CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

A Project Report on

Real time Facemask detection and accuracy prediction

Submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

S Yoshita Manavi 1760400

Under the Guidance of

Dr Praveen Naik

Computer Science and Engineering

School of Engineering and Technology,

CHRIST (Deemed to be University),
Kumbalagodu, Bengaluru - 560 074

May-2021

CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

School of Engineering and Technology

Department of Computer Science Engineering

CERTIFICATE

This is to certify that **S Yoshita Manavi** has successfully completed the project work entitled "**Real time Face mask detection and accuracy prediction**" in partial fulfillment for the award of **Bachelor of Technology** in **Computer Science and Engineering** during the year **2020-2021**.

Dr PRAVEEN NAIK
Assistant Professor

Dr Balachandran K
Head of the Department

Dr Iven Jose
Dean

CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

School of Engineering and Technology

Department of Computer Science Engineering

BONAFIDE CERTIFICATE

It is to certify that this project titled "**Real time Face mask detection and accuracy prediction**" is the bonafide work of

Name	Register Number	Department
S Yoshita Manavi	1760400	Computer Science and Information Technology

Examiners [Name and Signature]

1.

2.

Name of the Candidate :

Register Number :

Date of Examination :

Acknowledgement

I would like to thank **Dr Rev. Fr. Abraham V M**, Vice Chancellor, CHRIST (Deemed to be University), **Dr Rev. Fr. Joseph CC**, Pro Vice Chancellor, **Dr Fr. Benny Thomas**, Director, School of Engineering and Technology and **Dr Iven Jose**, Dean for their kind patronage.

I would also like to express sincere gratitude and appreciation to **Dr Balachandran K**, Head of the Department, Department of Computer Science Engineering for giving me this opportunity to take up this project.

I also extremely grateful to my guide, **Dr PRAVEEN NAIK**, who has supported and helped to carry out the project. His constant monitoring and encouragement helped me keep up to the project schedule.

Declaration

I, hereby declare that the project titled "**Real time Face mask detection and accuracy prediction**" is a record of original project work undertaken for the award of the degree of **Bachelor of Technology** in **Department of Computer Science Engineering**. I have completed this study under the supervision of **Dr PRAVEEN NAIK**

I also declare that this project report has not been submitted for the award of any degree, diploma, associate ship, fellowship or other title anywhere else. It has not been sent for any publication or presentation purpose.

Place: School of Engineering and Technology,
CHRIST (Deemed to be University),
Bengaluru

Date: 15-06-2021

Name	Register Number	Signature
S Yoshita Manavi	1760400	

Abstract

In a context of a virulent disease that can be transmitted via sputtering, wearing a face mask seems to appear important to both the wearer and other people to limit the propagation of the disease. The COVID-19 pandemic made it necessary for people around the world to wear a mask to reduce the risk of transmission of the virus. Research indicates that wearing a face mask properly would help reduce transmission and save lives.

Since the outbreak of the Covid-19 pandemic, substantial progress in the fields of image processing and computer vision has been made in the identification of face masks. Several algorithms and techniques have been used to construct a variety of face recognition models.

In this project, I proposed a design of a deep learning model using python, TensorFlow, Keras,MobileNet and open CV.The main goal is to use the trained AI model with a dataset consisting of masked and unmasked images to identify whether the person is wearing a mask or not during a Livestream via webcam.This model can be used for safety purposes since it is very resource efficient to deploy.

***Keywords*:** openCV, keras,Tensorflow,MobileNet

Contents

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

.

Item	Description
COVID-19	Coronavirus disease of 2019
CNN	Convulutional Nueral Network
DNN	Deep Neural Network
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
ConvL	Convolutional layer
Python	A modern programming language with huge community support
Open CV	Open Source Computer Vision Library
Numpy	Numerical Python
OFR	Occluded Face Recognition
GUI	**G**raphical **U**ser **I**nterface
FL	Facial Recognition

Chapter 1

INTRODUCTION

The trend of wearing face masks has rapidly increased due to the COVID- 19 epidemic all around the world. Before the epidemic, people used to wear face masks to safeguard their health from the air pollution. Where, few have a feeling of self-consciousness about their appearance, they hide their emotions from the public by covering their faces. Scientists have proved that wearing face masks works on curbing COVID-19 transmission. COVID-19 is the latest infectious virus that hit the health of human beings in the last century. In the year 2020, the accelerated/brisk spreading of COVID-19 has compelled the WHO(World Health Organization) to declare COVID-19 as a global pandemic.

More than hundred and fifty million people were infected by COVID-19 in less than a year across 191 countries. The Covid pandemic has led to an exceptional level of overall scientific cooperation. Artificial Intelligence (AI) based on Deep Learning and Machine Learning can assist in battling Covid-19 from various perspectives. AI permits scientists and clinicians to assess tremendous amounts of information to figure the circulation of COVID-19, to fill in as an early notice system for possible pandemics, and to characterize weak populaces. The arrangement of medical care needs subsidizing for arising innovation like artificial intelligence, IoT, Big Data and AI to handle and foresee new infections. Individuals are constrained by laws to wear face masks out in the open in numerous nations. These laws and regulations were developed as an action to the exponential growth in the number of cases and deaths in many regions. In any case, the way toward checking huge gatherings of individuals is getting more troublesome. The checking interaction includes the recognition of any individual who isn't wearing a face

cover.
In this situation, I propose to support this situation by designing an image- based analysis method and an associated digital tool that are dedicated to the verification of the correct mask wearing by exploiting a smartphone and its front facing camera. The model is an integration between the deep learning and classical machine learning techniques with opencv, tensor flow and keras. I then used deep learning for feature extractions and combined it with three classical machine learning algorithms.
The work mainly proposes a model designed for validating the correct wearing of protection masks. In particular, major advantages of the presented work are as follows:

1. a first m-health application is designed for reinforcing the sensitizing campaigns launched with posters towards informing general public about the correct way of mask wearing (tool for limiting the inter-human contamination)

2. an original mask- related approach is designed toward being accessible to all individuals having a smartphone (high accessibility),

3. an original mask- related approach is designed toward being accessible to all individuals having a smartphone (high accessibility),

4. the designed mask-related checking application takes place upstream of crowd monitoring applications (e.g., validating the correct wearing of masks for travelers at airport gates) to limit the spread of COVID-19 (i.e., a complementary attribute by an individual monitoring).

1.1 Problem Formulation

The year 2020 has thrown mankind a mind-boggling sequence of events, the most life-changing of which is the COVID-19 pandemic, which has shocked the world since the year began. COVID-19 has called for stringent steps to be followed in order to prevent the spread of disease, which has an effect on the health and lives of many people.People are doing whatever they can to ensure their own and society's protection, from the most basic grooming practises to medical treatments; face masks are one of the personal protective equipment. When people leave their homes, they wear face masks, and officials make it a point to ensure that people wear face masks in groups and public areas.

A policy should be built to ensure that citizens obey this basic safety concept. To verify this, a face mask detector device can be used. Face mask detection refers to determining whether or not someone is wearing a mask.To detect the presence of a mask on a face, the first move is to detect the face, which divides the technique into two parts: detecting faces and detecting masks on those faces. Face detection is one of the applications of object detection, and it can be used in a variety of settings including defence, biometrics, and law enforcement.Many detector systems have been developed and are in use around the world. Much of this research, however, requires improvement; a better, more accurate detector is needed because the world cannot afford any further increases in corona cases.

In this project, a face mask detector that can tell the difference between people wearing masks and people who arent will be created. The proposed system in this study uses facenet for face detection and a neural network to detect the existence of a face mask. The algorithm is used to process photographs, videos, and live video streams using the adam optimizer.

1.2 Problem Identification

COVID-19's spread is becoming increasingly concerning for everyone on the planet. This virus can spread from person to person through droplets and airborne transmission. To prevent the spread of COVID-19, WHO recommends that everyone wear a face mask, practise social distancing, avoid crowded areas, and keep their immune systems in good shape.As a result, everybody can wear a face mask while they are outside to protect each other. Many greedy individuals, on the other hand, would not properly wear the face mask for a variety of reasons. To overcome this problem, a reliable face mask detection system must be established. The object detection algorithm can be used to detect a mask on the forehead.

According to World Health Organization (WHO) data, the COVID-19 virus is spread primarily through respiratory droplets and physical contact. When an infected individual coughs or sneezes, respiratory droplets are formed.Anyone who comes into close contact (within 1 m) with someone who has respiratory symptoms (coughing, sneezing) is at risk of infecting respiratory droplets. Droplets can also land on surfaces where the

virus may remain active, making an infected person's immediate environment a source of transmission (contact transmission).Wearing a medical mask is one of the preventative measures that will help to reduce the spread of COVID-19 and other respiratory viral diseases.Medical masks are described in this study as surgical or procedure masks that are flat or pleated (some are shaped like cups) and strapped to the head. They're put through their paces to ensure that they have a good balance of filtration, breathability, and fluid penetration resistance.The study examines a series of video streams/images in order to recognise people who follow the government's medical mask law. This could aid the government in taking appropriate action against non-compliant citizens.

Face-mask identification is both a detection and a classification issue since it necessitates first locating people's faces in digital images and then deciding whether or not they are wearing a mask. Owing to the broad applicability of face-detection technology, the first part of this issue has received a lot of attention in the computer vision literature.The second element, on the other hand (predicting whether or not a face is masked), has only recently sparked interest in the wake of the COVID-19 pandemic. Despite the fact that this part has received a lot of attention in the last year, it usually only tries to detect if a mask is present in the picture.

There is no particular consideration paid to whether the masks are correctly put on the face and thus worn in compliance with medical experts' guidelines.This reduces the utility of current face-mask detection techniques and necessitates further research into computer vision models capable of not only detecting the existence of facial masks in photographs but also assessing if they are worn correctly.

1.3 Problem Statement & Objectives

1.3.1 Problem Statement

The aim of the project is to create a new facial recognition system that can recognise a person wearing a medical mask. The problem is split into four phases to better understand how the facial recognition system works. The purpose of the division is to make the problem more understandable and experimental by focusing on each stage individually and defining the characteristics and necessity of conducting experiments. As a

result, the thesis begins by introducing some common algorithms that have previously been used to implement facial recognition systems, as well as datasets that have been used to construct and test these algorithms. The next step is to look at the detection and alignment processes, and figure out how they work and how they can be used in the system.

The following are the work's key contributions:

1. In the context of the COVID-19 pandemic, we compile a dataset of facial images for the problem of predicting right face-mask placements and annotate it with regard to placement—correctness, gender, ethnicity, and pose. In the form of a dataset of annotations, we make the list of images and labels publicly accessible (annotation dataset hereafter) called Face-Mask Label Dataset (FMLD).

2. We examine the output of different face-detection models with masked and mask-free face images, as well as the effect of face masks on existing face detectors. While the effect of face masks on the problem of face recognition has recently been studied, this problem has not been thoroughly investigated for face detectors.

3. We introduce and train a highly accurate and efficient pipeline for detecting properly worn face-masks using off-the-shelf deep-learning models, demonstrating that current solutions advocated for this task in the literature are in reality designed for a different problem and have very limited application for the task studied in this work.

1.3.2 Objectives

The main objective of the "Face mask recognition" project is to develop some useful technologies for preventing Coronavirus spread. The following are the primary goals for the implementation of this system:

1. Promoting the use of face masks with the aid of effective technologies to detect the face mask would help to prevent the spread of Coronavirus.

2. By forecasting potential COVID -19 outbreaks, you will assist in taking the appropriate precautions for society's safety.

3. Maintain a secure working climate

4. People's lives will be saved.

The following are some of the project's practical goals:

1. Using the PyTorch library, identify the person wearing a face mask on an image/video stream using computer vision and deep learning algorithms.

