LAB REPORT

Submitted by

Ashish Prakash Singh [RA2011033010060]
Nilay [RA2011033010056]
Karan Keshri [RA2011033010044]

Under the Guidance of

Dr.B.Jothi

Lab Incharge, Cintel CSE SWE

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING

with specialization in Software

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

JUNE 2022

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

SRM INSTITUTION OF SCIENCE AND TECHNOLOGY KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this lab report titled Face Mask Detection is the bonafide work done by Ashish PrakashSingh(RA2011033010060),Nilay(RA2011033010056),KaranKeshri(RA2011033010044) who carried out the lab exercises under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

SIGNATURE
Dr.B.Jothi
SEPM – Lab Incharge

DEPARTMENT OF CINTEL

ABSTRACT

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

This project presents a simplified approach to achieve this purpose using some basic Machine Learning packages like TensorFlow, Keras, OpenCV and Scikit-Learn. The proposed method detects the face from the image correctly and then identifies if it has a mask on it or not. As a surveillance task performer, it can also detect a face along with a mask in motion. The method attains accuracy up to 95.77% and 94.58% respectively on two different datasets. We explore optimized values of parameters using the Sequential Convolutional Neural Network model to detect the presence of masks correctly without causing over-fitting.

18CSC206J

Software Engineering and Project Management Project

Face Mask Detection

CSE specialization in Software

Ashish Prakash Singh - RA2011033010060

Nilay - RA2011033010056

Karan Keshri - RA2011033010044

Index/Table of Contents

LIST OF FIGURE

LIST OF ABBREVIATIONS

YOLO – You Only Live Once

ALGO – Algorithm

SDLC – System Development Life Cycle

RIO – Region of Interest

OS – Operating System

KLOC – Thousands of Lines of Code

SWOT – Strength Weakness Opportunity Threats

CNN – Convolutional Neural Network

ER – Entity Relation

ISA – Instruction Set Architecture

DFD – Data Flow Diagram

UML – Unified Modelling Language

CV – Computer Vision

Problem Statement

The main objective of the face detection model is to detect the face of individuals and conclude whether they are wearing masks or not at that particular moment when they are captured in the image or in Video.

With the advancements in technology that the world has been witnessing, there are various available techniques that could prove valuable to society if used effectively. A real-time system which could itself classify, seeing a person, in two categories :

1. A person wearing a face mask
2. A person not wearing a face mask

could be useful in recent times. Such systems could find applications in public areas like hospitals, airports, malls, etc. One of the methods to make the detector is by first detecting the faces in real-time. And, after detecting the faces from the webcam stream, saving the frames containing the faces and next applying a classifier. The numerous algorithms that could be used for categorization have been discussed in the subsequent section. Another way that could be opted to execute the same is by using an object detection model. Following are the contributions in view of the current state-of-the-art.

- Although several precautions are recommended to get safe from covid-19, still face masking, and social distancing are significant factors. So, it was necessary to propose many face masking techniques under one umbrella for the research community.
- Pertaining to the need of the current time, the proposed work reviews several studies conducted in the field of face mask detection. The strong suit of plenty of publications has been discussed on face masking, which is still missing in terms of observations, future trends, a vast number of references, current trends, etc.

- Performance parameters of several algorithms are compared, and discussions on them are presented to increase the efficacy of the review paper.

With time, the surge in COVID-19 cases urged people to be cautious, alert, and take all safety measures possible. In situations such as this, where a mere sneeze could be harmful to many people, safety remains the priority. To ensure the well-being of all humans, a system that could itself monitor if a face mask is on or not is necessitated. It would not only secure a being rather fellows in the vicinity as well. Having access to the ultra-modern technological methods, implementing such a system could be a boon to society.

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	1
Title of Experiment	Business Case Template
Name of the candidate	Ashish Prakash Singh
Team Members	Karan Keshri, Nilay
Register Number	RA2011033010060
Date of Experiment	09/03/2022

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

Aim

To create business case template

Team Members:

S No	Register No	Name	Role
1	RA2011033010060	Ashish Prakash Singh	**Rep**
2	RA2011033010044	Karan Keshri	**Member**
3	RA2011033010056	Nilay	**Member**

Result:

Business case template was created successfully

ONE PAGE BUSINESS CASE TEMPLATE

DATE	09/03/2022
SUBMITTED BY	ASHISH, NILAY,KARAN
TITLE/ROLE	FACE MASK DETECTION

THE PROJECT

- The aim of our project is to develop a software for face mask detection.
- We are going to use python programming language and machine learning concept for detecting face mask.
- This software will be implemented in camera on entry point.
- This will be a real time project.

HISTORY

In this pandemic we saw that health is a first priority for all of us. As we all know WHO clearly instructed us to wear face mask for protecting us from this virus. As cases decreasing people became unaware of using mask but again virus is coming with new mutation, So we choose this project to find where people are using face mask or not.

Many people ignore wearing face mask so if we implement this software on the entry point this will only allow those people to enter inside building who wear face mask.

