DAYANANDA SAGAR UNIVERSITY

MINOR PROJECT REPORT

ON

"Mask Detection and Face Recognition"

BACHELOR OF TECHNOLOGY

IN

COMPUTERSCIENCE & ENGINEERING

Submitted by

Kaushal Yadav (ENG19CS0139)
Krishna Kant Vishnoi (ENG19CS0147)
Lakshman Parihar M (ENG19CS0152)
Ujjwal Kumar (ENG19CS0342)
Vivek Kumar Mishra (ENG19CS0365)

5th SEMESTER, 2022

Under the Guidance of

Prof. Gousia Thahniyath
(Assistant professor)

DAYANANDA SAGAR UNIVERSITY

School of Engineering, Kudlu Gate ,Bangalore-560068

CERTIFICATE

*This is to certify that the minor project report entitled "***Mask Detection and Face Recognition"** *being submitted by* ***Mr LAKSHMAN PARIHAR M*** *bearing USN* ***ENG19CS0152****. has* satisfactorily completed her Minor Project as prescribed by the University for the 5th semester B.Tech Program in Computer Science & Engineering during the academic year 2021 – 22 at the School of Engineering, Dayananda Sagar University, Bangalore.

Date: Signature of the faculty in-charge

Signature of Chairman
Department of Computer Science & Engineering

DAYANANDA SAGAR UNIVERSITY

School of Engineering, Kudlu Gate ,Bangalore-560068

CERTIFICATE

*This is to certify that the minor project report entitled "***Mask Detection and Face Recognition"** *being submitted by* ***MR LAKSHMAN PARIHAR M, MR KRISHNA KANT VISHNOI, MR KAUSHAL YADAV, MR UJJWAL KUMAR, MR VIVEK KUMAR MISHRA.*** *bearing USN* ***ENG19CS0152 , ENG19CS0147 , ENG19CS0139 , ENG19CS0342 , ENG19CS0365****. has* satisfactorily completed her Minor Project as prescribed by the University for the 5th semester B.Tech Program in Computer Science & Engineering during the academic year 2021 – 22 at the School of Engineering, Dayananda Sagar University, Bangalore.

Date: Signature of the faculty in-charge

Signature of Chairman
Department of Computer Science & Engineering

DECLARATION

We hereby declare that the work presented in this minor project entitled as "**Mask Detection and Face Recognition**", has been carried out by us and it has not been submitted for the award of any degree, diploma or the minor project of any other college or university.

Kaushal Yadav (ENG19CS0139)

Krishna Kant Vishnoi (ENG19CS0147)

Lakshman Parihar M (ENG19CS0152)

Ujjwal Kumar (ENG19CS0342)

Vivek Kumar Mishra (ENG19CS0365)

ACKNOWLEDGEMENT

The success and final outcome of this software requirement document required a lot of guidance and assistance from many people and we are extremely privileged to have got this all through the completion of the project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank our mentor, Professors, and the Chairman for providing us an opportunity to do the Software Requirement Document and giving us all support and guidance which made us complete the project duly. We are extremely thankful to all for providing such nice support and guidance

We are especially thankful to our **Dean Dr. A Srinivas**, **Chairman Dr. Girish A** Department of Computer Science & Engineering who continuously helped us throughout the project and without his guidance, this project would have been an uphill task.

We are pleased to acknowledge **Prof. Gousia Thahniyath** Department of Computer Science & Engineering for her invaluable guidance, support, motivation and patience during the course of this mini-project work.

We are thankful for and fortunate enough to get constant encouragement, support and guidance from all Teaching staff of the Computer Science Engineering department, which helped us in completing our report. Also, we would like to extend our sincere esteems to all staff in the laboratory for their timely support.

Kaushal Yadav (ENG19CS0139)
Krishna Kant Vishnoi (ENG19CS0147)
Lakshman Parihar M (ENG19CS0152)
Ujjwal Kumar (ENG19CS0342)
Vivek Kumar Mishra (ENG19CS0365)

ABSTRACT

The project is based on Deep Learning, which is a sub field of machine learning, concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. In the project, TensorFlow is used, which is an open-source software library created by Google for machine learning applications. It is used for detecting, identifying and tracking objects through the camera in real time. The project uses MobileNetV2 network which is trained on ImageNet dataset to train the model to detect face masks in real time. It detects the face mask, recognizes faces without mask and creates an excel sheet with the names of recognized faces. The main purpose of this project is to create excel sheets with the name of students without masks in college.

Key words : Deep Learning, Image Processing, Tensorflow, Keras

Table of Contents

Chapter 1.

Introduction

Coronavirus disease (COVID-19) is the latest epidemic caused by the newly discovered. Due to this deadly virus, WHO has made wearing masks compulsory to protect ourselves from this virus. The government also issued guidelines that the person found without face mask will be charged fine.

The project uses MobileNetV2 network which is trained on ImageNet dataset to train the model to detect face masks in real time and uses Viola Jones algorithm to detect faces. Here, Deep learning is used to train the machine for particular set of objects so that a system can be implemented for detecting face masks and recognizing faces without masks in real-time.

1.1 Overview

This project is a system which will help the authorization admin to list the students without masks. Mask detection and Face recognition is all about a system which is dedicated for safety of institution. Mask detection and Face Recognition is a project that is used for mask detection and face recognition of the students in the college and display names of students recognized without masks to admin.

We have chosen this topic because wearing masks in public is must to be safe from deadly disease. If the person does not wear masks in public, he/she is more likely to be affected by disease.

In this project we are trying to keep college safe from the deadly virus by charging students fine without masks. If each and every student will wear mask, then college is more likely able to be safe and less in danger.

1.2 Background and Motivation

As the holy grail of computer vision research is to tell a story from a single image or a sequence of images, object detection and recognition has been studied for more than four decades. Significant efforts have been paid to develop representation schemes and algorithms aiming at recognizing generic objects in images taken under different imaging conditions (e.g., viewpoint, illumination, and occlusion).

Mask Detection and Face Recognition becomes a necessity when there is a need of automation, where the identification is done by machines instead of doing it manually for better performance and reliability. Mask Detection and Face Recognition is the process of automatically detecting face masks and recognizing faces without mask. Being able to charge fines for students without mask will in turn keep the institution more safe and less dangerous.

Everyone need to weak mask in institution to be safe from the deadly virus.

1.3 Problem Statement and Objectives

After the breakout of the worldwide pandemic COVID-19, there arises a severe need of protection mechanisms in order to prevent the spread of disease. From the very basic hygiene standards to the treatments in the hospitals, people are doing all they can for their own and the society's safety; face masks are one of the personal protective equipment's. People wear face mask once they step out of their homes and authorities strictly ensure that people are wearing face masks while they are in groups and public places. The basic aim of this project is to detect the presence of face mask on human faces on live streaming video and also recognize the face of human who is not wearing mask.

It is now mandatory for every human to wear mask at public places, still there

are some people who don't follow this rule. Earlier, government issued guideline which stated that the person not wearing mask will have to pay a fine of few rupees; which in turn resulted in decreasing the number of people not wearing mask.

Thus, the system implemented has the following objectives:

1. **Objective 1:** To develop a Mask Detection and Face Recognition system.
2. **Objective 2:** To develop a system which can detect face masks and face using Deep Learning.
3. **Objective 3:** To develop a system which will recognize faces without mask.
4. **Objective 4:** To design a system which will create an everyday excel file with the names of person without mask.
5. **Objective 5:** To develop a deep learning model that will train on the dataset and then classify faces without masks.

1.4 Scope of the Project

We are providing a system to institutions admin which will help them to charge fine on the persons recognized without face masks which in turn will decrease the number of people without mask and will probably keep the institution safe.

- It can be used to count the number of students that enter the college campus without masks.
- People without masks can be counted in crowdy public places such as temples, shopping malls, industrial and corporate areas, concerts etc.
- Admin will be able to charge fine on all the students marked in an excel sheet.
- We are training a model which is going to detect face masks and recognize faces without mask.

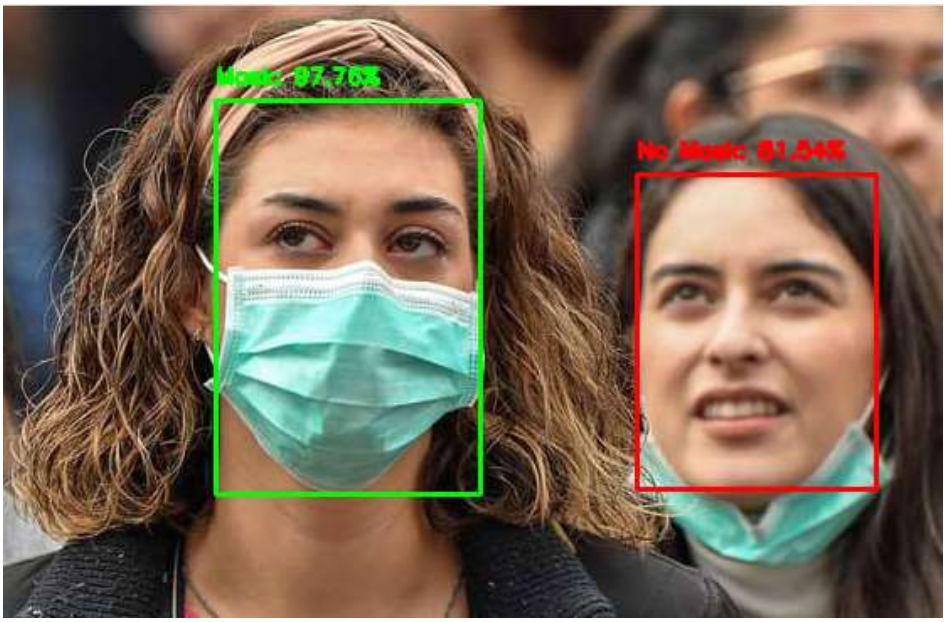


Figure 1-1: Detecting faces with mask and without masks.

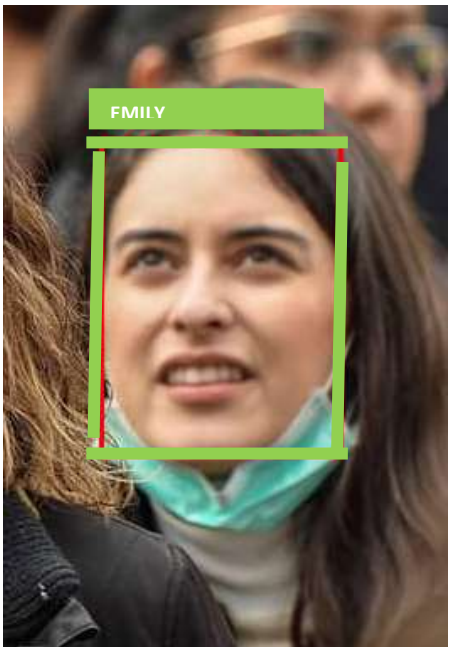


Figure 1-2: Recognizing faces without masks.

- This project ensures that each and every face is recognized without mask.
- It will reduce the risk of virus in institution.