2. To determine whether or not the person is wearing a mask

3. To show how accurate the mask placement is.

1.4 Limitations

A few limitations in our current model have been observed

1. As shown in Figure 1.1, the model is unable to correctly distinguish face images in which the face is partially obscured by an individual in the foreground. In addition, if the camera height is greater than 10 feet, the model is unable to detect faces.

FIGURE 1.1: Shows faces which get incorrectly identified as the face gets occluded by the person in the front.

2. The developed system faces difficulties in classifying faces covered by hands since it almost looks like the person wearing a mask. When a person without a face mask travels in any car, the device is unable to correctly locate the person. It is extremely difficult to differentiate each person's face in a densely populated

environment.

Identifying individuals without face masks in this situation will be extremely difficult for our proposed method. To get the best results from this scheme, the city needs a large number of CCTV cameras to track the entire city, as well as dedicated manpower to prosecute violators.

3. When it comes to physical distancing, there is often a constraint on measuring the distance between two individuals.

Chapter 2

RESEARCH METHODOLOGY

2.1 CASE STUDY : COVID-19 Pandemic

COVID-19, a pandemic caused by a novel coronavirus, has been spreading continuously across the world for a long time. It has had a rapid impact on our daily lives, affecting global commerce and movement. COVID-19 has called for stringent steps to be followed in order to prevent the spread of disease, which has an effect on the health and lives of many people. People are doing whatever they can to ensure their own and society's protection, from the most basic hygiene standards to medical treatments.

The healthcare system all over the world is in a state of emergency. Many precautionary measures were taken to reduce the spread of this disease. Face masks are one of the personal protective equipment used to curb the spread of the virus. When people leave their houses, they put on face masks, and officials make it a point to ensure strictly that they do so in groups and public areas.

2.2 CASE STUDY : Face detection

The objective of face detection is to see if the picture or video contains any faces. If there are several faces, each one is surrounded by a bounding box, so we know where they are.Human faces are difficult to model since several variables can alter, such as facial expression, orientation, lighting conditions, and partial occlusions like sunglasses,

scarfs, and masks etc.. The face position parameters are obtained as a result of the identification and can take different forms, such as a rectangle covering the central part of the face, eye centers, or landmarks such as eyes, nose and mouth corners, eyebrows, nostrils, and so on. Face detection can be performed using two main approaches, Feature Base Approach and Image Base Approach.

Feature Base Approach - Objects are generally recognized by their distinguishing characteristics. A human face has a number of distinguishing features that enable it to be identified from a variety of other objects. It detects faces by extracting structural features such as eyes, nose, and mouth and using them to locate them. Feature Base Approach is implemented using OpenCV.

Image Base Approach - Image-based approaches, in general, rely on statistical analysis and machine learning techniques to discover the related characteristics of face and non-face images. The learned characteristics are represented as distribution models or discriminant functions, which are then used to detect faces. We use a variety of algorithms in this process, including neural networks, HMM, SVM, and AdaBoost learning.

Chapter 3

LITERATURE SURVEY AND REVIEW

Face Mask detection has turned up to be an astonishing problem within the domain of image processing and computer vision. Face detection has various use cases ranging from face recognition to capturing facial motions, where the latter calls for the face to be revealed with very high precision.Due to the rapid advancement in the domain of machine learning algorithms, the jeopardies of face mask detection technology seem to be well addressed yet. This technology is more relevant today because it's wont to detect faces not only in static images and videos but also in real-time inspection and supervision.

With the advancements of deep learning and CNN ,very high accuracy in image classification and object detection can be achieved.The detection of face masks is a difficult task for the current proposed face detector models. This is due to the fact that mask-wearing faces have a variety of accommodations, degrees of obstruction, and mask sizes. They are used to make self-focusing, human-computer interaction, and image database management easier.

This chapter is mainly divided into two sub-chapters. Namely:

- Literature collection and segregation (called as literature survey – collection of data)
- Critical review of selected literature (from the ones collected during the survey)

3.1 Existing & Proposed Model

3.1.1 Existing Models

Various researchers and analysts have recently concentrated on gray-scale face images. Others used AdaBoost, which was an excellent classifier for training purposes. Some were entirely based on pattern recognition models, having initial knowledge of the face model, while others used AdaBoost, which was an excellent classifier for training purposes. Then there was the Viola-Jones Detector, which was a technological advance in face detection, and Face recognition in real time is now possible. It had a number of issues, including the orientation and brightness of the face, which made it difficult to intercept. So, in a nutshell, it didn't fit in dark or dim light. As a result, researchers began looking for a new alternative model capable of detecting faces as well as masks on the face. Many datasets for face detection have been created in the past to form an impression of face mask detection models.

Prior datasets comprised of pictures got in regulated environmental factors, while late datasets are developed by taking on the web pictures like WiderFace , IJB-A , MALF, and CelebA.Comments are accommodated present countenances in these datasets when contrasted with before ones. Enormous datasets are significantly more required for improving preparing and testing information and perform true applications in a lot easier way.

This necessitates a variety of deep learning algorithms that can read faces and masks directly from the user's data. Face Mask detection templates come in a variety of shapes and sizes. These can be classified into many groups. The Viola-Jones face detector, which was mentioned earlier in this section, was used to accept boosted cascades with easy haar features in boosting-based classification. The Viola-Jones detector model was then used to build a Multiview face mask detector. In addition, decision trees algorithms were used to build a face mask detector model. This category's face mask detectors were really good at detecting masks.

3.1.2 Proposed Model

The proposed model in this paper focuses on differentiating the faces with masks and without masks in Livestream and also tells the percentage accuracy of the position of the mask on the face, with help of a deep learning algorithm by using the OpenCV, Tensor flow, Keras, and PyTorch library.

1. This model has been trained using a dataset created by obtaining images from various sources such as Kaggle, few open source image libraries.
2. The training image generator for data augmentation is constructed, by which multiple images can be created using a single image by adding properties such as flip,rotate.
3. In our model, once the input image is processed into an array, it is being sent into MobileNet instead of convolution layer,followed by pooling.
4. MobileNet is being used in this model as it is very fast in processing compared to Cnn and it uses lesser parameters.
5. The pooling is done with a poolsize(7,7) and then is flattened.A dense layer with 12.8 neurons is added.Dropout is used to avoid non linear use cases.
6. The optimizer being used in the model is Adam optimizer which is a goto for image datasets.
7. To apply the model in the camera i.e livestream, firstly the face should be detected using readnet function of dnn module in open CV library and the previously trained model must be loaded.

3.2 Tools and Technologies used

3.2.1 Spyder

Spyder is a cross-platform, open-source integrated development environment (IDE) for scientific Python programming. Spyder works with a variety of common Python packages, including NumPy, SciPy, Matplotlib, pandas, IPython, SymPy, and Cython, as well as other open-source applications. It's made available under the MIT licence.
Spyder can be expanded much further by third-party plugins, in addition to its many built-in features. Spyder can also be used as a PyQt5 extension library, allowing you to extend its features and use its components in your own applications, such as the interactive console and advanced editor.

3.2.2 Machine Learning

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves.
Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop conventional algorithms to perform the needed tasks. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers to learn automatically without human intervention or assistance and adjust actions accordingly. But, using the classic algorithms of machine learning, text is considered as a sequence of keywords; instead, an approach based on semantic analysis mimics the human ability to understand the meaning of a text.
Machine Learning approaches are customarily partitioned into three general classifications, contingent upon the idea of the "signal " or "feedback" accessible to the learning framework:

1. Supervised learning: The PC is introduced with model sources of info and their ideal yields, given by an ”educator”, and the objective is to become familiar with an overall guideline that maps contributions to yields.

2. Unsupervised learning: No marks are given to the learning calculation, leaving it all alone to discover structure in its information. Unsupervised learning can be an objective in itself (finding covered up designs in information) or an implies towards an end (feature learning).

3. Reinforcement learning: A PC program connects with a powerful climate where it must play out a specific objective (like driving a vehicle or playing a game against a rival). As it explores its issue space, the program is given criticism that is practically equivalent to rewards, which it attempts to augment.

3.2.3 Keras

Keras is an API intended for individuals, not machines. Keras follows best practices for diminishing intellectual burden: it offers reliable and basic APIs, it limits the quantity of client activities needed for regular use cases, and it gives clear and noteworthy mistake messages. It additionally has broad documentation and engineer guides. Keras contains various executions of usually utilized neural organization building squares like layers, targets, initiation capacities, streamlining agents, and a large group of instruments to make working with picture and text information simpler to improve on the coding fundamental for composing profound neural organization code.

The code is facilitated on GitHub, and local area support discussions incorporate the GitHub issues page, and a Slack channel. Keras is a moderate Python library for profound discovery that can run on top of Theano or TensorFlow. It was created to make carrying out profound learning models as quick and simple as workable for exploration and development.It runs onPython 2.7 or 3.5 and can consistently execute on GPUs and CPUs given the hidden structures. It is delivered under the lenient MIT permit. Keras was created and kept up by François Chollet, a Google engineer utilizing four core values:

1. Modularity: A model can be understood as a sequence or a graph alone. All the concerns of a deep learning model are discrete components that can be combined in arbitrary ways.

2. Minimalism: The library provides just enough to achieve an outcome, no frills and maximizing readability.

3. Extensibility: New components are intentionally easy to add and use within the framework, intended for researchers to trial and explore new ideas.

4. Python: No separate model files with custom file formats. Everything is native Python.

Keras is intended for moderation and seclusion permitting you to rapidly characterize profound learning models and run them on top of a Theano or TensorFlow backend.

3.2.4 Open CV

(Open Source Computer Vision Library) is an open source PC vision and AI programming library. OpenCV worked to give a typical foundation for PC vision applications and to speed up the utilization of machine insight in the business items. Being a BSD-authorized item, OpenCV makes it simple for organizations to use and adjust the code. The library has more than 2500 upgraded algorithms, which incorporates an exhaustive arrangement of both work of art and state-of-the-craftsmanship PC vision and AI calculations. These calculations can be utilized to distinguish and perceive faces, distinguish objects, order human activities in recordings, track camera developments, track moving articles, extricate 3D models of articles, produce 3D point mists from sound system cameras, join pictures together to create a high goal picture of a whole scene, discover comparative pictures from a picture data set, eliminate red eyes from pictures taken utilizing streak, follow eye developments, perceive landscape and build up markers to overlay it with expanded reality, and so on OpenCV has more than 47 thousand individuals of client local area and assessed number of downloads surpassing 18 million.

The library is utilized widely in organizations, research gatherings and by legislative bodies. Along with well-set up organizations like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that utilize the library, there are numerous new businesses, for example, Applied Minds, Video Surf, and Zeitera, that make broad utilization of OpenCV. OpenCV's sent uses length the reach from sewing road see pictures together, identifying interruptions in reconnaissance video in Israel, observing mine gear in China,

assisting robots with exploring and get objects at WillowCarport, location of pool suffocating mishaps in Europe, running intuitive workmanship in Spain and New York, checking runways for trash in Turkey, investigating names on items in plants all throughout the planet on to fast face location in Japan.

It has C++, Python, Java and MATLAB interfaces and upholds Windows, Linux, Android and Mac OS. OpenCV inclines generally towards constant vision applications and takes advantage of MMX and SSE guidelines when accessible. A full-included CUDA and OpenCL interfaces are in effect effectively grown at the present time. There are more than 500 calculations and around 10 fold the number of capacities that make or back those calculations. OpenCV is composed locally in C++ and has a format interface that works consistently with STL holders.

