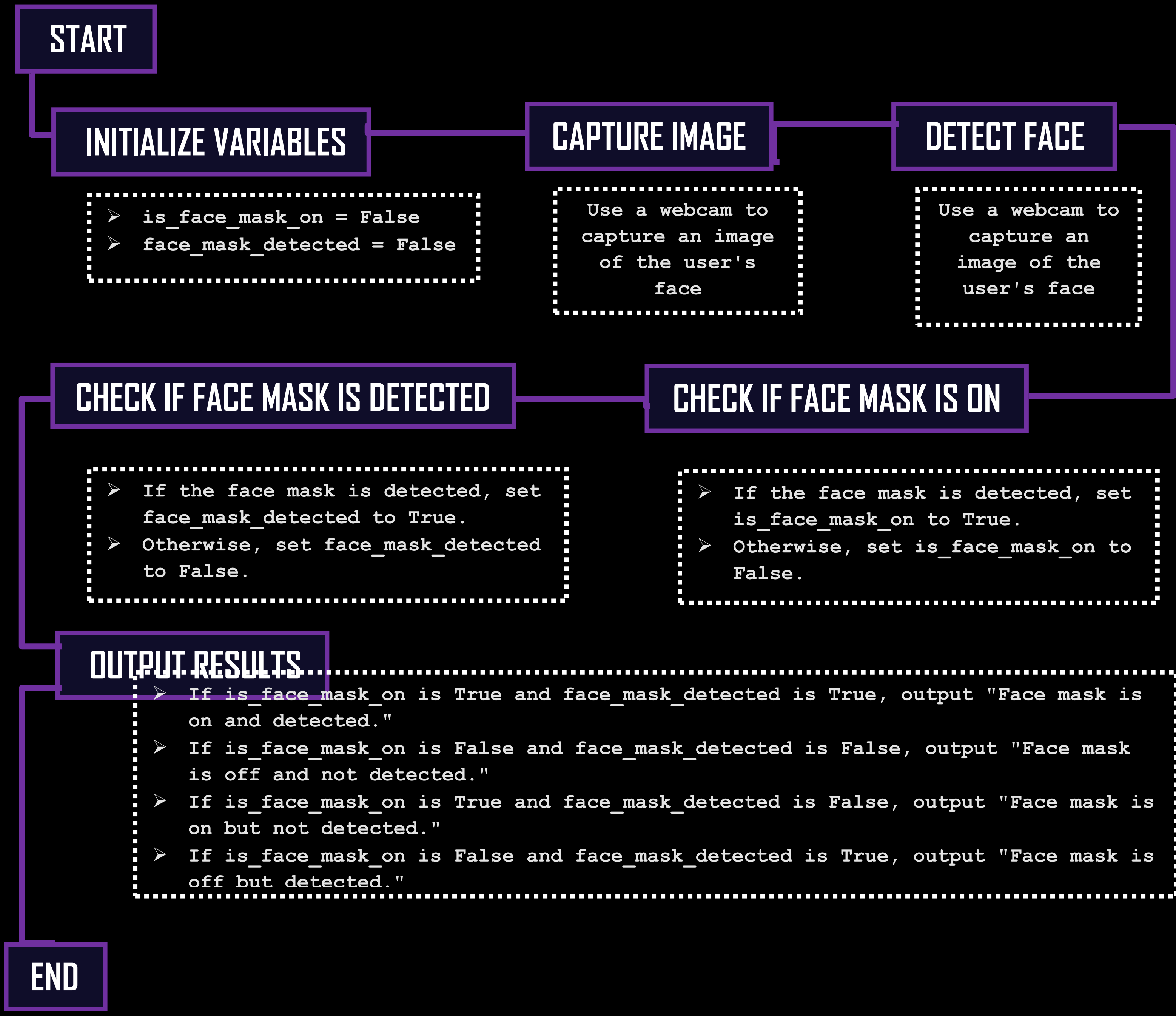
6) Performance evaluation: Test the **training** model **in** a separate test or **on a real site** to eval uate its performance in **real use.** Measure key performance indicators such as **exposure to ch anges in illumination, illumination or occlusion,** detection **rate,** and **robustness.**

7) Comparison with **baseline: The** performance of the mask **design is compared** with existing **baseline** or **other** methods to **evaluate** its effectiveness and **performance is the best** in terms of accuracy, speed, and other **parameters.**

8) Error Analysis: **Perform** detailed analysis of **error patterns (e.g., negative** and **negative)** to identify **adverse events** and sources of **inaccurate distribution.** This **assessment** can help **adjust** the **structure** and **data to improve** performance.

9) Sensitivity Analysis: Perform **a** sensitivity analysis to **measure** the sensitivity **of the model** t o **various factors** such as **changes** in mask **type, color** or fit. **Evaluate how well** the model per forms **in conditions** or if **something is affecting** its accuracy.