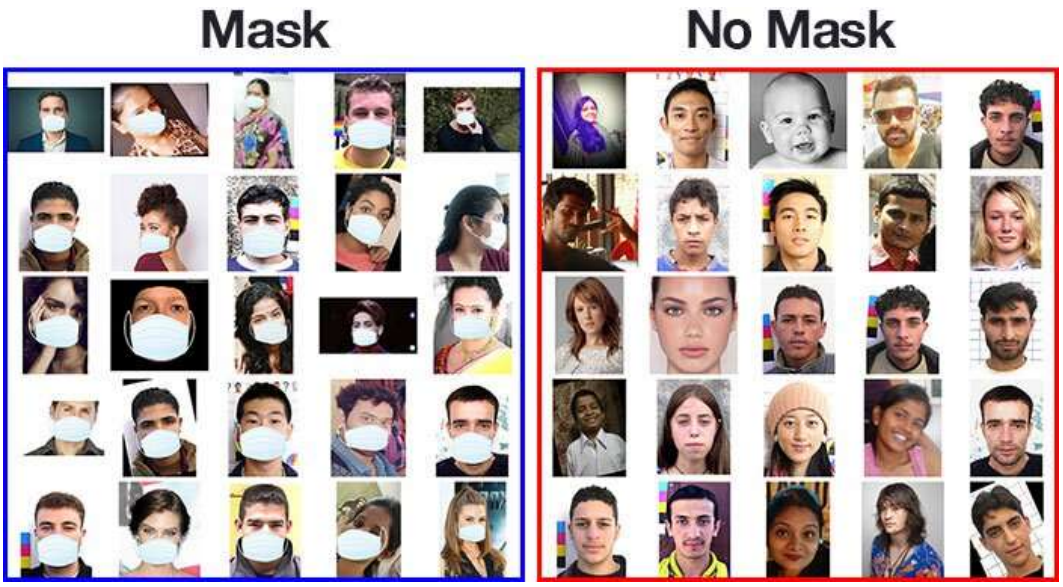


Figure 1-3: Dataset used

1.5 Team Organization

- **Vivek kumar Mishra:**

  Along with doing preliminary investigation and understanding the limitations of current system, I studied about the topic and its scope and documentation is also done by me.

- **Kaushal yadav:**

  Along with doing preliminary investigation and understanding the limitations of current system, I surveyed various research papers related to object detection and face recognition. Implementation logic for the project objective and coding of internal functionalities is also done by me.

- **Krishna kant Vishnoi:**

  I investigated and found the right technology and studied in deep about it. For the implementation of the project, I collected dataset and trained the model for it. Implementation logic for the project objective and coding of internal functionalities is also done by me.

- **Ujjwal Kumar:**

  Along with doing preliminary investigation and understanding the limitations of current system and documentation is also a part done by me.

- **Lakshman Parihar:**

  I investigated and found the right technology and studied in deep about it. I have also worked on GUI of the project, and documentation is also a part done by me.

1.6 Report Structure

The project ***Mask detection and Face Recognition*** is primarily concerned with the **detecting faces without mask and recognizing it** and whole project report is categorized into five chapters.

Chapter 1: Introduction- introduces the background of the problem followed by rationale for the project undertaken. The chapter describes the objectives, scope and applications of the project. Further, the chapter gives the details of team members and their contribution in development of project which is then subsequently ended with report outline.

Chapter 2: Review of Literature- explores the work done in the area of Project undertaken and discusses the limitations of existing system and highlights the issues and challenges of project area. The chapter finally ends up with the requirement identification for present project work based on findings drawn from reviewed literature and end user interactions.

Chapter 3: Proposed System - starts with the project proposal based on requirement identified, followed by benefits of the project. The chapter also illustrate software engineering paradigm used along with different design representation. The chapter also includes block diagram and details of major modules of the project. Chapter also gives insights of different type of feasibility study carried out for the project undertaken. Later it gives details of the different deployment requirements for the developed project.

Chapter 4: Implementation - includes the details of different Technology/ Techniques/ Tools/ Programming Languages used in developing the Project. The chapter also includes the different user interface designed in project along with their functionality. Further it discusses the experiment results along with testing of the project. The chapter ends with evaluation of project on different parameters like accuracy and efficiency.

Chapter 5: Conclusion - Concludes with objective wise analysis of results and limitation of present work which is then followed by suggestions and recommendations for further improvement.

Chapter 2.

Review of Literature

Although, there are many systems who are performing same task of face mask detection and Face recognition, but none of them is combining these two together.

In this project, we are trying to keep the institution less affected from the virus and all people are safe. Admin will charge fine from the people not wearing mask so that they will wear it every time.

2.1 Preliminary Investigation

2.1.1 Current System

- The Current System for detecting masks detects whether the person is wearing face mask or not. The person not wearing masks will be instructed by the authorities orally to wear masks.
- The presence of a CCTV camera at the entrance of the college campus which has no real-time significance and having just a camera cannot solve the problem of recognizing faces of people without masks.

2.2 Limitations of Current System

The limitations of these are as follows :

- At the economic front, it is not feasible to have a person who only stands at the gate and instructs people to wear face masks.
- Twins are hard to recognize.
- Person walking keeping his head down is difficult to detect and recognize.
- When a CCTV camera is used, students or persons without masks cannot be recognized but, in our system, they can be recognized in real-time.

2.3 Requirement Identification and Analysis for Project

Significant work has been done in the field of Mask Detection and Face Recognition; however, it is not easy to achieve desired results. The review of literature leads to draw certain major findings which are as under:

- Object detection is one of the most relevant topics in image processing and computer vision. From the small, to the personal application of the large-scale industrial applications, object detection and recognition is used in a wide range of industries. A few examples include image search, security intelligence, OCR, health care, and agricultural monitoring. [1]

- There are many problems which machine faces in learning detection from a small training database. Machine's performance depends mainly on features used for object representation. Specifically, OpenCV facial hair-like features were used to identify face recognition, the Principal Component Analysis (PCA) was employed in quick extraction of face features and the Euclidean Distance was also adopted in face recognition; as thus, data amount and computational complexity would be reduced effectively in face recognition, and the face recognition could be carried out on embedded platform. [2]

-

- One new approach used for object detection is YOLO. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance. Their base YOLO model processes images in real-time at 45 frames per second. [3]

  There is also a very fast image pre-processing by the introduction of a linearly shaded elliptical mask centred over the faces. Used in association with DCT, for features extraction, and MPL and RBF Neural Networks, for classification, it allows an improvement of system performances without modifying the global computation weight and also a learning time reduction for MLP neural networks. [4]

Image processing, which is a part of signal processing, takes image in the form of picture or video frame in input and outputs an image or characteristics

sets related to image. Image processing is mainly performed with the help of OpenCV library. OpenCV contains various tools to solve problems of computer vision. It consists of low-level image processing functions and high-level algorithms for face detection, feature matching and tracking. [5]

2.3.1 Conclusion

This chapter reviews the literature surveys that have been done during the research work. The related work that has been proposed by many researchers has been discussed. The research papers related to mask detection and face recognition from 2016 to 2021 have been shown which discussed about different methods and algorithm to detect masks.

Chapter 3.

Proposed System

3.1 The Proposal

The proposal is to deploy a system at the entrance which can identify face masks and recognize the faces without masks through webcam or CCTV cameras and perform the operation of creating an excel sheet of names of people without masks for every day.

It will help the admin of institution to keep their institution safe from the deadly virus, as each and every person will be wearing face masks.

3.2 Benefits of the Proposed System

The current system had a lot of challenges that are overcome by this system:

- **Economic:** The proposed system is economic as there will not be any person required to keep a watch on the entrance.
- **Real-Time Observation:** Unlike CCTV, masks can be identified also in real-time and can be saved for later use.
- **Man Power:** It does not require any person or their efforts to stand and note the names of people without mask.
- **24 x 7 Availability:** Camera implemented with this does not require the person to keep check on people every time.
- **Statistical analysis:** The number of faces can be recognized individually and kept a record for calculating various factors.

3.3 Flow Diagram

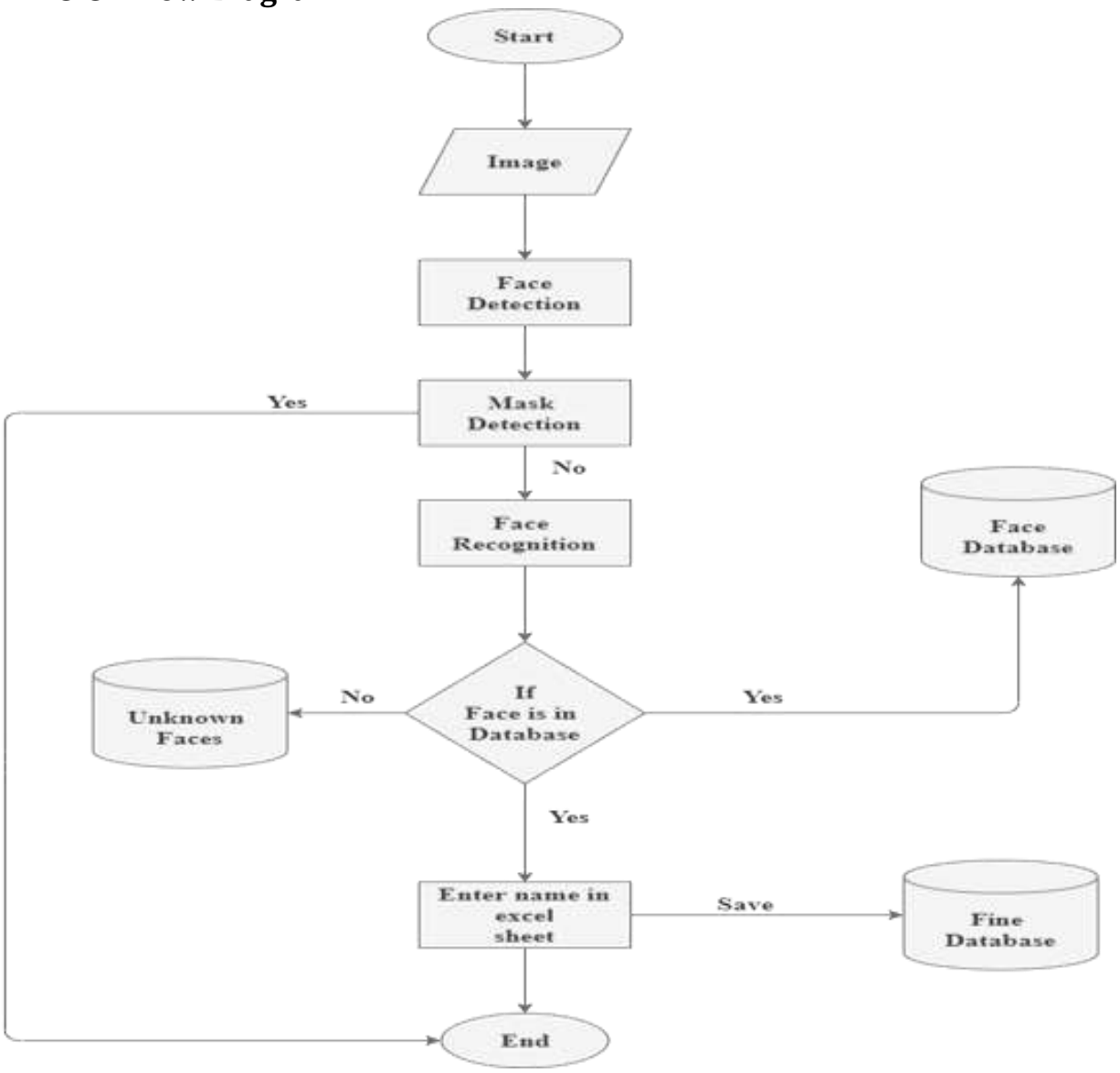


Figure 3-1: Flow Diagram

3.4 Feasibility Study

A feasibility study is an analysis of how successfully a system can be implemented, accounting for factors that affect it such as economic, technical and operational factors to determine its potential positive and negative outcomes before investing a considerable amount of time and money into it.