3.2.5 TensorFlow

TensorFlow is a free and open-source programming library for data flow and differentiable programming across a scope of errands. It is an emblematic numerical library, and is additionally utilized for AI applications like neural networks. It is utilized for both exploration and creation at Google, TensorFlow is Google Brain's second-age framework. Form 1.0.0 was delivered on February 11, While the reference execution runs on single gadgets, TensorFlow can run on numerous CPUs and GPUs (with discretionary CUDA and SYCL augmentations for universally useful figuring on graphical processing units).
Tensor Flow is accessible on 64-bit Linux, macOS, Windows, what's more, portable figuring stages including Android and iOS. Its adaptable design takes into consideration the simple arrangement of calculation across an assortment of stages (CPUs, GPUs, TPUs), and from work areas to clusters of servers to mobiles and edge gadgets. The name TensorFlow gets from the tasks that such neural organizations perform on multi-dimensional information clusters, which are alluded to as tensors. During the Google I/O Meeting in June 2016, Jeff Dean expressed that 1,500 archives on GitHub referenced TensorFlow, of which just 5 were from Google. Unlike other mathematical libraries proposed for use in Deep Learning like Theano, TensorFlow was intended for utilize both in innovative work and underway frameworks, not least RankBrain in Google search also, the fun DeepDream project.

3.2.6 Deep Learning

Machine Learning, on the other hand, is a subset of Artificial Intelligence, and Deep Learning is a subset of Machine Learning. Deep Learning, on the other hand, is a form of Machine Learning that is influenced by the human brain's structure. Deep learning algorithms analyse data with a predetermined logical framework in order to draw similar conclusions as humans. Deep learning is used to do this. Deep learning does this by using a multi-layered system of algorithms known as neural networks.
The layers will explicitly and independently learn an implicit representation of the raw data. Over several layers of artificial neural-nets, a more abstract and compressed representation of the raw data is made. The outcome may be the grouping of the input data into various groups, for example.

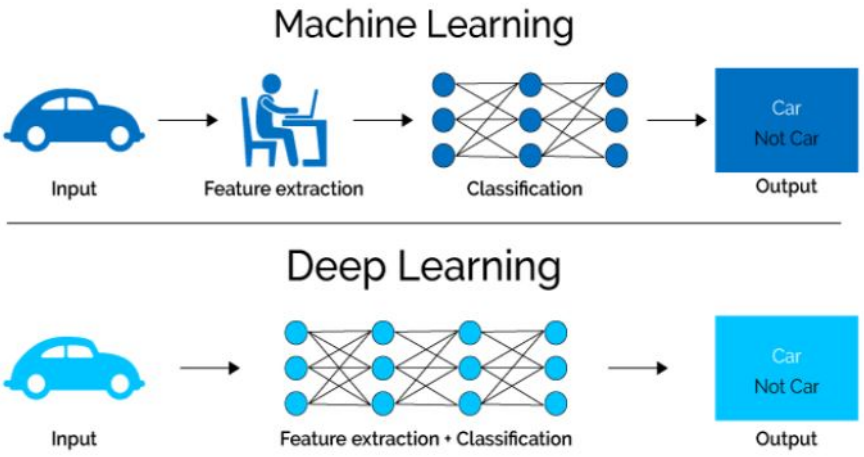

FIGURE 3.1: Machine Learning and Deep Learning

The neural network optimises this step during the training phase to achieve the best possible abstract representation of the input data. This compressed representation of the input data is then used to produce the result. The result can be, for example, the classification of the input data into different classes. During the training process, this step is also optimized by the neural network to obtain the best possible abstract representation of the input data. This means that performing and optimising the feature extraction process with deep learning models requires little or no manual effort.

3.3 Hardware and Software Requirements

3.3.1 Hardware Specification

TABLE 3.1: Hardware Specifications

Processor	4.0GHz
RAM	2 GB
Hard Disk	30 GB
Keyboard	104 Keys
Display	VGA

3.3.2 Software Specification

TABLE 3.2: Software Specifications

Operating Systems	Windows/Linux
Language	Python
Editor	Spyder
Interpreter	Python Interpreter

3.4 Critical Review

Taking into account the COVID 19 Pandemic, Face mask detection has become the need of the hour. Face mask is considered as one of the essentials to prevent the spread of the virus. The detection of face mask is a difficult task for the current proposed face detector models. This is due to the fact that mask-wearing faces have a variety of accommodations, degrees of obstruction, and mask sizes. However the advancements in the fields of deep learning and CNN helps to attain high accuracy image classification and object identification. Face mask detection can be made possible with the help of basic Machine Learning (ML) packages such as TensorFlow, Keras, OpenCV and Scikit-Learn, and MobileNetV2 architecture. The main objectives of all of the above mentioned papers are to show how the existing face mask detection models work. Most of the papers describe how to make face mask detection system to detect whether the person is wearing the mask or not.

All the technologies and frameworks mentioned in the above section are to show how

the respective technologies are selected. OpenCV - dnn is used for face detection in video streams. Applying the face detection model will help to get the number of faces detected, the location of their bounding boxes, and the confidence score in those predictions.

Keras and Tensorflow makes implementation of machine learning algorithm more easier. MobileNetV2 architecture was selected as it is computationally efficient and lighter when compared to other convolutional neural networks available. The lightweight of MobileNetV2 makes the integration of the model to various embedded devices much more easier in future. The literature review explains the downsides of various frameworks used for the project. It also mentions the specifications regarding hardware and software needed to host this particular project as a whole.

Chapter 4

ACTUAL WORK

Due to the worldwide COVID-19 corona virus outbreak, the wearing of face masks in public is becoming more common. People used to wear masks to protect their health from air pollution before Covid-19. Some people conceal their feelings from the public by covering their faces, while others are self-conscious about their appearance. COVID-19 transmission is slowed by wearing face masks, according to scientists.

So, here I present a computer vision and deep learning-based mask face detection model. The proposed model can be used in conjunction with security cameras to prevent COVID-19 transmission by detecting people who aren't wearing face masks. The model combines deep learning and classical machine learning techniques using opencv, tensor flow, and other tools.I have used deep transfer leering for feature extractions and combined it with classical machine learning algorithms.
This chapter is divided into five parts which explains the main working of the project. They are as follows:

1. Methodology for the Study
2. Experimental and Analytical Work Completed in the Project
3. Modeling, Analysis and Design

4.1 Methodology for the Study

Developing an optimal masked Face recognition algorithm is a significant challenge in computer vision with limited sample cases and quite a few reference data sets. The experiment for masked faces recognition system was implemented by connecting the three steps in chapter three (Detection, Alignment, and Recognition), by utilising multiple solutions for each stage were achieved, the author was able to evaluate the efficiency and the rapidity to accomplish the experiment.

This research includes collection of a face mask data set using Kaggle,google images etc available on the internet and increases the number of images using image data generator function of mobile net.Then the mask detector model is created using Mobile net ,DNN and Adam optimizer using the collected data set.For face detection we use the existing model of machine learning called face net. The model is then applied on live stream video to detect if a person is wearing a mask or not and also tell the percentage accuracy of the mask.

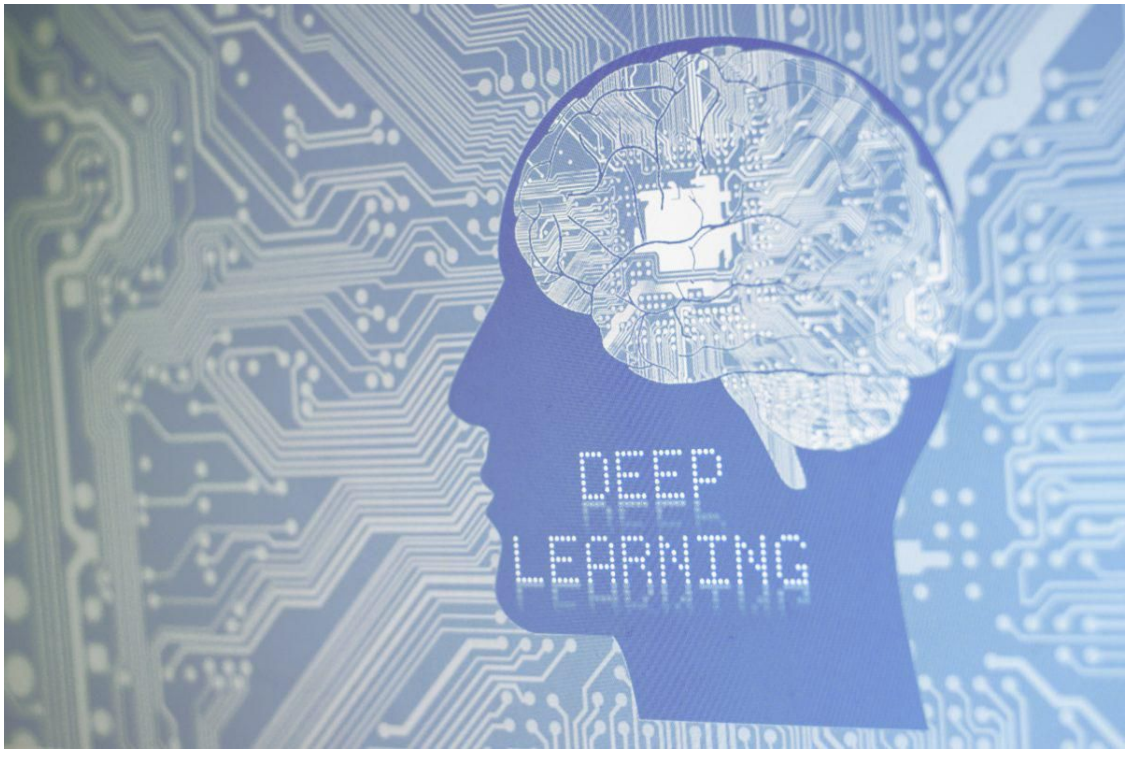

FIGURE 4.1: Deep Learning

4.2 Modelling, Analysis and Design

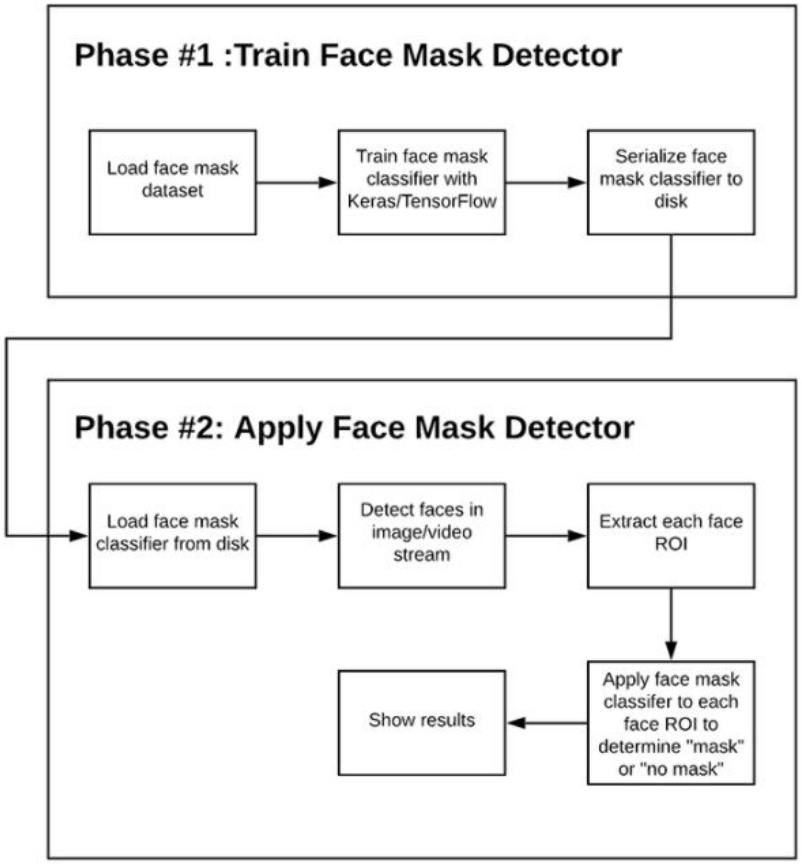

FIGURE 4.2: Block Diagram

The project is developed in 2 phases:

1. Phase 1: training the Face mask detector
2. Phase 2: Applying the face mask detector

Under Phase 1:

Under this phase various steps like creation of data set , pre-processing of data, training the model are included.