Limitations

- Involve too many complexities and machine learning concepts.
- Lack of knowledge in python programming and machine learning.
- Sometimes it is unable to detect face properly due to camera quality.

Benefits

- This will help many organization in detecting whether people are wearing face mask or not.
- No need of any other person of checking mask.
- Save manual labour.

Stack holder

- This project will be funded by either government or private company, those who want to implement this software

Requirements

- Camera (1080p,720p)
- Python 3
- YOLO algorithm

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	2
Title of Experiment	Stakeholder and Process Model
Name of the candidate	Ashish Prakash Singh
Team Members	Karan Keshri, Nilay
Register Number	RA2011033010060
Date of Experiment	23/03/2022

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		**10**	

Staff Signature with date

Aim

To create stakeholder and process model

Team Members:

S No	Register No	Name	Role
1	RA2011033010060	Ashish Prakash Singh	**Rep**
2	RA2011033010044	Karan Keshri	**Member**
3	RA2011033010056	Nilay	**Member**

Result:

Stakeholder and process model was created successfully.

STAKEHOLDER AND PROCESS MODEL

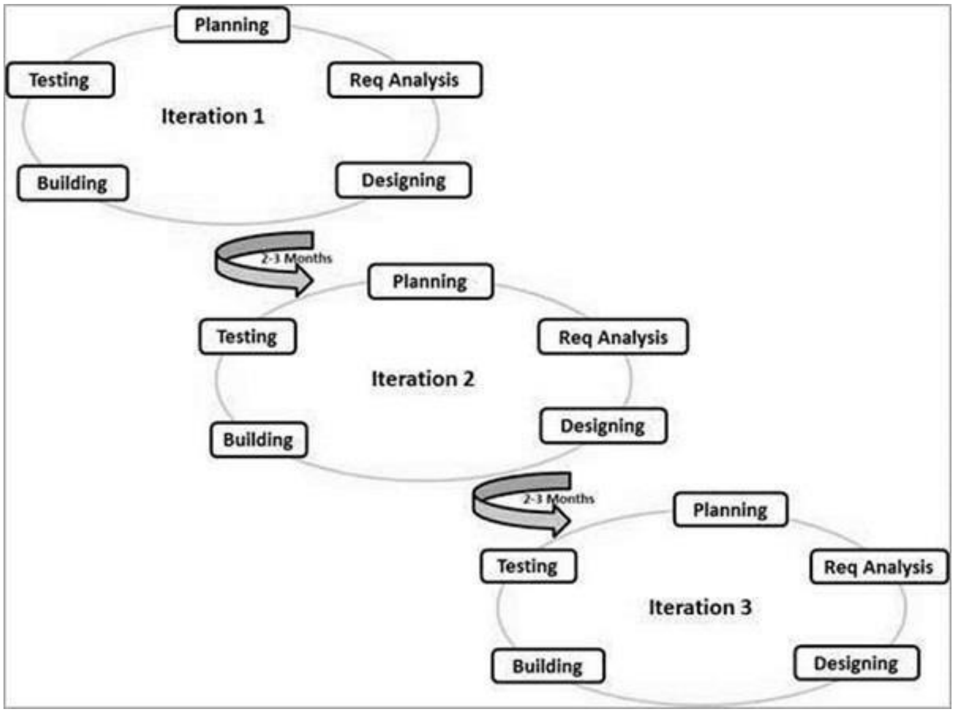
<table>
<tr><td colspan="5">PROJECT NAME : Face Mask Detection</td></tr>
<tr><td colspan="5">Prepared by : Ashish Prakash Singh, Karan Keshri , Nilay</td></tr>
<tr><td colspan="5">Date : 23-March-2022</td></tr>
<tr><th>PROJECT STAKEHOLDER NAME</th><th>SPECIFIC INFORMATION NEEDS</th><th>PROJECTS INTERESTS</th><th>IMPACT ON PROJECT</th><th>ROLE</th></tr>
<tr><td>MS.B.JYOTI</td><td>Frequent communication, provides instructions and guidance</td><td>Decision making , leader</td><td>Influencer</td><td>Directorial, Management</td></tr>
<tr><td>ASHISH PRAKASH SINGH</td><td>Managing project progress</td><td>Technical Project Manager, team leader</td><td>Supporter</td><td>Decision Making</td></tr>
<tr><td>Nilay</td><td>Checking the Demands</td><td>Co-Operating with team members</td><td>Supporter</td><td>Consultant</td></tr>
<tr><td>Karan Keshri</td><td>Work on assigned task</td><td>Co-Operating with team members</td><td>Supporter</td><td>Decision Making</td></tr>
<tr><td>Canon Camera Company</td><td>Investor/sponsor</td><td>Sales Increment</td><td>Financial Supporter</td><td>Collaborator</td></tr>
<tr><td>Srm University</td><td>Customer</td><td>Safety</td><td>Positive</td><td>Participant</td></tr>
<tr><td>Government office</td><td>Customer</td><td>Safety</td><td>positive</td><td>participant</td></tr>
</table>

Project model

We are going to use Agile model for our project implementation.