3.4.1 Technical

For any real-time detection system, there is a need to process images from the video. For this, the kind of framework used must be the one that is capable of extracting those objects from the images easily and accurately in real-time. The framework used in this is Tensorflow, which is a framework designed by Google for efficiently dealing with deep learning and concepts like neural networks, making the system technically feasible.

The system once set up completely, works automatically without needing any person to operate it. The result gets automatically saved in the database, without requiring any manual effort for saving it.

For making the system technically feasible, there is a requirement of GPU built system with high processor for better performance.

3.4.2 Economical

For any real-time mask detection and face recognition system, there is a need of a High-definition Camera for better and accurate results.

Since the system is completely automated, there is a need of continuous electricity supply for it to operate 24X7.

The Tensorflow framework used in the system works great with GPU built systems, which are a little on the expensive side.

Since the system uses high performance processors continuously, so to save any disaster from occurring due to very high temperatures, there is a requirement of a cooling system in the environment where it is implemented.

3.4.3 Operational

The main motto of our system is to reduce the chances of increasing dangerous virus in the institution.

The system is able to do that accurately and efficiently making the system operationally feasible.

3.5 Design Representation

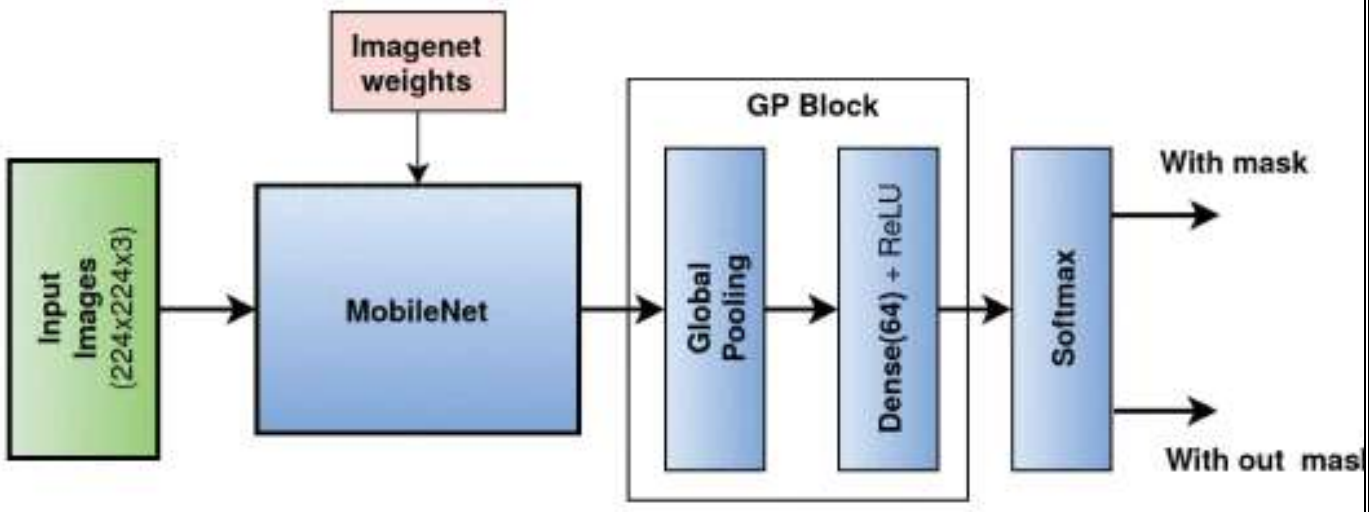


Figure 3-2: MobileNet CNN Diagram

3.5.1 Data Flow Diagrams

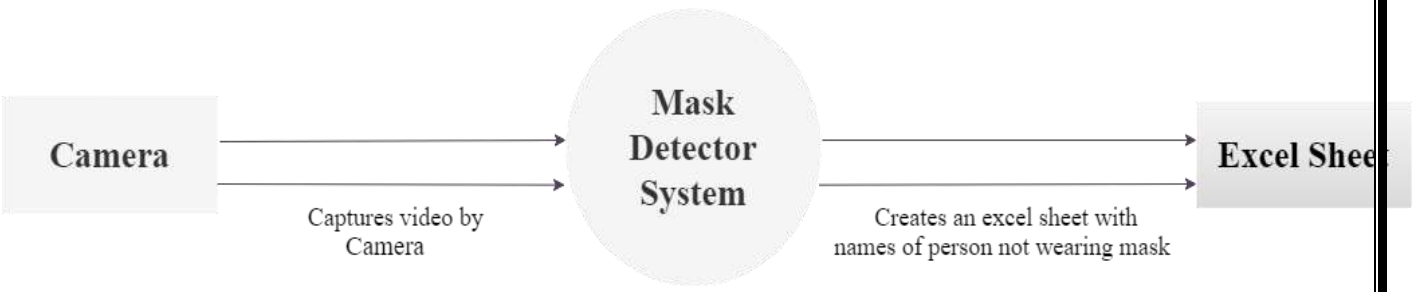


Figure 3-3: Data Flow Diagram Level 0

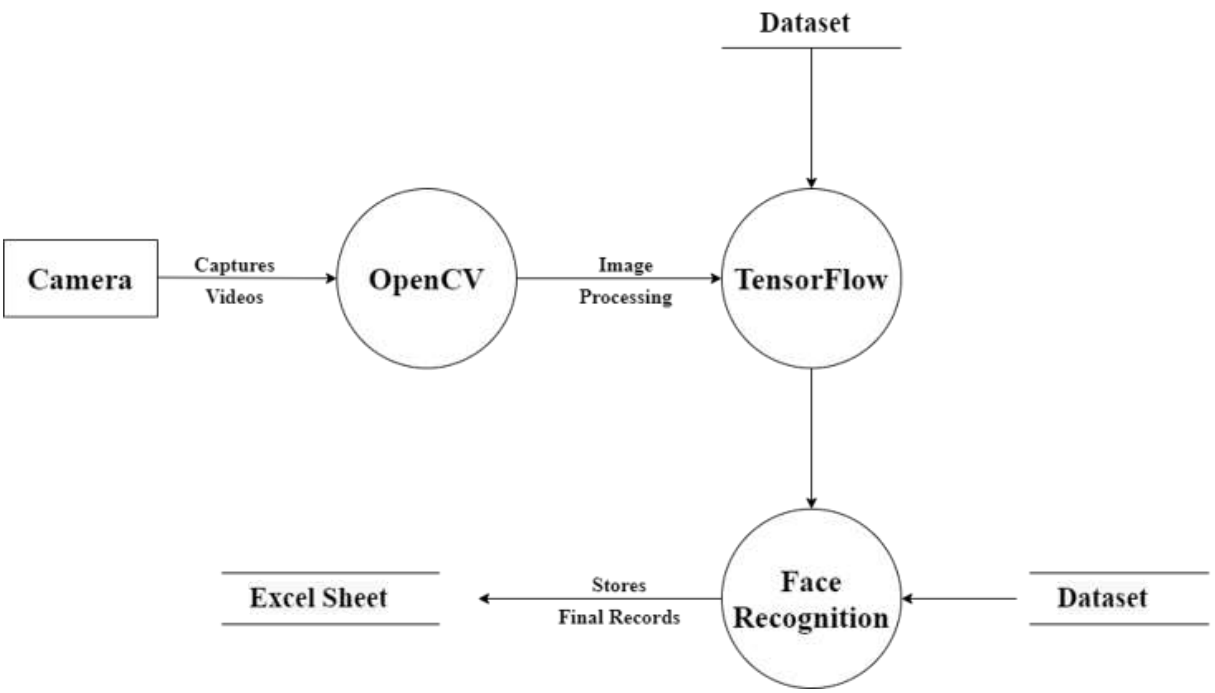


Figure 3-4: Data Flow Diagram Level 1

3.5.2 Sequence Diagram

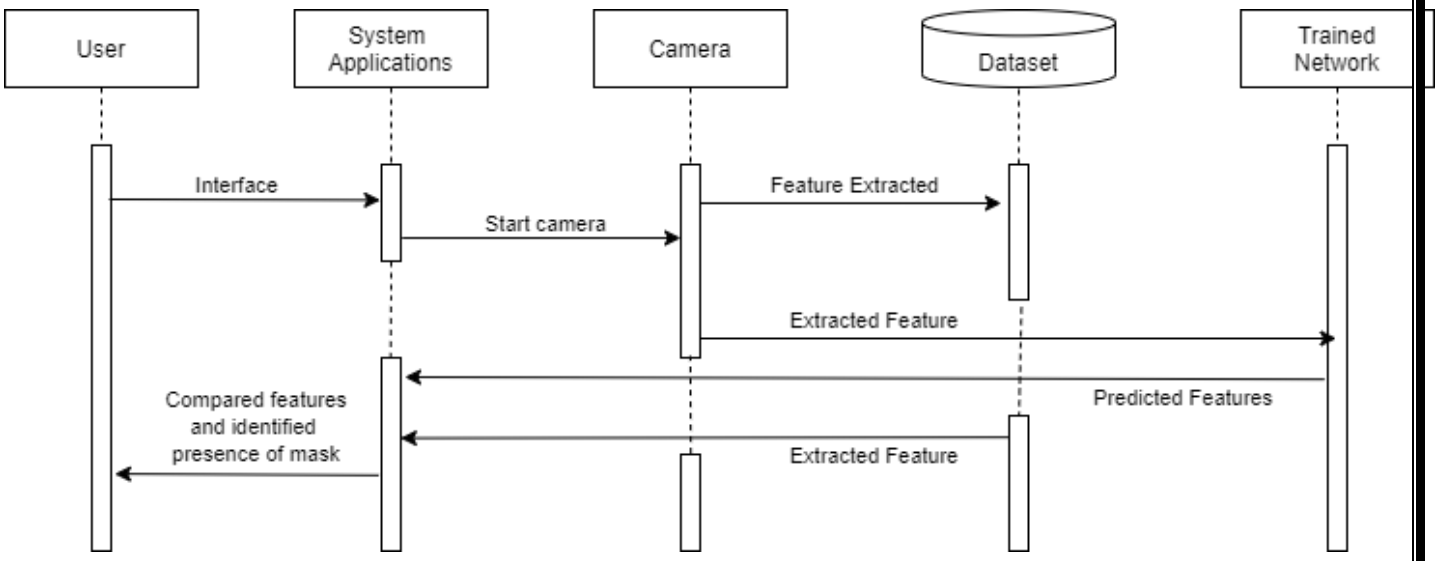


Figure 3-5: Sequence Diagram

3.5.3 Use Case Diagram

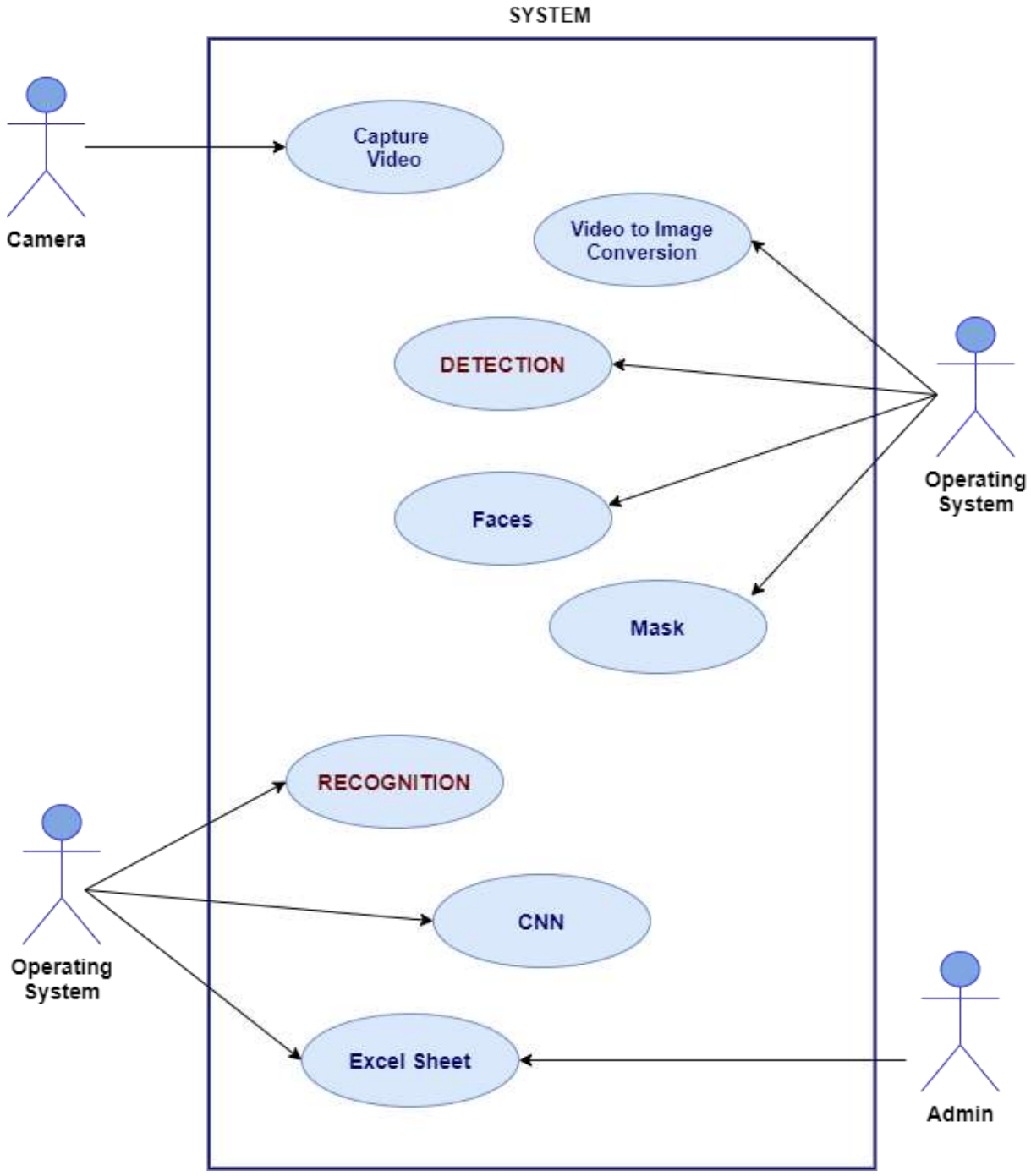


Figure 3-5: Use Case Diagram

3.6 Deployment Requirements

There are various requirements (hardware, software and services) to successfully deploy the system. These are mentioned below:

3.6.1 Hardware

- 32-bit, x86 Processing system
- Windows 7 or later operating system
- High processing computer system without GPU or with GPU (high performance)
- High- definition Camera

3.6.2 Software

- OpenCV
- Python and its supported libraries
- Tensor Flow
- If Installing Tensorflow in GPU systems:
  1. CUDA® Toolkit 9.0.
  2. The NVIDIA drivers associated with CUDA Toolkit 9.0. cuDNN v7.0.