4.2.1 DATASET

This model has been trained using a data set containing 2 categories of pictures namely, with mask and without mask. This data set is created by obtaining images from various sources such as Kaggle,google images and few open source image libraries.

TABLE 4.1: Dataset

Total images	3833 images
With mask	1915 images
Without mask	1918 images

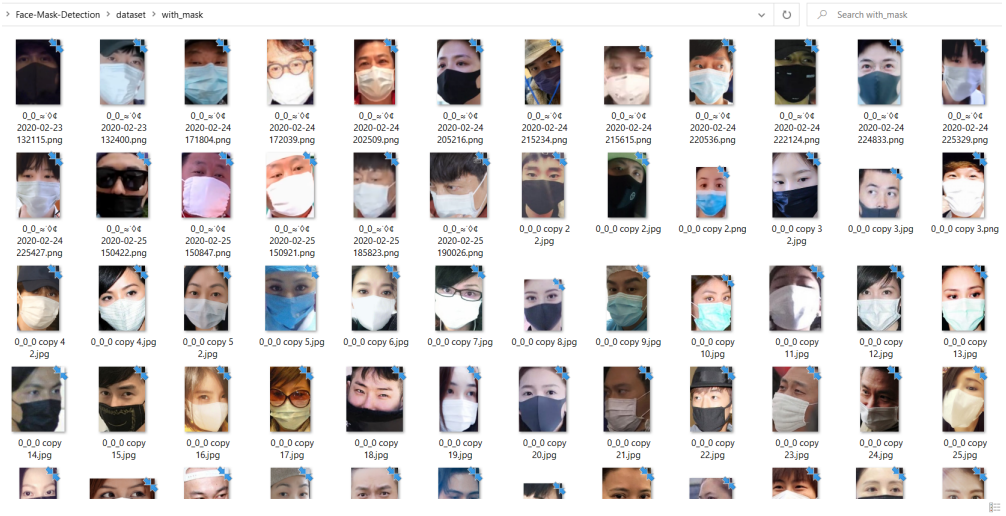

FIGURE 4.3: Dataset with mask

The dataset shown in figure 4.3 with mask contains various images of people wearing mask.

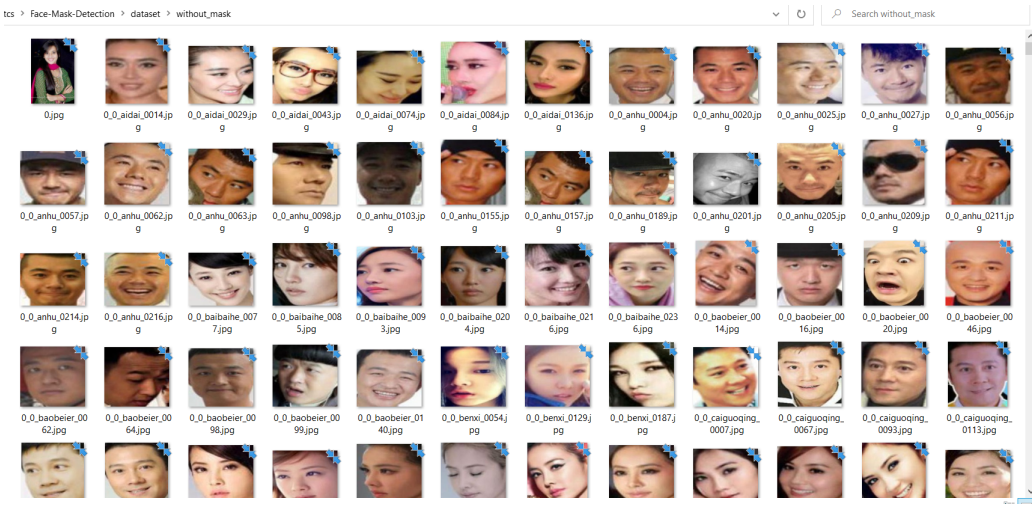

FIGURE 4.4: Dataset without mask

The dataset shown in figure 4.4 without mask contains various images of people not wearing mask.

4.2.2 DATA PRE-PROCESSING

The data pre-processing is one of the major steps of this project.The steps as mentioned below was applied to all the raw input images to convert them into clean versions, which could be fed to a neural network machine learning model.

1. converting all the images in the data set to (224 x 224) pixels using the load image function of keras.
2. the images are then converted into Numpy arrays using image to array function of keras.
3. The images are appended to label list and then convert them into arrays as deep learning model work with arrays.
4. The data set is then split into training and testing set.Here it is in the ratio 80:20

```
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)

# perform one-hot encoding on the labels
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)
```

FIGURE 4.5: The data Pre-processing

4.2.3 TRAINING

1. The training image generator for data augmentation is constructed, by which multiple images can be created using a single image by adding properties such as flip,rotate thus increasing the dataset.

2. To train the model CNN with a little modification has been used. In our model, once the input image is processed into an array, it is being sent into MobileNet instead of convolution layer,followed by pooling.

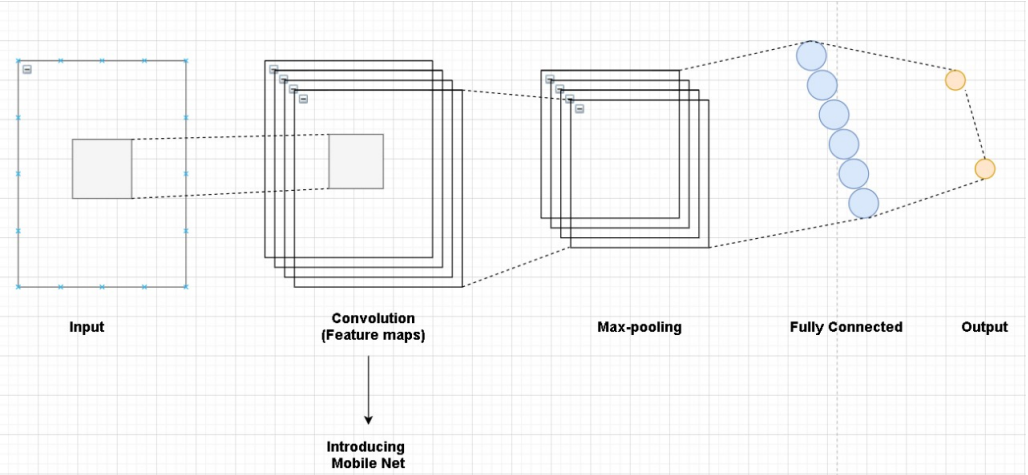

FIGURE 4.6: Training model using Mobilenet

3. It is then flattened and the output is given as masked or unmasked.MobileNet is being used in this model as it is very fast in processing compared to Cnn and it uses lesser parameters.

4. The initial weights of the imagenet model are going to be changed for yielding better results and accuracy.The pooling is done with a poolsize(7,7) and then is flattened.A dense layer with 12.8 neurons is added.Dropout is used to avoid non linear use cases.

5. The optimizer being used in the model is Adam optimizer which is a goto for image datasets.

6. After training of the model,a classification report is generated and saved using the predict method and the graph is plotted between training loss and accuracy using matplotlib.

7. The trained model is saved and applied on the face detection model in next phase.

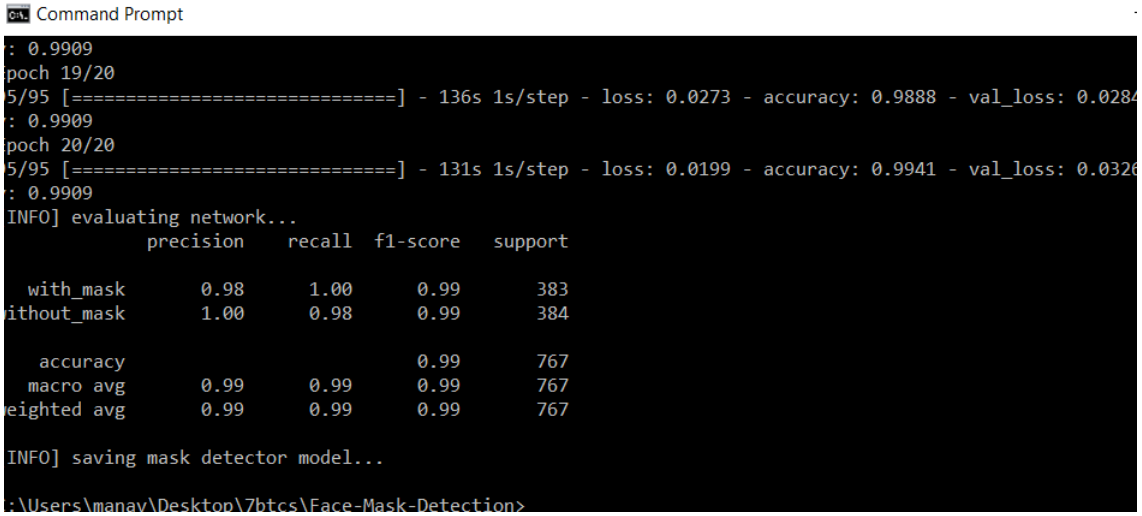

FIGURE 4.7: Model after Training

Under Phase 2:

Under this phase the mask detector model that is generated in phase1 is applied on the facenet model to detect mask during livestream.

4.2.4 APPLYING THE MODEL IN LIVESTREAM

1. To apply the model in the camera i.e livestream, firstly the face should be detected using readnet function of dnn module in open CV library and the previously trained model must be loaded.

2. Every frame in the video is resized to 400 pixels.The model detects the face and predicts if the person is wearing a mask or not and mentions it using the color green for mask and red for no mask with the help of color=(b,g,r) of opencv.

3. An additional function of percentage of prediction of the position of the mask is also added to the model.

4. Thus the model returns prediction and location.

4.3 Experimental and or Analytical Work

In this section we will discuss about the experimental work that I have conducted in our project.

```
# initialize the initial learning rate, number of epochs to train for,
# and batch size
INIT_LR = 1e-4
EPOCHS = 40
BS = 32
```

FIGURE 4.8: Experimental values

As we can see in the figure 4.9, The epoch value is taken as 40.
The learning rate is taken as 1e-4 as less learning rate gives less losses and increase the accuracy.