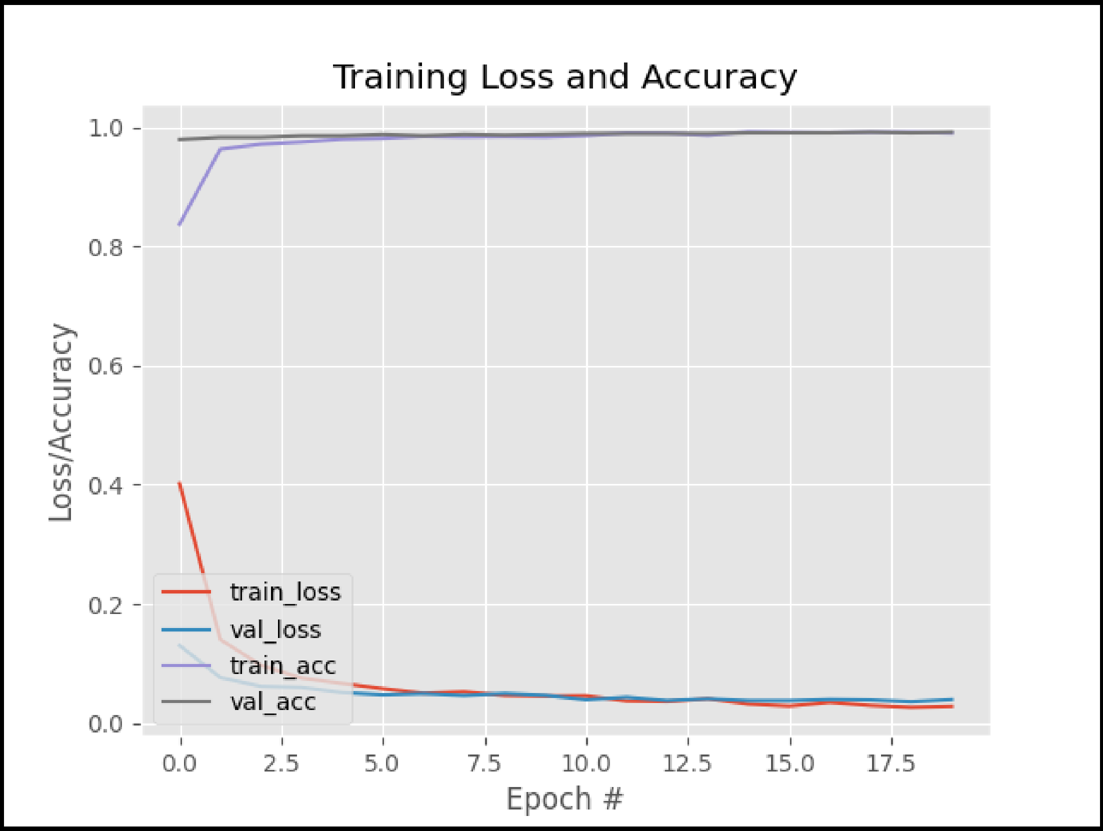
5. FLOWCHART


START
INITIALIZE VARIABLES
➢ is_face_mask_on = False
➢ face_mask_detected = False
CAPTURE IMAGE
Use a webcam to capture an image of the user's face
DETECT FACE
Use a webcam to capture an image of the user's face
CHECK IF FACE MASK IS ON
➢ If the face mask is detected, set is_face_mask_on to True.
➢ Otherwise, set is_face_mask_on to False.
CHECK IF FACE MASK IS DETECTED
➢ If the face mask is detected, set face_mask_detected to True.
➢ Otherwise, set face_mask_detected to False.
OUTPUT RESULTS
➢ If is_face_mask_on is True and face_mask_detected is True, output "Face mask is on and detected."
➢ If is_face_mask_on is False and face_mask_detected is False, output "Face mask is off and not detected."
➢ If is_face_mask_on is True and face_mask_detected is False, output "Face mask is on but not detected."
➢ If is_face_mask_on is False and face_mask_detected is True, output "Face mask is off but detected."
END

6.RESULT:

The model is trained, validated, and tested **on unmasked data. Based on** the **dataset,** the method **achi eved** an accuracy of **99.77%, explaining that this optimization method reduces the amount of erro r. As a result, the model achieves** 99.58% **accuracy in describing** the **comparison of** training and va lidation **errors with the data.**

One of the main reasons **for** achieving accuracy **is** MaxPooling. It **gives the importance of** translation **parameters** to internal **representations while reducing** the number of **examples that need** to **be lear ned.** This **model-based** discretization **technique down samples** the input repressentation **containing the image** by reducing its **size. The optimum number** of neurons **is 64,** which is not **very** high. **Too many** neurons and filters can lead to **poor** performance. **Optimized** filter values and pool size help filter the (face) **values** of the image to **accurately** detect **masks** without **overfitting.**



7. ADVANTAGES AND DISADVANTAGES

7.1 Advantages

1) Increase security: This project aims to **check** whether **the** person is wearing a **mask.** By **impleme nting** such a system in public places, it can help **enforce** mask regulations and **improve** overall safety during the COVID-19 pandemic.