  3. GPU card with CUDA Compute Capability 3.0 or higher

Chapter 4.

Implementation

For the problem of instructing students to wear mask entering the college campus manually, the system is designed in such a way so as to automate the process by placing a camera at the entrance gate so that students without masks are recognized and noted in an excel sheet.

4.1 Technique Used

4.1.1 Deep- Learning

Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, semi- supervised or unsupervised.

Deep learning models are loosely related to information processing and communication patterns in a biological nervous system, such as neural coding that attempts to define a relationship between various stimuli and associated neuronal responses in the brain.

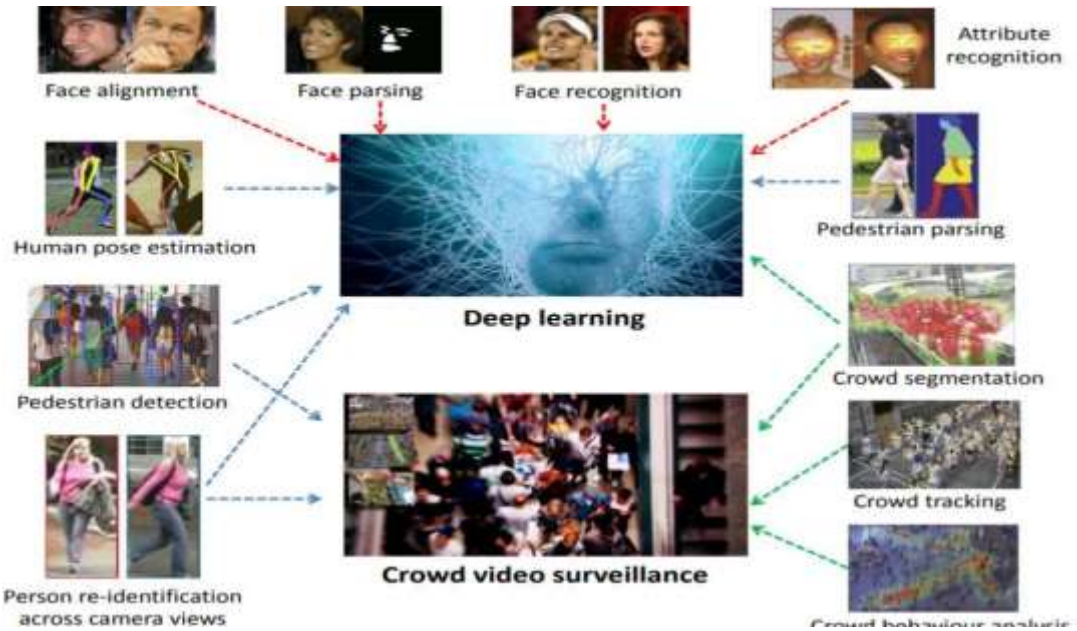


Figure 4-1: Deep Learning

Deep learning architectures such as deep neural networks, deep belief networks and recurrent neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics and drug design, where they have produced results comparable to and in some cases superior to human experts.

4.1.2 Neural Networks:

In machine learning, a convolutional neural network (CNN, or ConvNet) is a class of deep, feed-forward artificial neural networks that has successfully been applied to analyzing visual imagery.

CNNs use a variation of multilayer perceptions designed to require minimal preprocessing. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared- weights architecture and translation invariance characteristics.

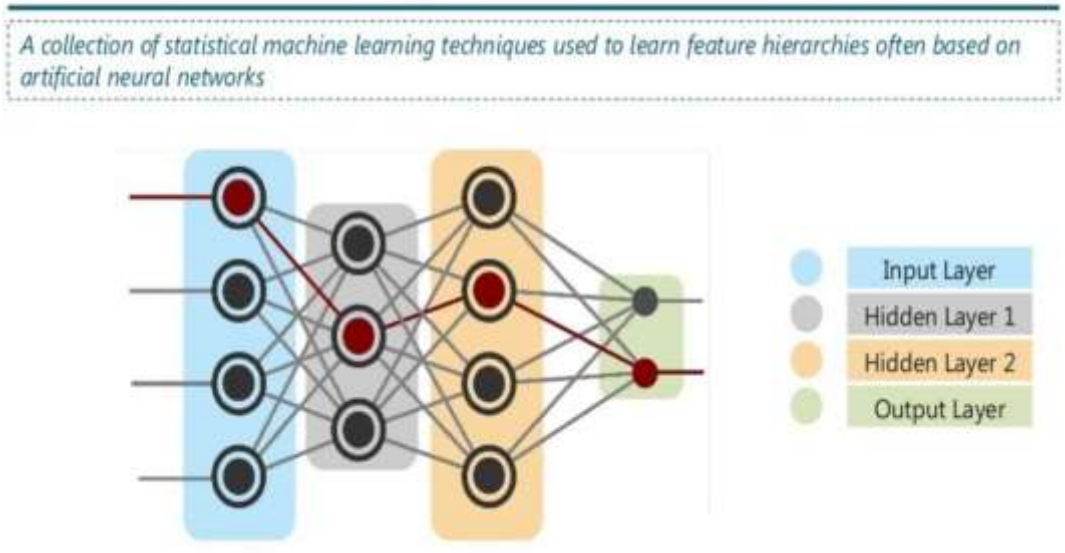


Figure 4-2: Neural Networks

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field. CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage. They have applications in image and video recognition, recommender systems and natural language processing.

4.1.3 MobileNetV2

MobileNet is a class of CNN that was open-sourced by Google, and therefore, this gives us an excellent starting point for training our classifiers that are insanely small and insanely fast.

MobileNet uses **depth wise separable convolutions.** It significantly **reduces the number of parameters** when compared to the network with regular

convolutions with the same depth in the nets. This results in lightweight deep neural networks. A depth wise separable convolution is made from two operations. **Depth wise**

convolution and **Pointwise convolution**.

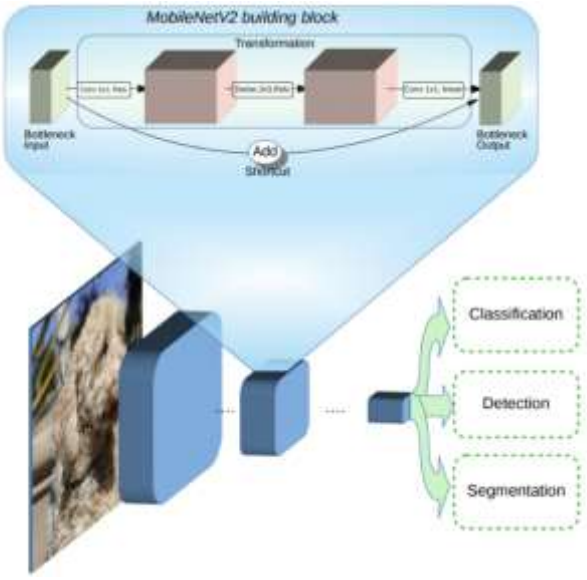


Figure 4-3: MobileNet CNN

The main difference between MobileNet architecture and a traditional CNN instead of a single 3x3 convolution layer followed by the batch norm and Relu. Mobile Nets split the convolution into a 3x3 depth-wise conv and a 1x1 pointwise conv, as shown in the figure.