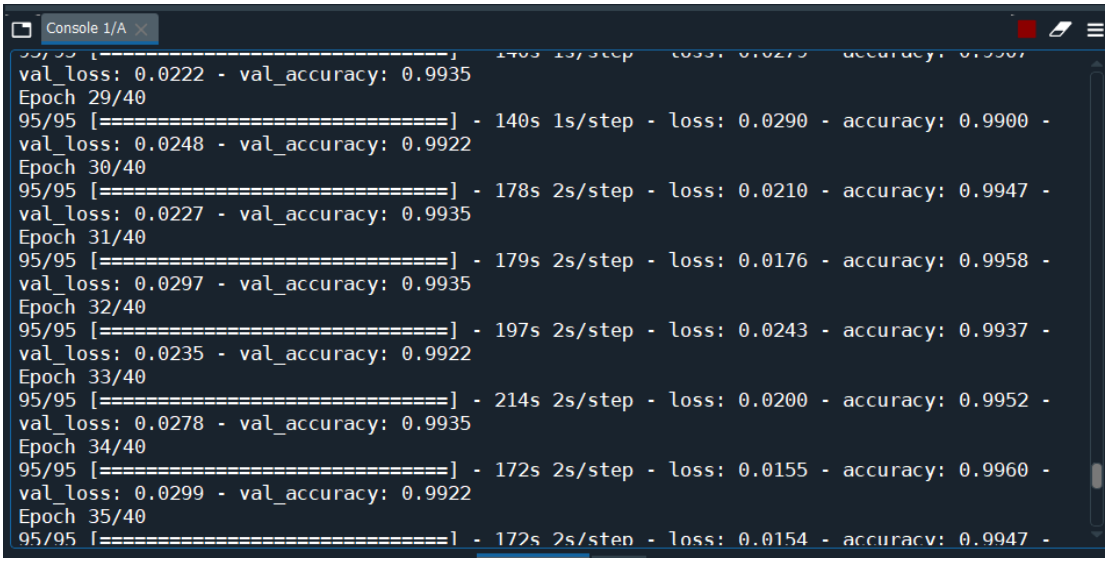

FIGURE 4.9: Experimental values

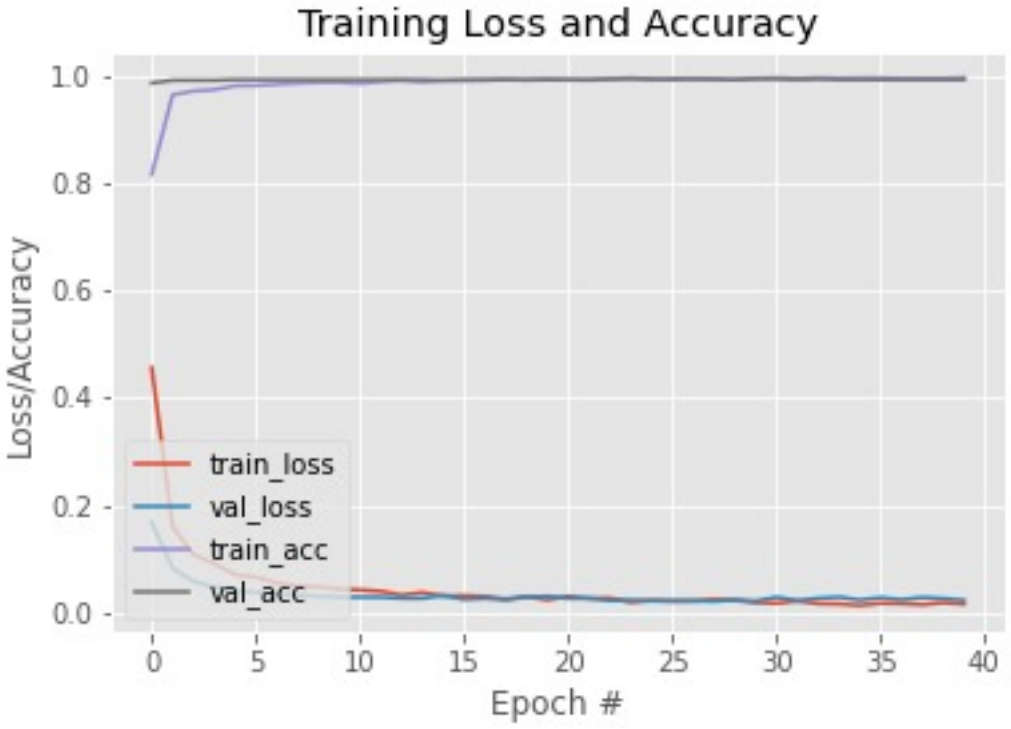

FIGURE 4.10: Graph plotted between Training loss and accuracy

The figure 4.10 shows the graph plotted between Training loss and accuracy on y axis and epoch values on x axis.It is plotted using MatplotLib and is automatically saved.

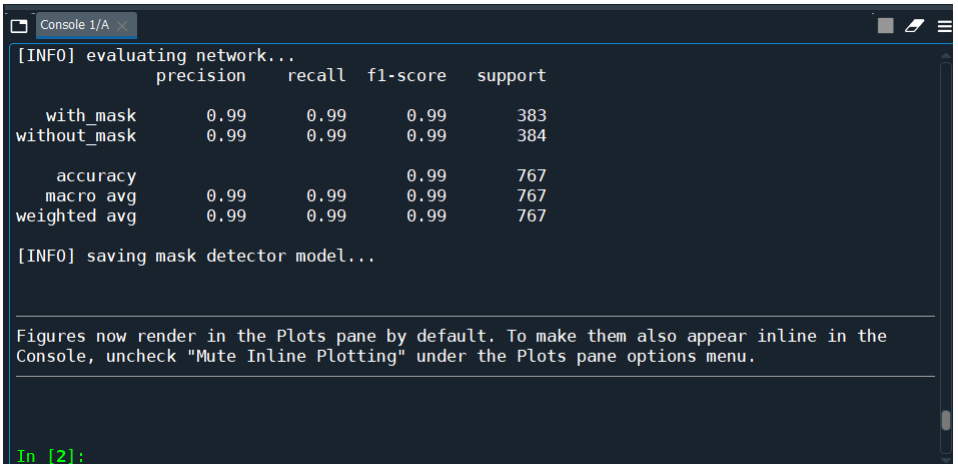

FIGURE 4.11: Experimental values taking Epoch as 40

The values of epoch and batch size can be subjected to change.

Testing with Graph with epoch value 20

FIGURE 4.12: Experimental values taking Epoch as 20

The figure 4.12 shows the graph plotted between Training loss and accuracy when epoch value is 20.

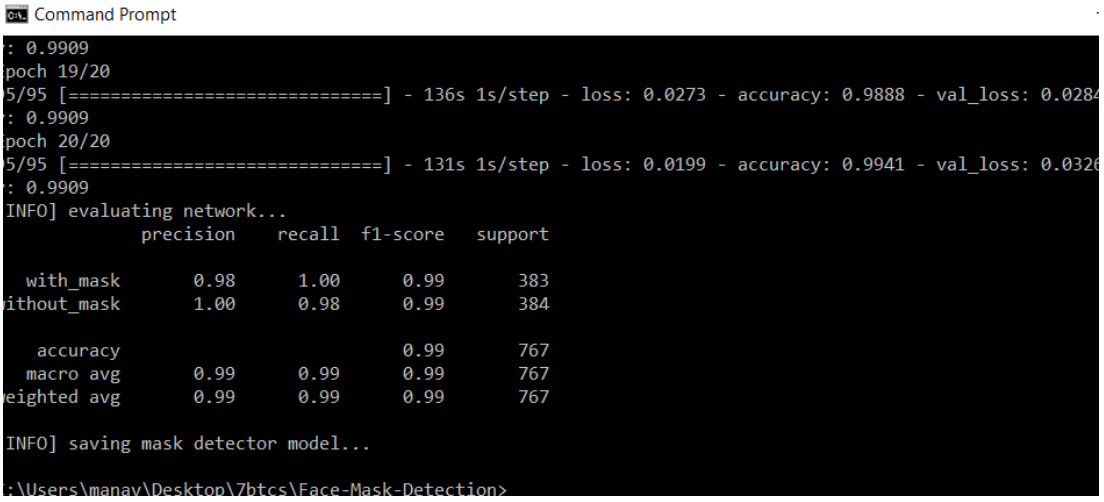

FIGURE 4.13: Experimental values taking Epoch as 20

The figure 4.13 shows the values of precision, recall and support when epoch value is 20.

Testing with Graph with epoch value 30

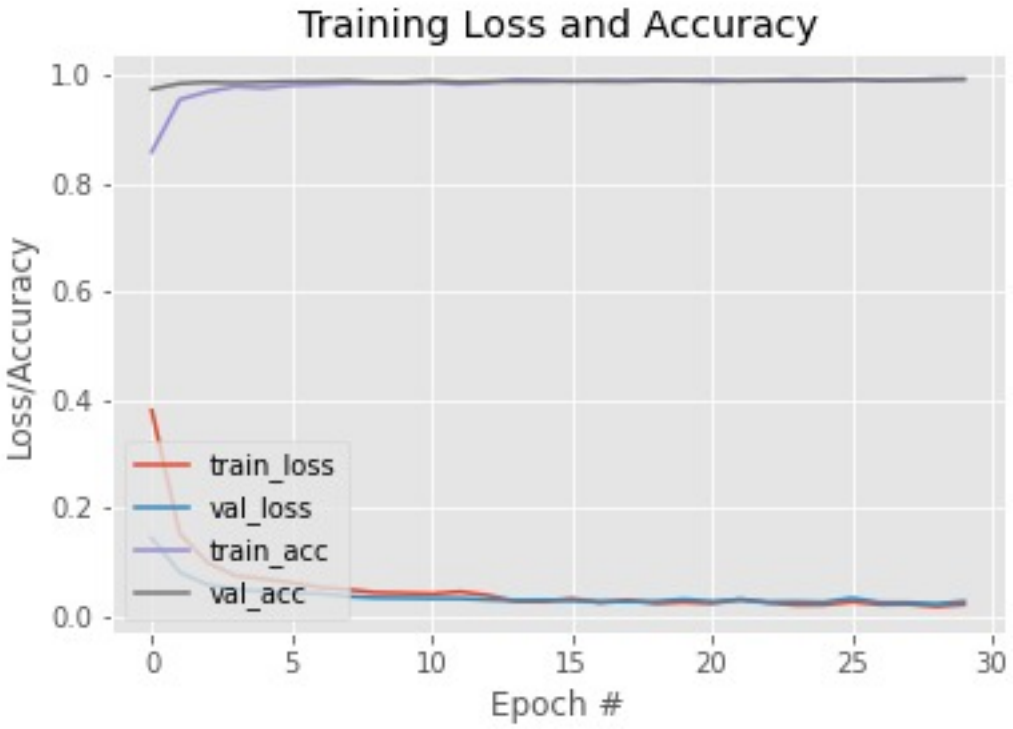

FIGURE 4.14: Experimental values taking Epoch as 30

The figure 4.14 shows the graph plotted between Training loss and accuracy when epoch value is 30.

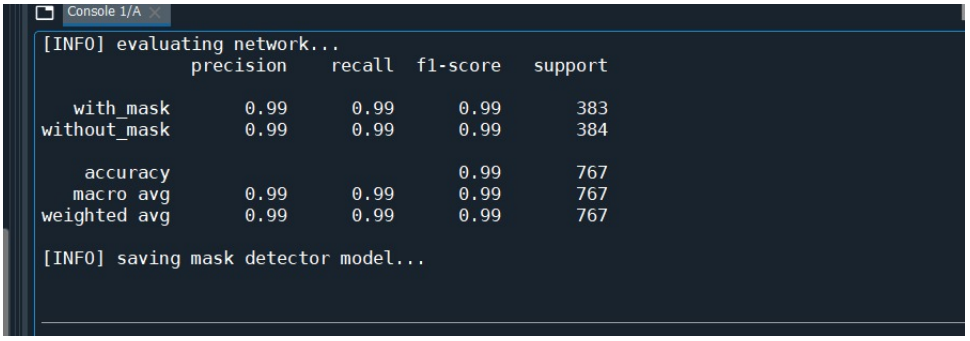

FIGURE 4.15: Experimental values taking Epoch as 30

The figure 4.15 shows the values of precision, recall and support when epoch value is 30.