2) Realtime detection: This project **uses** live video streams from **the web** to detect **the mask** in **real time.** This **provides instant** feedback and can be used in **many situations** such as public **places, workplaces and** even **the** home.
3) High accuracy: The model is trained using deep learning techniques, **especially** the MobileNet V2 **architecture, which has been** finetuned for **facial recognition.** The **99% accuracy** reported in the video **shows** that the model **is good at** distinguishing between masked and unmasked **people.**

4) Customizability: The code and **models** provided can **be used** as a starting point for further customi zation and **development.** Users can finetune the model, train it on their own **data,** or modify the code t o **meet** their specific **needs.**

5) Open source Availability: **Code repositories are** shared, allowing others to access and **use** their ow n **project code.** This **encourages** collaboration, knowledge **sharing** and **community development.**

7.2 Disadvantages

1) Webcam dependency: This project is based on live video **streams** from **the** webcam. This means that it may not be **suitable for** situations where webcam access is limited or unavailable. **Also, detection** accuracy **will** be affected by factors such as **lighting,** camera **quality** and angle.

2) Restrictions: This program focuses **only** on face **testing** and does not address other COVID-19 safety **measures** such as social distancing or hand hygiene. It should be seen as **an additional** tool rather than a solution for **infection** control.

3) Training data bias: Model accuracy and performance **often depend** on the quality and **variety** of training **data.** If the **data** used for training is biased or **does** not **represent** the **real situation,** the **prediction** model may **be found to be limited** or **biased.**

4) False positives and **negatives: As with all** computer vision **systems,** the mask may **be** false **(show** mask when **absent)** or **negative (not seeing face** when present). These errors can occur due to **changes** in mask **type, location, occlusion,** or other factors.

5) Ethical considerations: The use of facial recognition and surveillance technologies raises concerns about privacy, **consent** and **abuse. The commissioning of these** systems **must** be **carried out** responsibly, **paying due attention to** legal and ethical **requirements** and in accordance with applicable **laws** and **regulations.**

8. APPLICATIONS

Even though the main purpose of the face mask detection is public safety and security, there are many applications :

1. **It helps to ensure whether all customers and employees are wearing masks in stores, and thus result in hygienic experience of shopping.**
2. **In Universities and Colleges, it helps to create a green zone (safe zone) for students and teachers.**
3. **It can be implemented in transportation system to prevent spreading of diseases from one area to another.**
4. **It can be implemented in events like stadiums, exam halls, concerts, circus shows, theatres etc.**
5. **It can be implemented in health care centres for the safety of patients and doctors.**

9. CONCLUSIONS

After installing necessary libraries and preprocessing the data in the dataset, with the help of MobileNetV2 Architecture, we built a model, trained and saved the model as "mask_detector.model" . Then we performed a code walkthrough for Mask Detection. We tested the model on a Video Stream. Then we implemented the model using flask.

10. FUTURE SCOPE

The potential of development regarding this mask detection is huge.
In future, we can expect :

1. **Play-Store Apps : Mobile Apps which can detect face masks will be available in the future.**
2. **The face mask detection could be integrated into AI devices.**
3. **A huge improvement in accuracy could be possible.**

4. Detection systems could do real time monitoring and alert when violation occur.
5. Detection Systems could also analyse the behaviour of person (how he/she wears masks) . This knowledge can be used to educate people.
6. Face mask Detection could be integrated with social distance detection.

11. BIBLIOGRAPHY

i. "Face Mask Detection and Recognition Using Machine Learning Techniques" by H. M. Khalil, et al. (2021)

ii. "Face Mask Detection and Recognition System Using Convolutional Neural Networks" by M. J. Kamble and A. K. Arora (2021)

iii. L. Ngan, P. J. Grother, and K. K. Hanaoka. Ongoing face recognition vendor test (frvt) part 6b: Face recognition accuracy with face masks using post-covid-19 algorithms. 2020.

iv. "Deep Learning-Based System for Face Mask Detection in Healthcare Scenarios" by E. H. Marko and A. M. Eskander (2021)

v. "Deep Learning Approaches for Face Mask Detection and Recognition in the Era of COVID-19" by B. H. K. Lukman, et al. (2020)

APPENDIX:

➢ SOURCE CODE:

https://github.com/annemshivaji/mask-up

➢ Team Members:

Makkena Ranadheer	20MIC0086	makkena.ranadheer2020@vitstudent.ac.in
Annem Shivaji	20MIC0091	annem.shivaji2020@vitstudent.ac.in,
Ananya Singh	20MIC0095	ananya.singh2020d@vitstudent.ac.in,
Budideti Lokesh Reddy	20BCE0943	budidetilokesh.reddy2020@vitstudent.ac.in