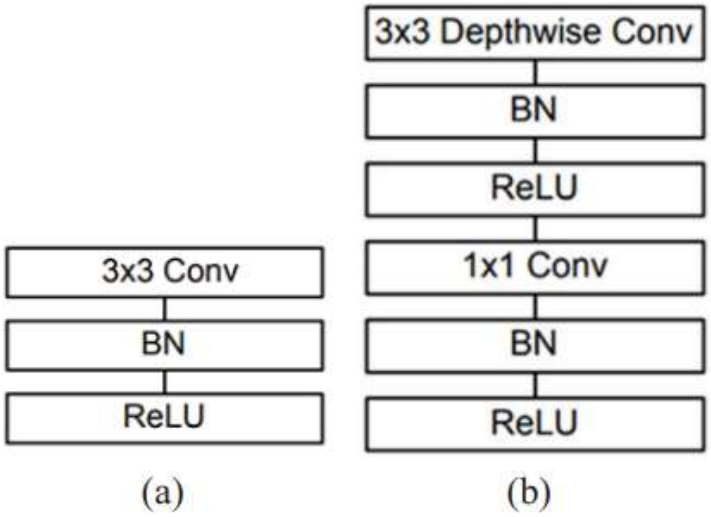


Figure 4-4: (a) Standard Convolutional layer (b) Depth-wise Separable Convolution

4.2 Tools Used

4.2.1 OpenCV

OpenCV (Open-Source Computer Vision Library) is released under a BSD license and hence it's free for both academic and commercial use. It has C++, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform.

Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics.

4.2.2 Tensor Flow

TensorFlow is an open source software library for high performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. Originally developed by researchers and engineers from the Google Brain team within Google's AI organization, it comes with strong support for machine learning and deep learning and the flexible numerical computation core is used across many other scientific domains.

Figure 4-5: TensorFlow Architecture

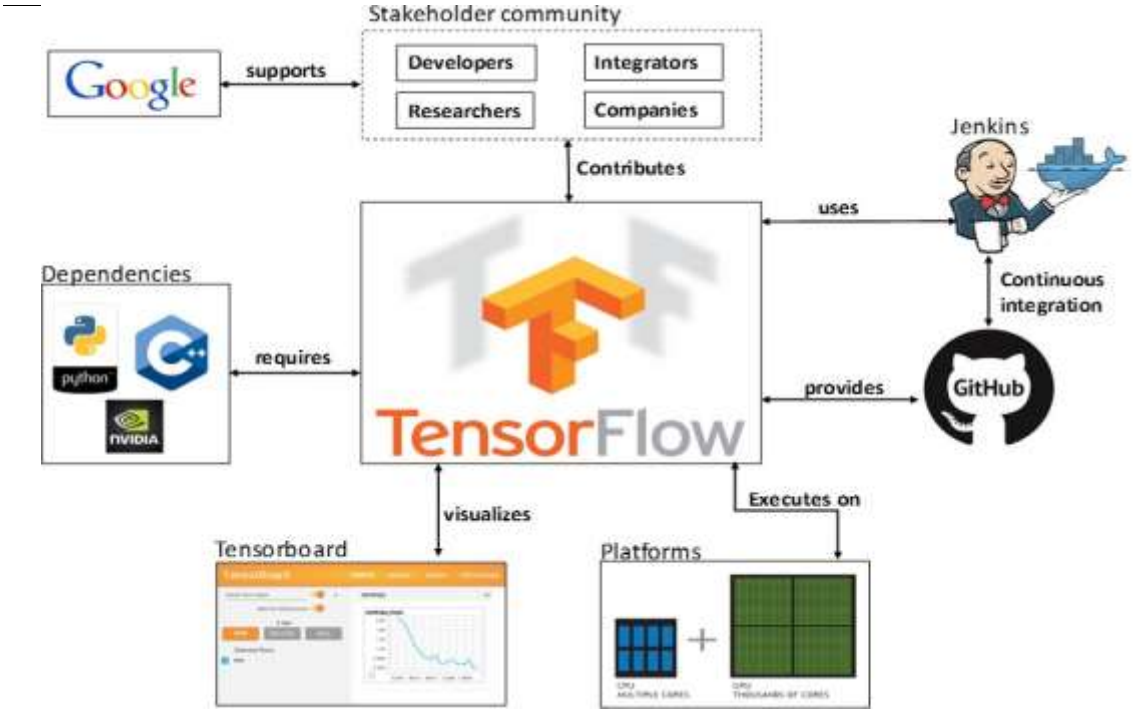


Figure 4-6: TensorFlow Working

4.2.3 Models

The TensorFlow official models are a collection of example models that use TensorFlow's high-level APIs. They are intended to be well-maintained, tested, and kept up to date with the latest TensorFlow API. They should also be reasonably optimized for fast performance while still being easy to read.

Model name	Speed (ms)	COCO mAP[^1]	Outputs
ssd_mobilenet_v1_coco	30	21	Boxes
ssd_inception_v2_coco	42	24	Boxes
faster_rcnn_inception_v2_coco	58	28	Boxes
faster_rcnn_resnet50_coco	89	30	Boxes
faster_rcnn_resnet50_lowproposals_coco	64		Boxes
rfcn_resnet101_coco	92	30	Boxes
faster_rcnn_resnet101_coco	106	32	Boxes
faster_rcnn_resnet101_lowproposals_coco	82		Boxes
faster_rcnn_inception_resnet_v2_atrous_coco	620	37	Boxes
faster_rcnn_inception_resnet_v2_atrous_lowproposals_coco	241		Boxes
faster_rcnn_nas	1833	43	Boxes
faster_rcnn_nas_lowproposals_coco	540		Boxes

Types of Models

Below is a list of the models available.

- **COCO:** COCO is a large-scale object detection, segmentation, and captioning dataset.
- **Mnist**: A basic model to classify digits from the MNIST dataset.
- **Resnet**: A deep residual network that can be used to classify both CIFAR-10 and ImageNet's dataset of 1000 classes.
- **Wide deep**: A model that combines a wide model and deep network to classify census income data.

The system uses COCO data set Model which is Common Object in Context designed by Microsoft.

COCO Model:

This database has several features:

- Object Detection
- Recognition in context
- Super pixel stuff segmentation
- 330K images (>200K labeled)
- 1.5 million object instances
- 80 object categories
- 91 stuff categories
- 5 captions per image
- 250,000 people with key points

COCO currently has three annotation types: object instances, object key points, and image captions. The annotations are stored using the JSON file format. All annotations share the basic data structure below:

```
{
   "info"              : info,
   "images"            : [image],
   "annotations"       : [annotation],
   "licenses"          : [license],
}

info{
   "year"              : int,
   "version"           : str,
   "description"       : str,
   "contributor"       : str,
   "url"               : str,
   "date_created"      : datetime,
}

image{
   "id"                : int,
   "width"             : int,
   "height"            : int,
   "file_name"         : str,
   "license"           : int,
   "flickr_url"        : str,
   "coco_url"          : str,
   "date_captured"     : datetime,
}
```

Figure 4-7: Data Structure in JSON Format

The distribution of the objects in this database can be obtained from their website. In section Explore, it is possible to choose and combine each of the objects and observe how many images these objects appear. The distribution for each of the objects in training/validation set is shown in the following image:

	images		images		images		images
person	66808	cat	4298	wine glass	2643	dinning table	12338
backpack	5756	dog	4562	cup	9579	toilet	3502
umbrella	4142	horse	3069	fork	3710	tv	4768
handbag	7133	sheep	1594	knife	4507	laptop	3707
tie	3955	cow	2055	spoon	3682	mouse	1964
suitcase	2507	elephant	2232	bowl	7425	remote	3221
bicycle	3401	bear	1009	banana	2346	keyboard	2221
car	12786	zebra	2001	apple	1662	cell phone	5017
motorcycle	3661	giraffe	2647	sandwich	2463	microwave	1601
airplane	3083	frisbee	2268	orange	1784	oven	2992
bus	4141	skis	3202	broccoli	2010	toaster	225
train	3745	snowboard	1703	carrot	1764	sink	4865
truck	6377	sports ball	4431	hot dog	1273	refrigerator	2461
boat	3146	kite	2352	pizza	3319	book	5562
traffic light	4330	baseball bat	2603	donut	1585	clock	4863
fire hydrant	1797	baseball glove	2729	cake	3049	vase	3730
stop sign	1803	skateboard	3603	chair	13354	scissors	975
parking meter	742	surfboard	3635	couch	4618	teddy bear	2234
bench	5805	tennis racket	3561	potted plant	4624	hair drier	198
bird	3362	bottle	8880	bed	3831	toothbrush	1041

Figure 4-8: Objects in training set

Finally, in order to understand better both databases, the following image shows some characteristics of them, comparing them with other important databases.

Figure 4-9: Comparison Graphs

4.3 Language Used

Python language is used in the system due to the following Characterstics:

Simple:

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English (but very strict English!). This pseudo-code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the syntax i.e. the language itself.

Free and Open Source:

Python is an example of a FLOSS (Free/Libre and Open-Source Software). In simple terms, you can freely distribute copies of this software, read the software's source code, make changes to it, use pieces of it in new free programs, and that you know you can do these things. FLOSS is based on the concept of a community which shares knowledge. This is one of the reasons why Python is so good - it has been created and improved by a community who just want to see a better Python.

Object Oriented:

Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object-oriented languages, the program is built around objects which combine data and functionality. Python has a very powerful but simple way of doing object-oriented programming, especially, when compared to languages like C++ or Java.

Extensive Libraries:

The Python Standard Library is huge indeed. It can help you do various things involving regular expressions, documentation generation, unit testing,

threading, databases, web browsers, CGI, ftp, email, XML, XML-RPC, HTML, WAV files, cryptography, GUI(graphical user interfaces) using Tk, and also other system-dependent stuff. Remember, all this is always available wherever Python is installed. This is called the "batteries included" philosophy of Python.