Testing with epoch value 30 and batch size 50

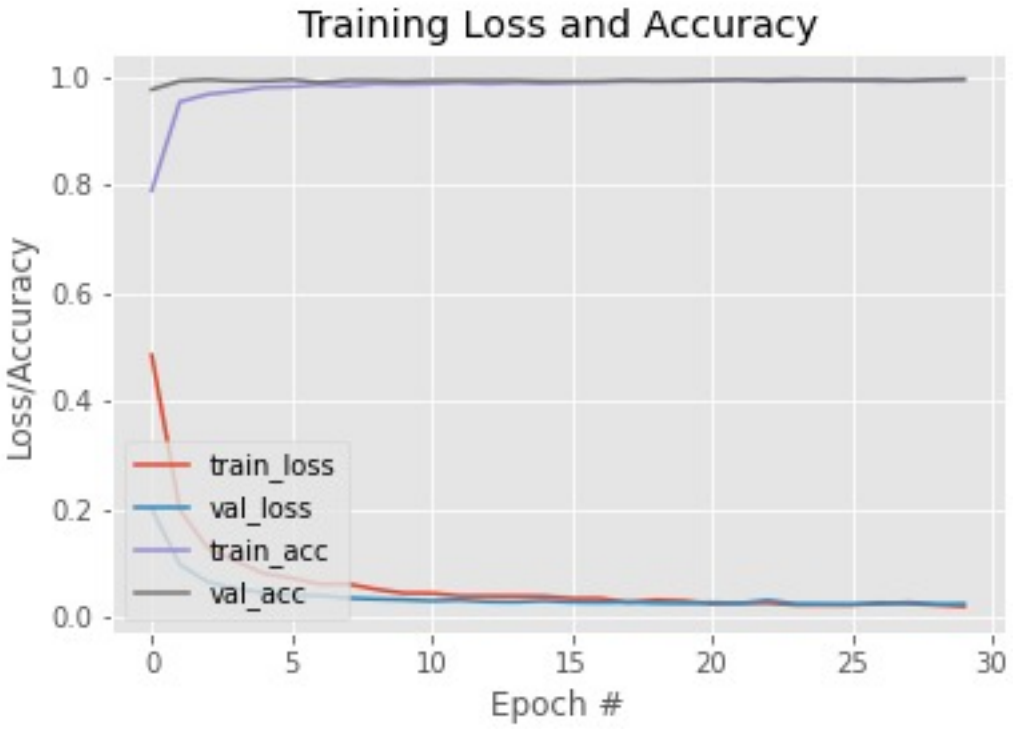

FIGURE 4.16: Experimental values taking Epoch as 30 and batchsize 50

The figure 4.16 shows the graph plotted between Training loss and accuracy when epoch value is 30 and batchsize 50.

```
              precision    recall  f1-score   support

   with_mask       0.99      0.99      0.99       383
without_mask       0.99      0.99      0.99       384

    accuracy                           0.99       767
   macro avg       0.99      0.99      0.99       767
weighted avg       0.99      0.99      0.99       767

[INFO] saving mask detector model...
```

FIGURE 4.17: Experimental values taking Epoch as 30 and batchsize 50

The figure 4.17 shows the values of precision, recall and support when epoch value is 30 and batchsize 50.

4.4 Prototype and testing

The Prototype of this model is generated using mobilenet.Two models are being generated here(Base and Head)

```
# 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)
```

FIGURE 4.18: Head and Base model

Base Model:

1. In the base model the imagenet function is used with modifying he default weights
2. input tensor function is used to input images and is set to (224,224,3) which corresponds to(height,width,colour).The colour is set to 3 as we follow 3 channel colour system(r,g,b)

```
baseModel = MobileNetV2(weights="imagenet", include_top=False,
    input_tensor=Input(shape=(224, 224, 3)))
```

FIGURE 4.19: ImageNet

Head Model:

1. The output of the base model is passed and a pool size of 7*7 is created
2. the layer is then flattened by adding a dense layer of 128 nuerons
3. ”relu” activation function is used on this layer

```
# place the head FC model on top of the base model (this will become
# the actual model we will train)
model = Model(inputs=baseModel.input, outputs=headModel)
```

FIGURE 4.20: Output Masknet model

The output model is now of 2 layer.And ”softmax” activation function is used on the top of all the layers.

Adam optimizer is used here to track the accuracy metrices.

4.4.1 Dependencies and Requirements

Adam Optimizer

The optimization algorithm you choose for your deep learning model will make all the difference in terms of getting good results in minutes, hours, and days. The Adam optimization algorithm is a stochastic gradient descent extension that has recently gained traction in computer vision and natural language processing applications. Adam is an optimization algorithm that can be used to update network weights iteratively based on training data instead of the traditional stochastic gradient descent technique.

Numpy

NumPy is a Python scripting language. ’Numerical Python’ is what it stands for. It is a library that contains multidimensional array objects as well as a set of array processing routines.Jim Hugunin created Numeric, the forerunner of NumPy. Numarray, a new kit with some additional features, was also developed.Travis Oliphant developed the NumPy package in 2005 by combining the features of Numarray with the Numeric package. This open source project has a large number of contributors.

NumPy is a Python-based alternative to MatLab.

NumPy is often used in conjunction with SciPy (Scientific Python) and Matplotlib (plotting library). This combination is frequently used as a substitute for MatLab, a popular technical computing framework. The Python alternative to MatLab, on the other hand, is now regarded as a more modern and comprehensive programming language.It is open-source, which is an added advantage.

Matplotlib

Matplotlib is a Python plotting library. It is used in conjunction with NumPy to build an ecosystem that is a viable open source replacement for MatLab. It's also compatible with graphics frameworks like PyQt and wxPython.
John D. Hunter was the first to create the Matplotlib module. Michael Droettboom has been the lead developer since 2012.Matplotlib 1.5.1 is the most recent stable version available. On www.matplotlib.org, the package is available as a binary distribution as well as source code.
The package is normally imported into a Python script by inserting the following statement:

From matplotlib import pyplot as plt

The most important function in the matplotlib library is pyplot(), which is used to plot 2D data.

Batch size

In modern deep learning systems, batch size is a significant hyperparameter to tune. Practitioners often choose to train their models with a larger batch size because it helps them to benefit from the parallelism of GPUs. However, it is well understood that using a large batch size results in poor generalisation.Smaller batch sizes, on the other hand, have been shown to have faster convergence to good solutions because they allow the model to begin learning before it has seen all of the data. However, there is a risk that the model will not converge to the global optima.
It's widely assumed that there's a sweet spot for batch size somewhere between one

and the entire training dataset.Smaller batch sizes, on the other hand, have been shown to have faster convergence to good solutions because they allow the model to begin learning before it has seen all of the data. However, there is a risk that the model will not converge to the global optima.

It's widely assumed that there's a sweet spot for batch size somewhere between one and the entire training dataset.

Chapter 5

RESULTS, DISCUSSIONS AND CONCLUSIONS

5.1 Results & Analysis

Based on the implementation, the output of the module is expected to distinguish people wearing masks properly and those who are not wearing masks. After training and testing the model is implemented to check whether the people are wearing face masks accurately in the real time video. By making slight changes in the weights while training the efficacy and relevance of the result have improved. This model has been further tested using live video streams and it has shown highly accurate results. The model is built using computationally efficient and light weight MobileNetV2 architecture, which makes it easier to deploy the model in embedded devices like android phones, Raspberry Pi, Google Coral etc.The dataset is partitioned into training and testing sets by retaining a rational proportion of different classes. The dataset contains 3833 samples in total, with 80 percent of them being used in the training process and 20percent in the testing phase. There are 3066 and 767 images in the training and research datasets, respectively. The built architecture is trained for 40 epochs. The file which was given as an input for validation was successfully analyzed and the results obtained were accurate.The output shows how accurately the people in the live video stream is wearing their face mask.

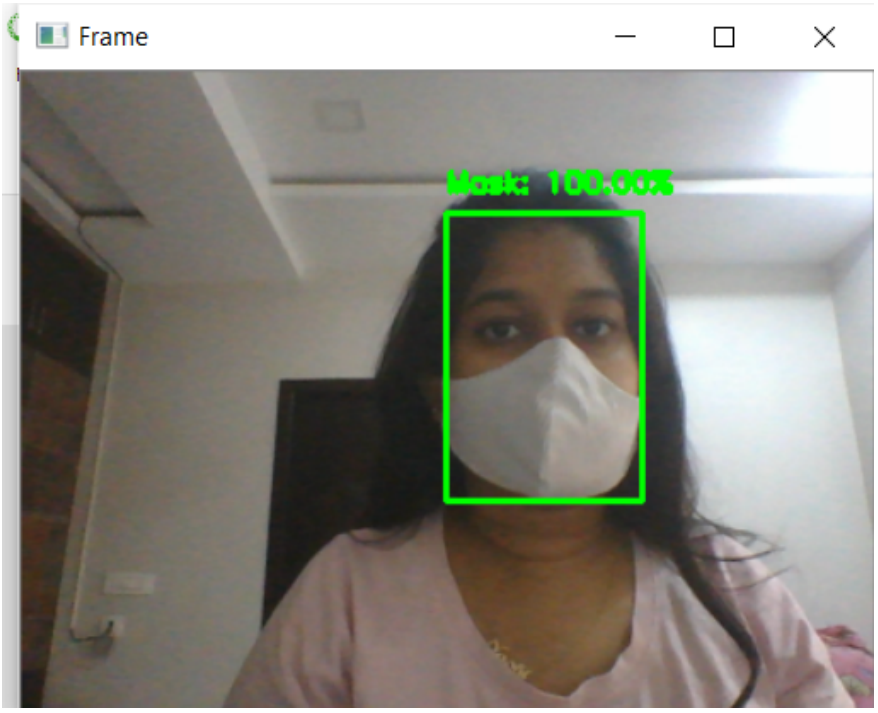

FIGURE 5.1: Output with mask

The figure above shows a person wearing mask properly.

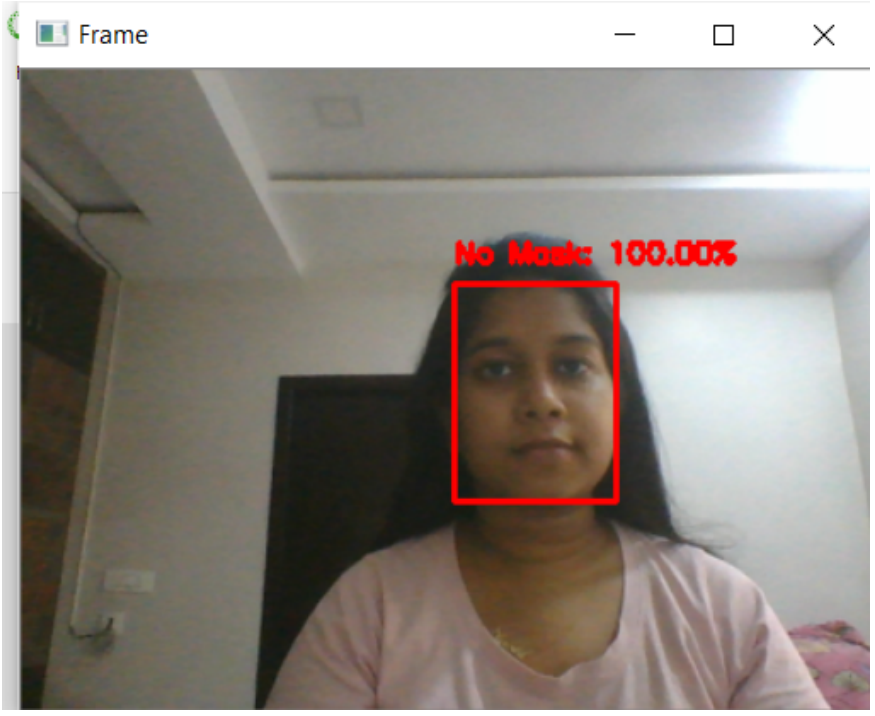

FIGURE 5.2: Output with mask

The figure above shows a person wearing no mask.

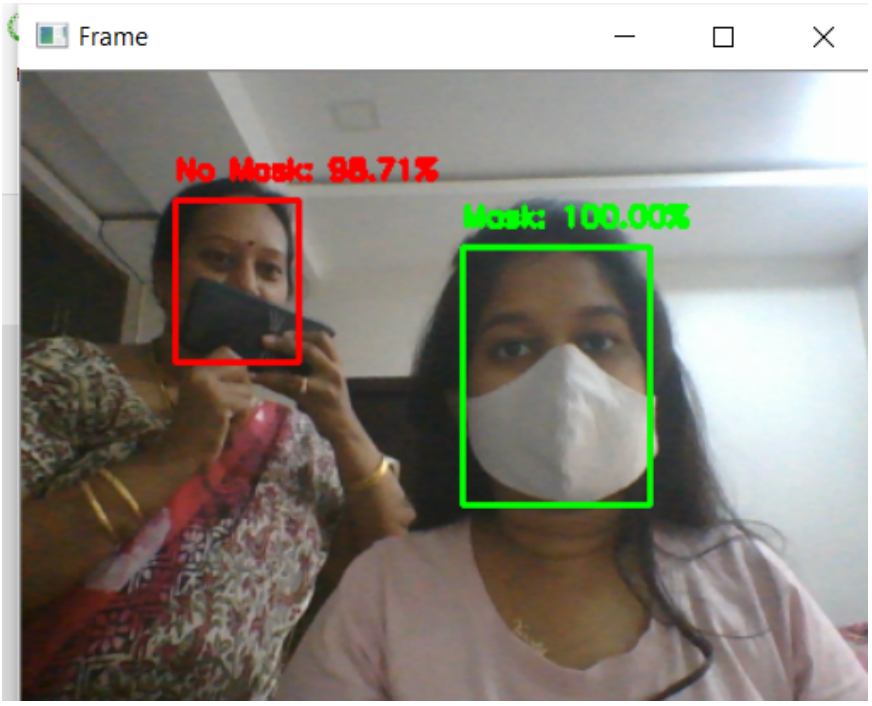

FIGURE 5.3: Obstacle differentiation

The figure above shows a person putting phone in place of mask.

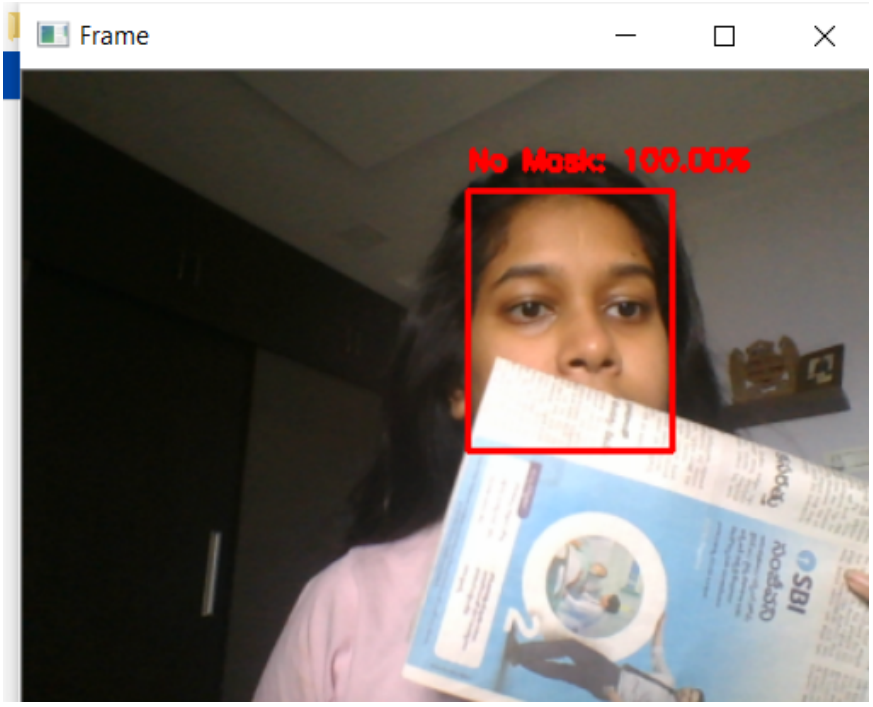

FIGURE 5.4: Obstacle differentiation

The figure above shows a person putting a paper in place of mask.

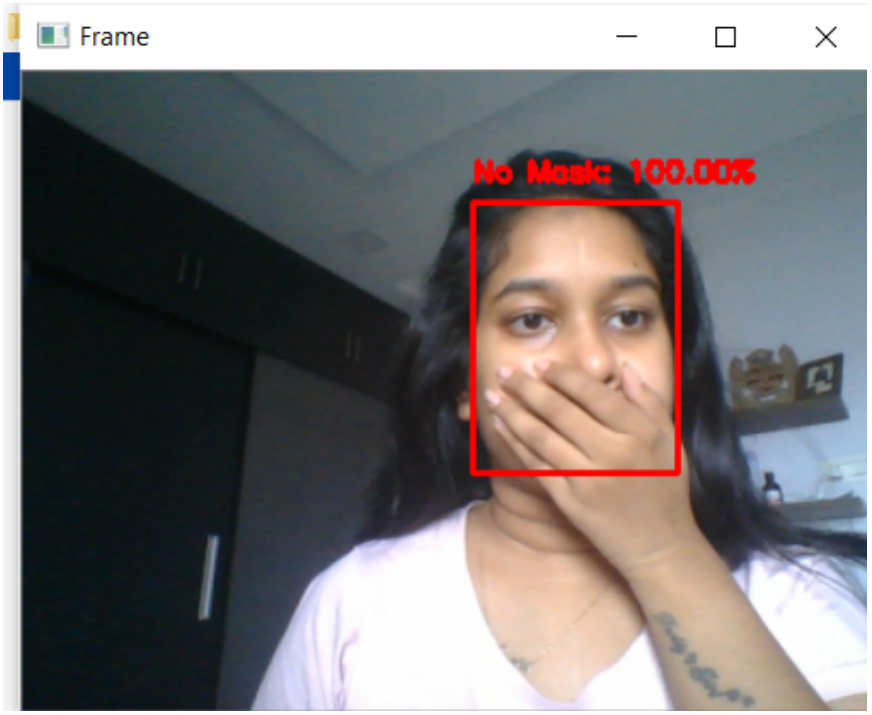

FIGURE 5.5: Obstacle differentiation

The figure above shows a person putting hand in place of mask.

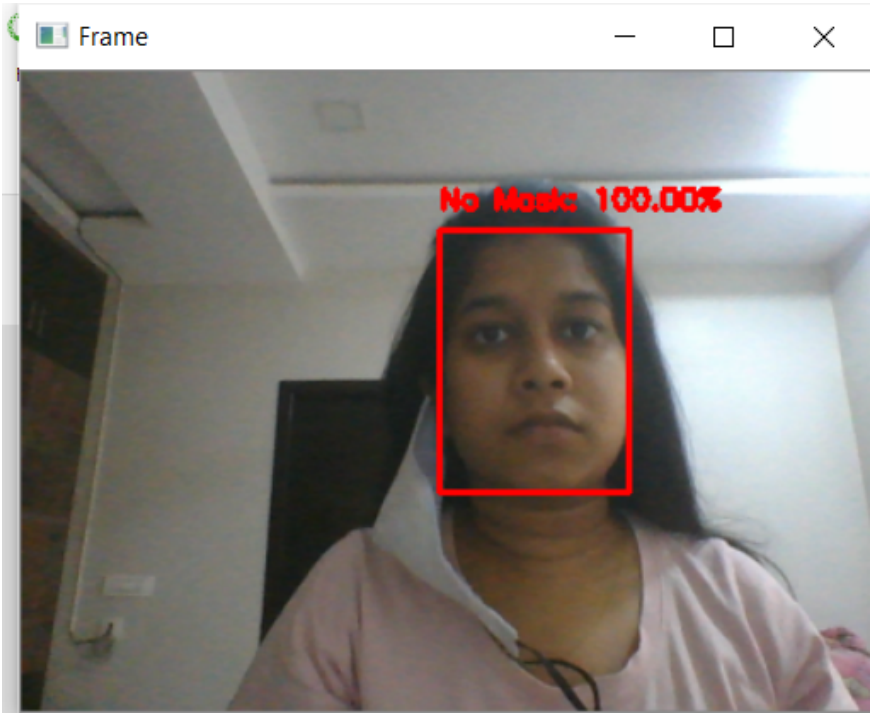

FIGURE 5.6: Mask position

The figure above shows a person putting mask in the wrong position.

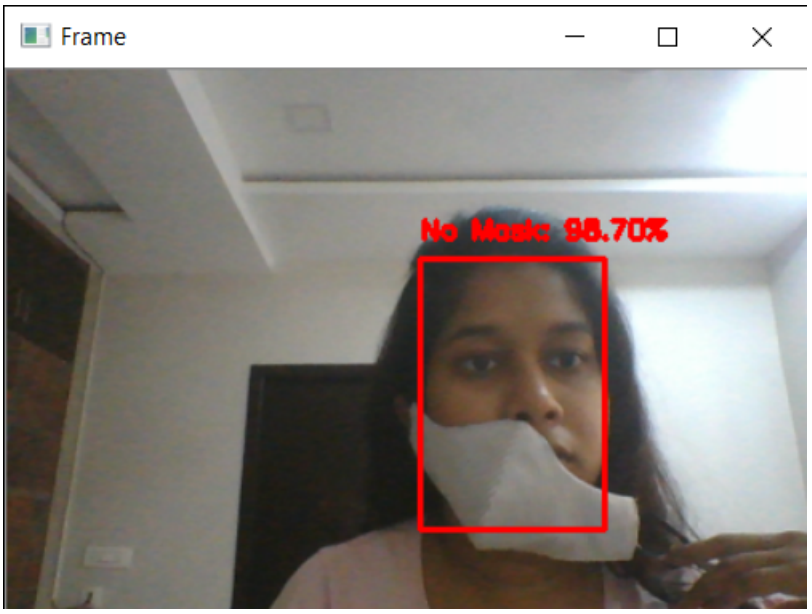

FIGURE 5.7: Mask position

The figure above shows a person putting mask in the wrong position.

5.2 Comparative Study

There are several tools and architectures available today to create similar applications. Among the available convolutional neural network architectures MobileNetV2 architecture proved to be more computationally efficient and light weight. The lightweight deep neural network makes the MobileNetV2 architecture more easier to deploy in mobile applications. When it comes to a face mask detection system it is necessary for it to be deployable in mobile applications and systems. Each type of CNN models has variety of features and the appropriate model should be selected based on the need of the proposed system.
As the MobileNetV2 model is made of light weight deep neural network, it consumes less time. The architecture After the completion of the development process, here are a few aspects that stand out from other face mask detection system. The proposed system shows how accurately the captured the person in the live video stream is wearing his face mask if and only if he is wearing one, otherwise it shows no face mask detected.

5.3 Discussions

The face mask detection system was developed to identify people wearing face masks. Different waves of the virus is halting the livelihood of common public in order to prevent the spread of virus face masks are necessary. This application can be deployed to make sure that the COVID 19 protocol is followed properly in various institutions and thus reduce the spread of the virus. This application is strictly to stay within the limits of the institution in which it is deployed for the concerns of security and privacy of the general public. The code, logic and wireframes are allowed to be shared amongst the internal organizations for a more guided development process.