4.4 Screenshots

The following are the screenshots of the project:

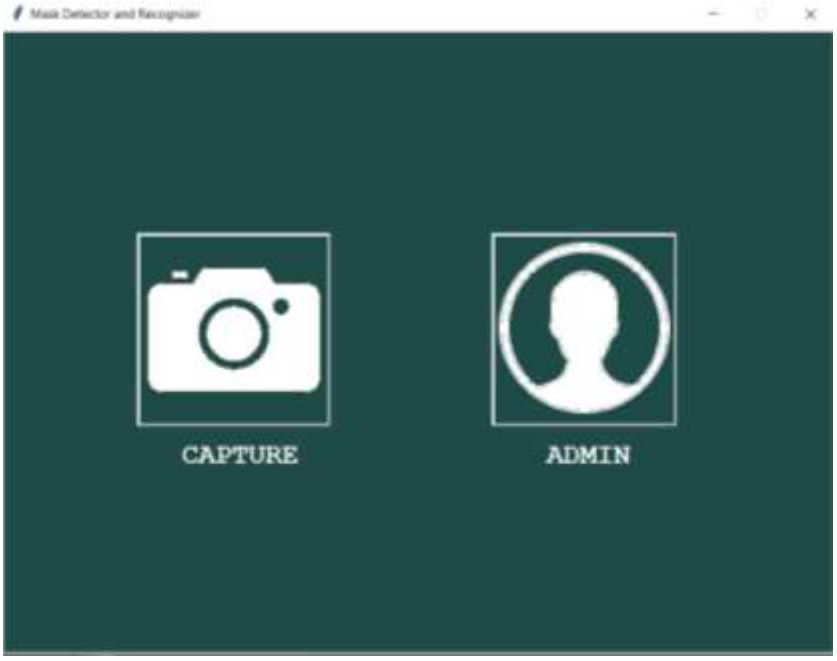


Figure 4-10: Screenshot of gui

When the admin clicks on Admin button, it will ask to enter username and password:

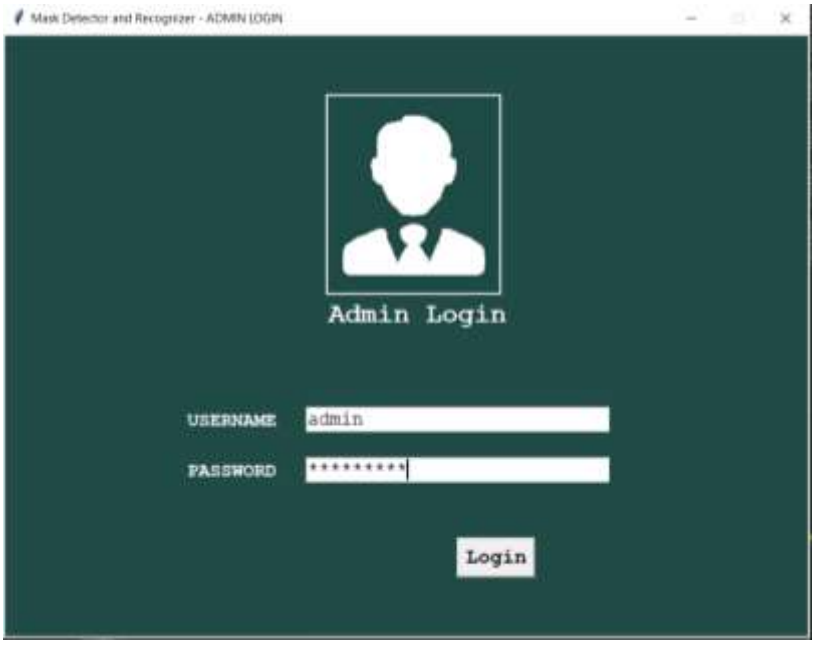


Figure 4-11: Screenshot 2

When the admin enters correct username and password, and clicks on login; then the

following screen opens to see the stored excel file.

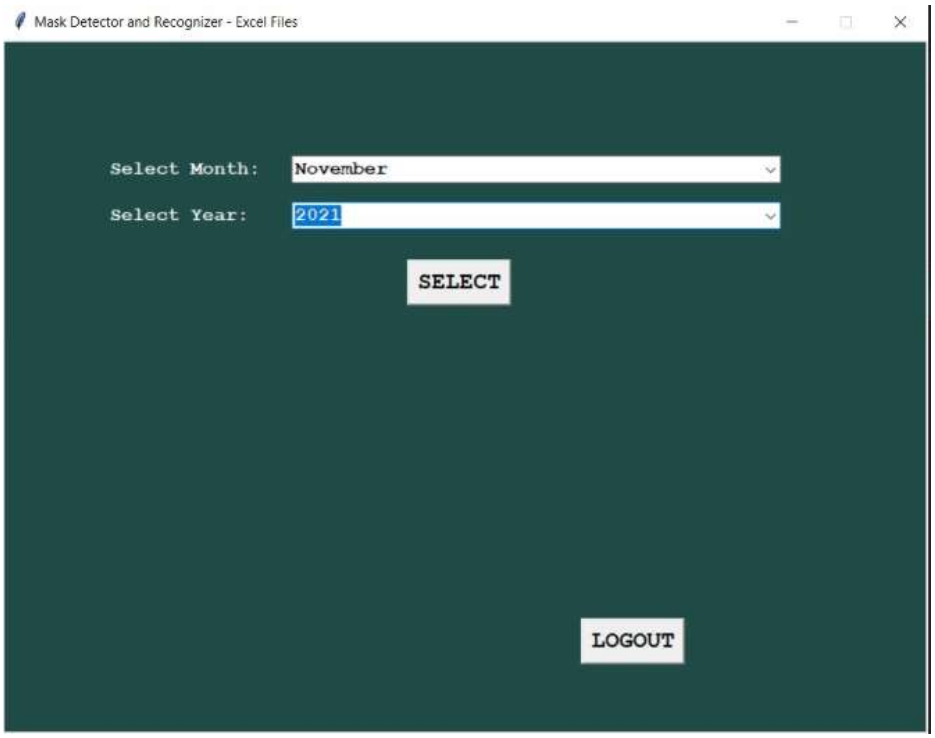


Figure 4-12: Screenshot 3

After admin selects the desired month and year and clicks on SELECT button, then

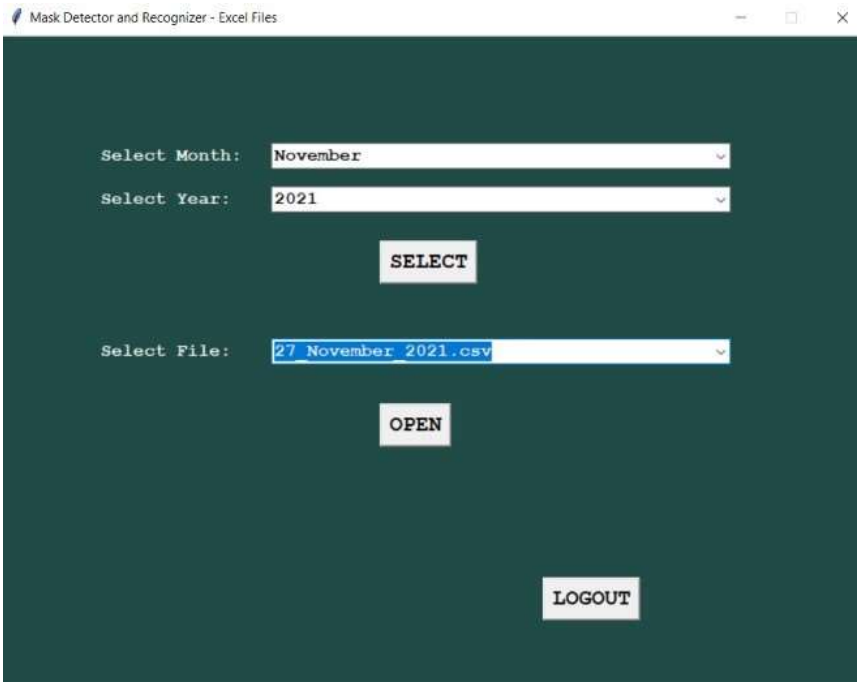


Figure 4-13: Screenshot 4

When the user clicks on Camera button. The following message box-opens with the

instruction for how to close the camera.

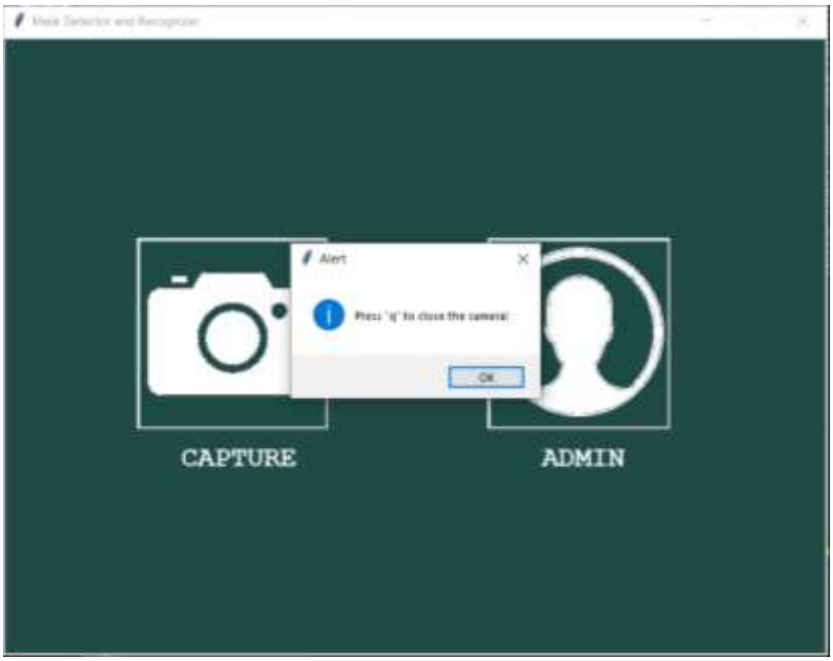


Figure 4-14: Screenshot 5

The Results of the project are:

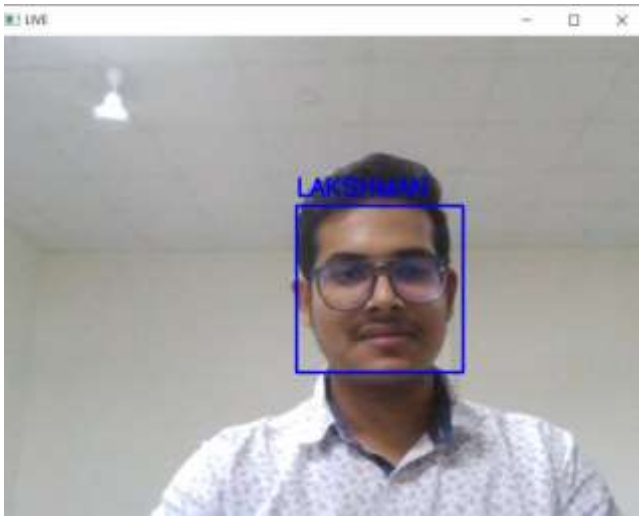


Figure 4-15: Screenshot 6

Figure 4-16: Screenshot 7

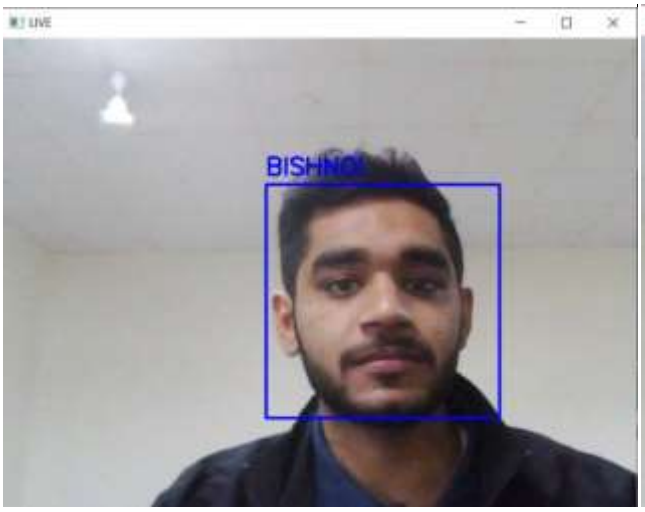


Figure 4-17: Screenshot 8

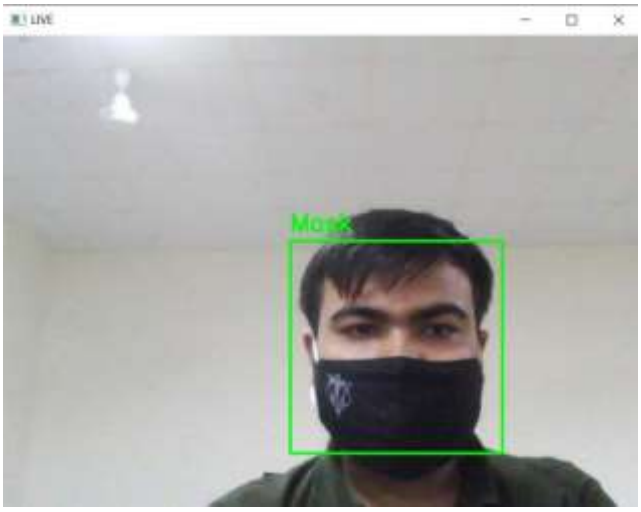


Figure 4-18: Screenshot 9

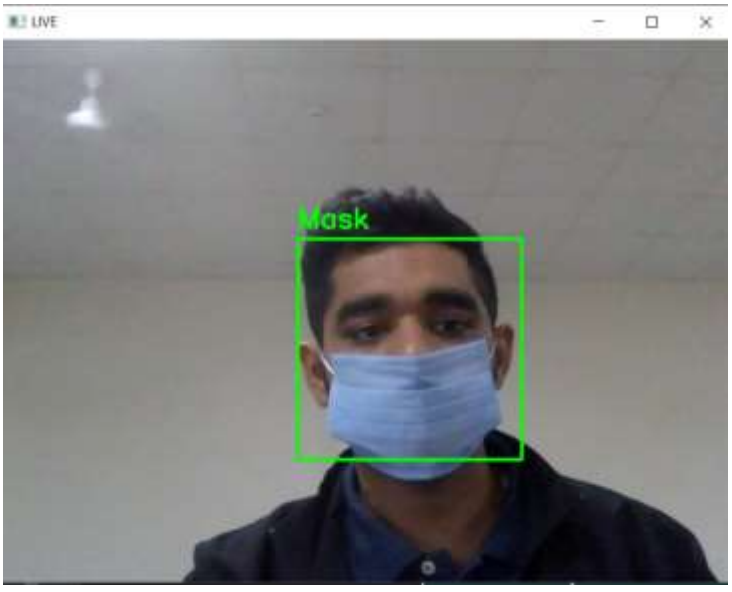


Figure 4-19: Screenshot 10

4.5 Testing

Testing is the process of evaluation of a system to detect differences between given input and expected output and also to assess the feature of the system. Testing assesses the quality of the product. It is a process that is done during the development process.

4.5.1 Strategy Used

Tests can be conducted based on two approaches –

- Functionality testing
- Implementation testing

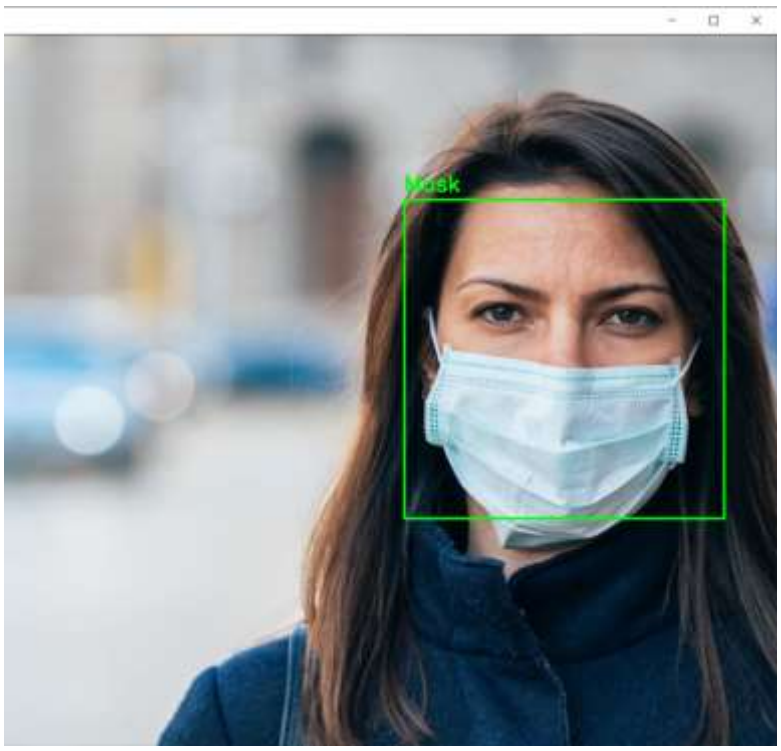
The texting method used here is Black Box Testing. It is carried out to test functionality of the program. It is also called 'Behavioral' testing. The tester in this case, has a set of input values and respective desired results. On providing input, if the output matches with the desired results, the program is tested 'ok', and problematic otherwise.

4.5.2 Test Case and Analysis

TEST CASE: 1

Test Case ID	TC001
Test Case Summary	It will check whether the system detects the face mask or not.
Test Procedure	Place and start the camera at the college entrance.
Expected Result	There should be written 'MASK' on bounding box.
Actual Result	MASK is written on the detected face.
Status	Pass

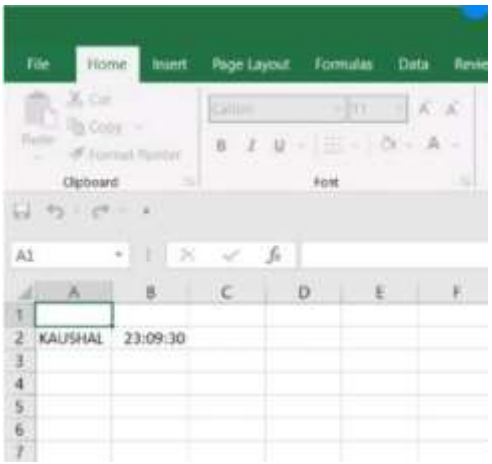
TEST CASE: 1 OUTPUT



TEST CASE: 2

Test Case ID	TC002
Test Case Summary	It will check whether the recognized face without mask is stored in excel sheet or not.
Test Procedure	Place and start the camera at the college entrance. Check excel sheet afterwards.
Expected Result	For each student without mask, there should be entry in excel sheet.
Actual Result	Kaushal Yadav was recognized and his entry was there in excel sheet with time.
Status	Pass

TEST CASE: 2 OUTPUT



Source Code

1. front.py (GUI)

```
import tkinter as tk
from tkinter import *
from tkinter import ttk
import tkinter.messagebox
import os

def login():
    if v_pass.get() == "Admin@123" and v_uname.get()=="admin":
        sheets()
    else:
        tk.messagebox.showinfo('ALERT','Invalid Username or Password!')

def admin():
    m.destroy()
    global mm
    mm = tk.Tk()
    mm.title('Mask Detector and Recognizer - ADMIN LOGIN')
    canvas = Canvas(mm, width = 500, height = 350, bg = '#214B46',relief='raised')
    canvas.pack(expand=YES, fill=BOTH)
    width = mm.winfo_screenwidth()
    height = mm.winfo_screenheight()
    x = int(width / 2 - 800 / 2)
    y = int(height / 2 - 600 / 2)
    str1 = "800x600+" + str(x) + "+" + str(y)
    mm.geometry(str1)
    mm.resizable(width=False, height=False)
    mm.frame = Frame(mm, height=500, width=650, bg = '#214B46',relief='raised')
    mm.frame.place(x=80, y=50)
    x,y = 70,20
    img = PhotoImage(file='admin1.png')
    label = Label(mm, image=img)
    label.place(x=x + 250, y=y + 40)

    label = Label(mm, text="Admin Login",bg='#214B46', fg='white')
    label.config(font=("Courier", 20, 'bold'))
    label.place(x=320, y=y + 240)

    global v_uname
    global v_pass
    v_uname = StringVar()
    v_pass = StringVar()

    emlabel = Label(mm, text="USERNAME",bg='#214B46', fg='white')
    emlabel.config(font=("Courier", 14, 'bold'))
```

```
    emlabel.place(x=180, y=y + 350)

    email = Entry(mm, font='Courier 14', textvariable = v_uname, width = 27)
    email.place(x=300, y=y +350)


    pslabel = Label(mm, text="PASSWORD", bg='#214B46', fg='white')
    pslabel.config(font=("Courier", 14, 'bold'))
    pslabel.place(x=180, y=y + 400)

    password = Entry(mm, show='*', font='Courier 14',textvariable = v_pass, width=27)
    password.place(x=300, y=y + 400)

    button = Button(mm, text="Login", font='Courier 15 bold',command=login)
    button.place(x=450, y=y + 480)

    mm.mainloop()

def onClick():
    tk.messagebox.showinfo('Alert','Press "q" to close the camera!')
    os.system('python d_mask.py')

def sheets():
    mm.destroy()
    global root
    root = tk.Tk()
    root.title('Mask Detector and Recognizer - Excel Files')
    canvas = Canvas(root, width = 500, height = 350, bg = '#214B46',relief='raised')
    canvas.pack(expand=YES, fill=BOTH)
    width = root.winfo_screenwidth()
    height = root.winfo_screenheight()
    x = int(width / 2 - 800 / 2)
    y = int(height / 2 - 600 / 2)
    str1 = "800x600+" + str(x) + "+" + str(y)
    root.geometry(str1)
    root.resizable(width=False, height=False)
    root.frame = Frame(root, height=500, width=650, bg = '#214B46',relief='raised')
    root.frame.place(x=80, y=50)
    x,y = 70,20
    label = Label(root,text = "Select Month:",bg='#214B46', fg='white')
    label.config(font=("Courier", 12, 'bold'))
    label.place(x=x+20, y=y +80)
    global month
    month = StringVar()
    monthchoosen = ttk.Combobox(root, width = 40, textvariable=month, font=("Courier", 12, 'bold'))
    monthchoosen['values'] = ('January',
                              'February',
                              'March',
                              'April',
                              'May',
                              'June',
```

```
                    'July',
                    'August',
                    'September',
                    'October',
                    'November',
                    'December')
    monthchoosen.place(x = x+180, y = y+80)
    monthchoosen.current()

    label = Label(root,text = "Select Year:",bg='#214B46', fg='white')
    label.config(font=("Courier", 12, 'bold'))
    label.place(x=x+20, y=y +120)
    global year
    year = StringVar()
    yearchoosen = ttk.Combobox(root, width = 40, textvariable=year, font=("Courier", 12, 'bold'))
    yearchoosen['values'] = ('2021','2020')
    yearchoosen.place(x = x+180, y = y+120)
    yearchoosen.current()

    button = Button(root, text="SELECT", font='Courier 15 bold',command = show_sheets)
    button.place(x=350, y=y + 170)

    button = Button(root, text="LOGOUT", font='Courier 15 bold', command = logout)
    button.place(x=500, y=y + 480)

    root.mainloop()

def show_sheets():
    m = month.get()
    ye = year.get()
    x,y = 70,20

    global fil
    fil = StringVar()
    filechosen = ttk.Combobox(root, width=40,textvariable=fil, font=("Courier", 12, 'bold'))
    mylist = os.listdir('Fine')
    files = []
    count = 0
    for i in mylist:
        s = i.split('_')
        if s[1]==m and s[2]==ye+'.csv':
            files.append(i)
            count += 1
    if count>0:
        label = Label(root,text = "Select File:",bg='#214B46', fg='white')
        label.config(font=("Courier", 12, 'bold'))
        label.place(x=x+20, y=y +260)
        filechosen['values'] = tuple(files)
        filechosen.place(x=x+180,y=y+260)
        filechosen.current()
```

```
        button = Button(root, text="OPEN", font='Courier 15 bold', command = open_file)
        button.place(x=350, y=y + 320)
    else:
        tk.messagebox.showinfo('ALERT','No Files found for the entered month and year!\nTRY AGAIN!')

def open_file():
    name = fil.get()
    name = name.strip()
    os.system('start excel D:\\PROJECT\\Fine\\'+name)

def logout():
    root.destroy()
    first()

def first():
    global m
    m = tk.Tk()
    m.title('Mask Detector and Recognizer')
    canvas = Canvas(m, width = 500, height = 350, bg = '#214B46',relief='raised')
    canvas.pack(expand=YES, fill=BOTH)
    width = m.winfo_screenwidth()
    height = m.winfo_screenheight()
    x = int(width / 2 - 800 / 2)
    y = int(height / 2 - 600 / 2)
    str1 = "800x600+" + str(x) + "+" + str(y)
    m.geometry(str1)
    m.resizable(width=False, height=False)
    m.frame = Frame(m, height=500, width=650, bg = '#214B46',relief='raised')
    m.frame.place(x=80, y=50)
    x,y = 70,20

    img = PhotoImage(file='camera.png')
    label = Button(m, image=img, command=onClick)
    label.place(x=x + 60, y=y + 175)

    label = Label(m, text="CAPTURE",bg='#214B46', fg='white')
    label.config(font=("Courier", 20, 'bold'))
    label.place(x=x+100, y=y +370)

    img1 = PhotoImage(file='admin.png')
    label = Button(m,image=img1,command = admin)
    label.place(x=x+400,y=y+175)

    label = Label(m, text="ADMIN",bg='#214B46', fg='white')
    label.config(font=("Courier", 20, 'bold'))
    label.place(x=x+450, y=y +370)
    m.mainloop()

first()
```