5.4 Conclusions

This project was created, keeping in mind the need of face masks in the present pandemic situation. The work is motivated by people who disobey the laws that are in place to prevent the spread of coronavirus. A deep learning algorithm is used to detect the mask on the face in the system's face mask detection architecture. Labeled image data were used to train the model, with the images being facial images with and without

masks. A face mask is detected by the proposed device.Detecting whether people are wearing masks or not are done using OpenCV, TensorFlow, Keras, PyTorch and MobileNet. Images and real time videos were used to test the model. The optimization of the model is a continuous process, by tunning the hyper parameters, the accuracy of the model is achieved. In the current situation the face mask detector can be used to verify whether the person is not wearing the mask properly and restrict his entry.The system proposed in this study will act as a valuable tool to strictly impose the use of a facial mask in public places for all people.The creation of the application is challenging and commands focused efforts and time.

The application has immense potential and an ability to issue well-planned business requirements and ideas that may help to improve the present pandemic scenario.Despite the involved complexity, the application development and all its aspects were thoroughly enjoyed. The number of things learned in the period is simply not quantifiable. The development of the application has motivated the student with an ardent interest in the domain of 'Machine learning' and opened a window of opportunity to pursue it. Most importantly, an exposure to how projects are handled.

5.5 Scope for Future Work

Future work concerns trying different methods and further developing the work by adding new functionalities or exploring out of curiosity. Face masks have recently become mandatory in more than fifty countries around the world. In public places such as malls, public transportation, offices, and shops, people must cover their faces. Technology is often used by businesses to count the number of customers who visit their shops. They would also want to monitor impressions on digital displays and advertising screens.The Face Mask Detection tool can be improved and released as an open-source project. This programme can be used to identify people without a mask using any current USB, IP, or CCTV cameras. This live video feed for identification can be integrated into web and desktop applications so that the operator can be alerted. If someone isn't wearing a mask, software operators may even get a picture of the person.Additionally, an alarm device may be installed to sound a beep if someone enters the area without wearing a mask. Only people wearing face masks can enter the premises when this application is linked to the entrance gates.

Citing the latest strain of COVID, the way the model is constructed could also be changed, instead of just identifying mask it can be further improved to detect if the

mask is virus prone or not i.e. the type of the mask is surgical, N95 or not.It can also be further developed to detect violation of social distancing in public spaces by assessing the gap between people. Further improving the algorithm to identify the person even though he/she is wearing a mask.

Obviously, the use of other types of individual representations and functions could be investigated since they have an important influence on the results obtained at the end.

Bibliography

[1]. A. Rota, M. S. Oberste, S. S. Monroe, W. A. Nix, R. Campagnoli, J. P. Icenogle, S. Penaranda, B. Bankamp,K. Maher, M.-h. Chenet al., "Characterization of a novel coronavirus associated with severe acute respiratory syndrome,"science, vol. 300, no. 5624, pp. 1394–1399, 2003.

[2]Vinitha, V., Velantina, V. (2020). Covid-19 Facemask Detection With Deep Learning and Computer Vision. Int. Res. J. Eng. Technol, 7(8), 3127-3132.

[3] Z. A. Memish, A. I. Zumla, R. F. Al-Hakeem, A. A. AlRabeeah, and G. M. Stephens, "Family cluster of middleeast respiratory syndrome coronavirus infections,"New England Journal of Medicine, vol. 368, no. 26, pp.2487–2494, 2013.

[4]C. Li, Y. Diao, H. Ma and Y. Li., 2008. "A Statistical PCA Method for Face Recognition", Available from: https://ieeexplore.ieee.org/abstract/document/4740022/citationscitations

[5] Y. Liu, A. A. Gayle, A. Wilder-Smith, and J. Rocklöv, "The reproductive number of covid-19 is higher comparedto sars coronavirus,"Journal of travel medicine, 2020.

[6]Melissa P. Johnston. 2014. "Secondary Data Analysis: A Method of which the time Has Come" Available from: http://www.qqml-journal.net/index.php/qqml/article/view/169/170

[7] Y. Fang, Y. Nie, and M. Penny, "Transmission dynamics of the covid-19 outbreak and effectiveness of governmentinterventions: A datadriven analysis,"Journal of medical virology, vol. 92, no. 6, pp. 645–659, 2020.

[8] N. H. Leung, D. K. Chu, E. Y. Shiu, K.-H. Chan, J. J. McDevitt, B. J. Hau, H.-L. Yen, Y. Li, D. KM, J. Ipet al.,"Respiratory virus shedding in exhaled breath and efficacy of face masks."

[9]F. H. Alhadi, W. Fakhr and A. Farag. 2005."Hidden Markov Models for Face Recognition", Available from: https://www.researchgate.net/publication/220939899

[10] S. Feng, C. Shen, N. Xia, W. Song, M. Fan, and B. J. Cowling, "Rational use of face masks in the covid19pandemic,"The Lancet Respiratory Medicine, 2020.

[11] Z. Wang, G. Wang, B. Huang, Z. Xiong, Q. Hong, H. Wu, P. Yi, K. Jiang, N. Wang, Y. Peiet al., "Masked facerecognition dataset and application,"arXiv preprint arXiv:2003.09093, 2020.Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review,"IEEE transactions on neural networks and learning systems, vol. 30, no. 11, pp. 3212–3232, 2019.

[12] A. Kumar, A. Kaur, and M. Kumar, "Face detection techniques: a review,"Artificial IntelligenceReview, vol. 52,no. 2, pp. 927–948, 2019.D.-H. Lee, K.-L. Chen, K.-H. Liou, C.-L. Liu, and J.-L. Liu, "Deep learning and control algorithms of direct perception for autonomous driving,2019.

[13]Joseph Redmon, Santosh Divvalay, Ross Girshick, Ali Farhadi. 2016. "You Only Look Once: Unified, Real-Time Object Detection". Available from: https://arxiv.org/abs/1506.02640

[14]Laurenz Wiskott, Jean-Marc Fellous, Norbert Kr uger, and Christoph von der Malsburg (1999). "Face Recognition by Elastic Bunch Graph Matching". Available from: https://www.researchgate.net

[15]Paul Viola and Michael Jones. 2001. "Rapid Object Detection using a Boosted Cascade of Simple Features". Available from: http://web.iitd.ac.in/ sumeet/viola-cvpr-01.pdf

[16]W. Zhao, R. Chellappa, P. J. Phillips, A. Rosenfeld. 2003. "Face Recognition: A Literature Survey", Available from: https://inc.ucsd.edu/ marni/Igert/$Zhao_2003.pdf$

Appendix A

A.1 Adam Optimizer

Adam understands the value of AdaGrad and RMSProp. Instead of using the average first moment (the mean) as in RMSProp, Adam uses the average of the second moments of the gradients to adjust the parameter learning rates (the uncentered variance).
The algorithm calculates an exponential moving average of the gradient and the squared gradient, with beta1 and beta2 controlling the decay rates of these moving averages. The tendency of moment estimates towards zero is caused by the initial value of the moving averages and beta1 and beta2 values close to 1.0 (recommended). To overcome this bias, first calculate the bias results, then calculate the bias-corrected estimates.

It has many advantages like:

- Straightforward to implement.
- Computationally efficient.
- Little memory requirements.
- Invariant to diagonal rescale of the gradients.
- Well suited for problems that are large in terms of data and/or parameters.
- Appropriate for non-stationary objectives.
- Appropriate for problems with very noisy/or sparse gradients.

- Hyper-parameters have intuitive interpretation and typically require little tuning. Adam is not the same as traditional stochastic gradient descent.For all weight changes, stochastic gradient descent retains a single learning rate (called alpha), which does not alter during training.Each network weight (parameter) has its own learning rate, which is adjusted separately as learning progresses. Adam essentially incorporates the benefits of two other stochastic gradient descent extensions. Particularly:

- Adaptive Gradient Algorithm (AdaGrad) increases performance on problems with sparse gradients by maintaining a per-parameter learning rate (e.g. natural language and computer vision problems).

- Root Mean Square Propagation (RMSProp) also retains per-parameter learning rates that are adjusted based on the average of recent gradient magnitudes for the weights (e.g. how quickly it is changing). This indicates that the algorithm is effective for both online and non-stationary problems (e.g. noisy).

A.2 Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.
The most important function in the matplotlib library is pyplot(), which is used to plot 2D data. The equation y = 2x + 5 is plotted using the script below.

Example:

```
import numpy as np
from matplotlib import pyplot as plt
x = np.arange(1,11)
y = 2 * x + 5
plt.title(”Matplotlib demo”)
plt.xlabel(”x axis caption”)
plt.ylabel(”y axis caption”)
plt.plot(x,y)
plt.show()
```

As the values on the x axis, an ndarray object x is generated using the np.arange() function. The values on the y axis are stored in a separate ndarray object called y. The plot() feature of the pyplot submodule of the matplotlib package is used to plot these values. The show() function displays the graphical representation.The performance from the code above should be as follows:

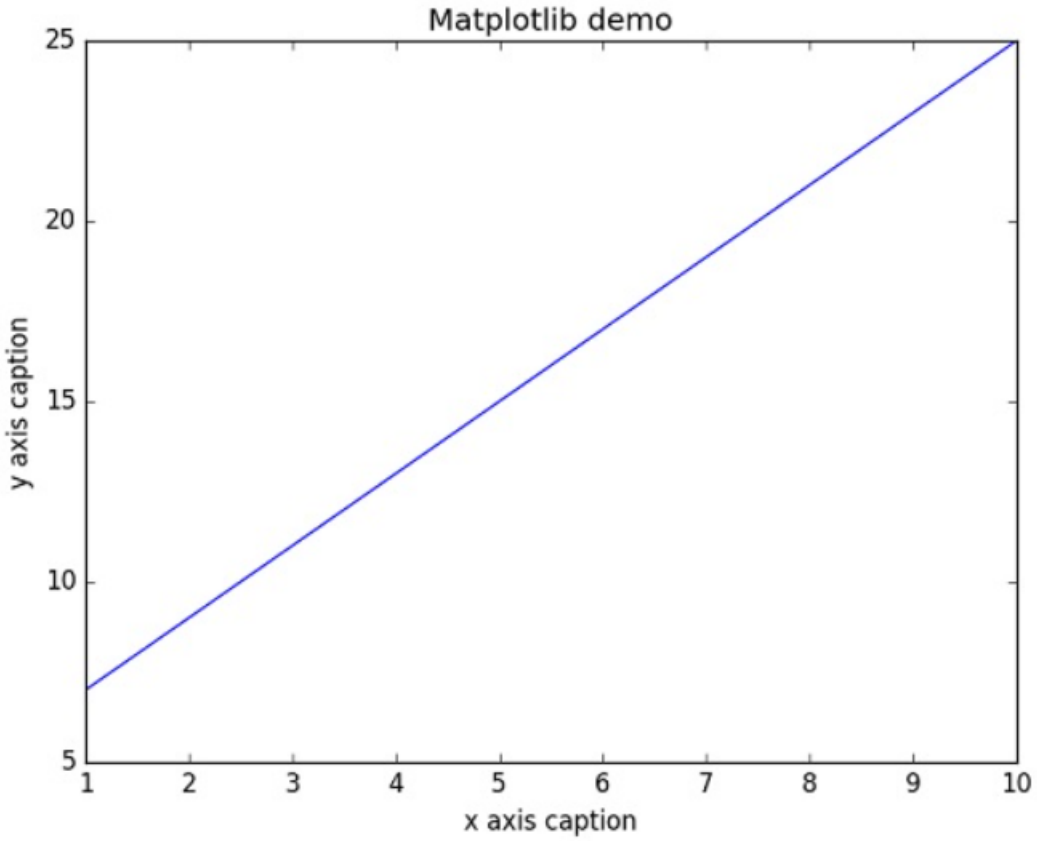

FIGURE A.1: Matplotlib