2. mask_detect_train.py

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.layers import AveragePooling2D
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Input
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.utils import to_categorical
from sklearn.preprocessing import LabelBinarizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from imutils import paths
import matplotlib.pyplot as plt
import numpy as np
import os

INIT_LR = 1e-4
EPOCHS = 20
BS = 32

DIRECTORY = r"D:\PROJECT\dataset"
CATEGORIES = ["with_mask", "without_mask"]

#PREPROCESSING
data = []
labels = []

for category in CATEGORIES:
    path = os.path.join(DIRECTORY, category)
    for img in os.listdir(path):
        img_path = os.path.join(path, img)
        image = load_img(img_path, target_size=(224, 224))
        image = img_to_array(image)
        image = preprocess_input(image)

        data.append(image)
        labels.append(category)
```

```
lb = LabelBinarizer()
labels = lb.fit_transform(labels)
labels = to_categorical(labels)

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

(trainX, testX, trainY, testY) = train_test_split(data, labels,
   test_size=0.20, stratify=labels, random_state=42)

aug = ImageDataGenerator(
      rotation_range=20,
      zoom_range=0.15,
      width_shift_range=0.2,
      height_shift_range=0.2,
      shear_range=0.15,
      horizontal_flip=True,
      fill_mode="nearest")
```

#TRAINING MOBILENET NETWORK

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

headModel = baseModel.output
headModel = AveragePooling2D(pool_size=(7, 7))(headModel)
headModel = Flatten(name="flatten")(headModel)
headModel = Dense(128, activation="relu")(headModel)
headModel = Dropout(0.5)(headModel)
headModel = Dense(2, activation="softmax")(headModel)

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

for layer in baseModel.layers:
   layer.trainable = False

print("[INFO] compiling model...")
opt = Adam(lr=INIT_LR, decay=INIT_LR / EPOCHS)
model.compile(loss="binary_crossentropy", optimizer=opt,
   metrics=["accuracy"])

print("[INFO] training head...")
H = model.fit(
   aug.flow(trainX, trainY, batch_size=BS),
   steps_per_epoch=len(trainX) // BS,
   validation_data=(testX, testY),
   validation_steps=len(testX) // BS,
   epochs=EPOCHS)
```

```
print("[INFO] evaluating network...")
predIdxs = model.predict(testX, batch_size=BS)
predIdxs = np.argmax(predIdxs, axis=1)

print(classification_report(testY.argmax(axis=1), predIdxs,
   target_names=lb.classes_))

print("[INFO] saving mask detector model...")
model.save("mask_detector.model", save_format="h5")
```

```
#PLOT GRAPH

N = EPOCHS
plt.style.use("ggplot")
plt.figure()
plt.plot(np.arange(0, N), H.history["loss"], label="train_loss")
plt.plot(np.arange(0, N), H.history["val_loss"], label="val_loss")
plt.plot(np.arange(0, N), H.history["accuracy"], label="train_acc")
plt.plot(np.arange(0, N), H.history["val_accuracy"], label="val_acc")
plt.title("Training Loss and Accuracy")
plt.xlabel("Epoch #")
plt.ylabel("Loss/Accuracy")
plt.legend(loc="lower left")
plt.savefig("plot.png")
```

3. mask_face_recognition.py

```
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
import numpy as np
import imutils
import time
from cv2 import cv2
import os
from datetime import datetime
import face_recognition
import csv

path = 'Face_Dataset'
images = []
classNames = []
myList = os.listdir(path)
for cl in myList:
   curImg = cv2.imread(f'{path}/{cl}')
   images.append(curImg)
   classNames.append(os.path.splitext(cl)[0])
```

```
def findEncodings(images):
    encodeList = []
    for img in images:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
        encodeList.append(encode)
    return encodeList



def markName(name):

    filename = datetime.now()
    x = filename.strftime("%d_%B_%Y")+'.csv'
    p = "D:\\PROJECT\\Fine\\"+x
    if os.path.exists(p):
        with open(p, 'r+') as f:
            myDataList = f.readlines()
            nameList = []
            for line in myDataList:
                entry = line.split(',')
                nameList.append(entry[0])
            if name not in nameList and name !='Unknown' and name != 'Mask':
                now = datetime.now()
                dtString = now.strftime('%H:%M:%S')
                f.writelines(f'\n{name},{dtString}')

    else:
        with open(p, 'w') as f:

            nameList = []
            if name not in nameList and name!='Unknown' and name != 'Mask':
                now = datetime.now()
                dtString = now.strftime('%H:%M:%S')
                f.writelines(f'\n{name},{dtString}')
encodeListKnown = findEncodings(images)
print('Encoding Complete')



faceCascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
model = load_model("mask_detector.model")

video_capture = cv2.VideoCapture(0)
while True:
    # Capture frame-by-frame
    ret, frame = video_capture.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(gray,
```

```
                        scaleFactor=1.1,
                        minNeighbors=5,
                        minSize=(60, 60),
                        flags=cv2.CASCADE_SCALE_IMAGE)
faces_list=[]
preds=[]

facesCurFrame = face_recognition.face_locations(frame)
encodesCurFrame = face_recognition.face_encodings(frame, facesCurFrame)
faces_name = []

for encodeFace in encodesCurFrame:
    matches = face_recognition.compare_faces(encodeListKnown, encodeFace)
    name = 'Unknown'
    faceDis = face_recognition.face_distance(encodeListKnown, encodeFace)
    matchIndex = np.argmin(faceDis)

    if matches[matchIndex]:
        name = classNames[matchIndex].upper()

    faces_name.append(name)

for (x, y, w, h) in faces:
    face_frame = frame[y:y+h,x:x+w]
    face_frame = cv2.cvtColor(face_frame, cv2.COLOR_BGR2RGB)
    face_frame = cv2.resize(face_frame, (224, 224))
    face_frame = img_to_array(face_frame)
    face_frame = np.expand_dims(face_frame, axis=0)
    face_frame =  preprocess_input(face_frame)
    faces_list.append(face_frame)

    if len(faces_list)>0:
        preds = model.predict(faces_list)

    for pred in preds:
        (mask, withoutMask) = pred

    if mask > withoutMask:
        label = "Mask"
    else:
        for (top, right, bottom, left), name in zip(facesCurFrame, faces_name):
            label = name
        markName(label)
    color = (0, 255, 0) if label == "Mask" else (0, 0, 255)

    cv2.putText(frame, label, (x, y- 10),
                cv2.FONT_HERSHEY_SIMPLEX, 0.45, color, 2)

    cv2.rectangle(frame, (x, y), (x + w, y + h),color, 2)
    # Display the resulting frame
```

```
    cv2.imshow("LIVE", frame)
    key = cv2.waitKey(1) & 0xFF

    # if the `q` key was pressed, break from the loop
    if key == ord("q"):
        break

video_capture.release()
cv2.destroyAllWindows()
```

Chapter 5.

Conclusion

5.1 Conclusion

The aim of the project that was to automatically detect mask and recognize the name of person without mask and save their name and time at which they were recognized in an excel sheet and are accurately done by this project with the use of concepts like Deep learning, MobileNet Convolutional Neural Networks, OpenCV and TensorFlow.

The work done manually can now be completely replaced by this automated system and it can reduce all the extra efforts of maintain the records.

5.2 Limitations of the Work

- The working of this project would be a little slow because framework like TensorFlow, & deep learning need high-processing hardware and GPU (graphical processing unit) systems but we are using CPU only.
- The models that we are using for identifying the objects are pre- trained models. So, if we want to train our own model, it takes a lot of time and processing.
- In the system, scanning of each frame is one per second but still it needs improvement. If the objects move too fast, it may not detect them.
- The faces not visible clearly cannot be recognized.
- Currently, it can be used for a less face dataset, and also dataset is to be managed by admin

5.3 Suggestion and Recommendations for Future Work

- The Model would be trained for detecting a greater number of objects and also recognizing faces.
- SNS service will be integrated in this project for alert notification when an unwanted object is detected.
- Currently, the bounding box technique is used which is bounding the targeted object within a rectangle. In future, Segmentation will be used.
- Currently, it can be used where the face dataset is already available with admin.

Bibliography

[1] Raza Ali, Saniya Adeel, Akhyar Ahmed, Md Hasan Shahriar, Md Shohel Mojumder, Dr. Christoph Lipper t, "Face Mask Detector" in July 2020

[2] Hong Zhao, Xi-Jun Liang, and Peng Yang, "Research on Face Recognition Based on Embedded System"

In Hindawi Publishing Corporation in 2013

[3] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman and A. Blake, "Real-Time Human Pose Recognition in Parts from Single Depth Images", Computer Vision and Pattern Recognition, 2011

[4] Marco Grassi, Marcos Faundez-Zanuy, "Face Recognition with Facial Mask Application and Neural Network", Computational and Ambient Intelligence, 2007

[5] Naveen Kumar Mahamkali, Vadivel Ayyasamy, "OpenCV for Computer Vision Applications" Research Gate, march 2015.