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations.

Diagram



Reason

Like in this model, every iteration has to be complete like that in iteration(lab section) we also follow all the steps.

1. Planning- In this phase first we decide that what we have to mean how much of our project work we have to done. According to it we plan our activity and we try to complete.
2. Requirement Analysis- In this phase we decide what is our requirement for this iteration.
3. Designing- In this phase we design our work according to task assigned our project coordinator Mrs Jyoti ma’am.
4. Building – After completing all early phase we work on our assigned work and complete it.

5. Testing- In Testing phase we complete our work and it going to be test by our project coordinator and after testing she gives feedback and approve or iteration.

After completion of all the iteration of project it will deployed to our customer and for maintenance We will give update for two year . Pros –

1. It promotes teamwork and cross training
2. Easy to manage.
3. Highly Flexible.
4. Little planning required.

Cons –

1. Limited documentation
2. No finite end

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	3
Title of Experiment	Software Requirements Specifications
Name of the candidate	Ashish Prakash Singh
Team Members	Karan Keshri, Nilay
Register Number	RA2011033010060
Date of Experiment	30/03/2022

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

Aim

To add software requirements specification

Team Members:

S No	Register No	Name	Role
1	RA2011033010060	Ashish Prakash Singh	**Rep**
2	RA2011033010044	Karan Keshri	**Member**
3	RA2011033010056	Nilay	**Member**

Result:

Software requirements spectification was added successfully.

REQUIREMENTS

1. SYSTEM REQUIREMENTS

- 4GB RAM for reach Core
- 4Core3GHzEach
- Standard hard drive
- Network connectivity
- Disk storage
- UPS
- Internet connectivity

2. USER REQUIREMENTS

- Camera with 1080p or 720p quality
- Windows OS
- Keyboard and mouse
- Google chrome latest version
- Github account
- Python

3. FUNCTIONAL REQUIREMENTS

- Provides face mask detection feature which can be used with camera
- System will detect face mask of a person through camera and if a person is not wearing a mask, it will give a warning message.
- The system must be correctly able to load the face mask classifier model.
- The system must be able to detect faces in images or video stream.
- The system must be able to extract each face's Region of Interest (ROI)
- Example- If a person is wearing mask below the chin, then system will show less percentage than required to pass the face mask detection test.

4. NON–FUNCTIONAL REQUIREMENTS

- The face should be localized by detecting the facial landmarks and the background must be ignored.
- The system will be implemented in Python script with an accuracy of the model of over 90%.
- The user must not move his/her face out of camera's sight in order to get correct results.

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

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	4
Title of Experiment	Cost Estimation and Work break down structure
Name of the candidate	Ashish Prakash Singh
Team Members	Karan Keshri, Nilay
Register Number	RA2011033010060
Date of Experiment	06/04/2022

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

Aim

To create cost estimation and work breakdown structure

Team Members:

S No	Register No	Name	Role
1	RA2011033010060	Ashish Prakash Singh	**Rep**
2	RA2011033010044	Karan Keshri	**Member**
3	RA2011033010056	Nilay	**Member**

Result:

cost estimation and work breakdown was created successfully.

4. Cost Estimation using COCOMO Model

4.1 Modules:

● Login Module — 20 KLOC

● Home Module — 50 KLOC

● Detection Module — 30KLOC

● Feedback Module — 20 KLOC

□ Calculation Module — 30 KLOC

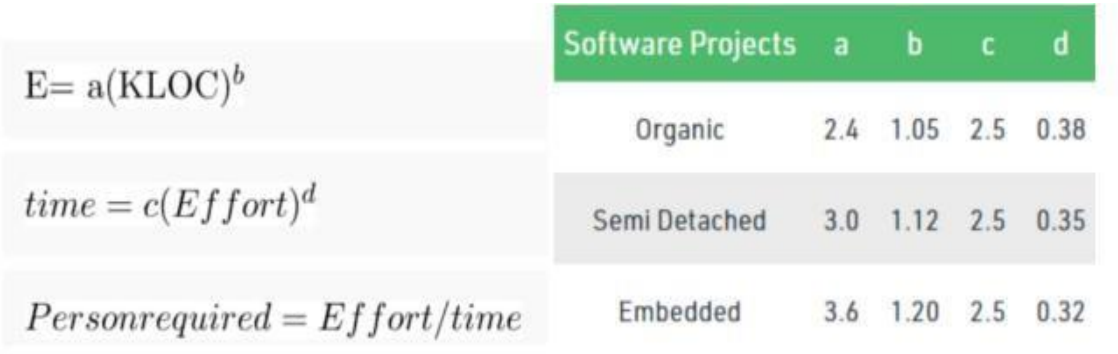
Total — 150KLOC

4.2 Organic

If the team size is small enough, the problem is well understood and has been solved before, and the team members have some expertise with the problem, the software project is said to be organic.

4.3 Estimation of Efforts: Calculations:

Basic-Model:



$$E = a(KLOC)^b$$

$$time = c(Effort)^d$$

$$Personrequired = Effort/time$$

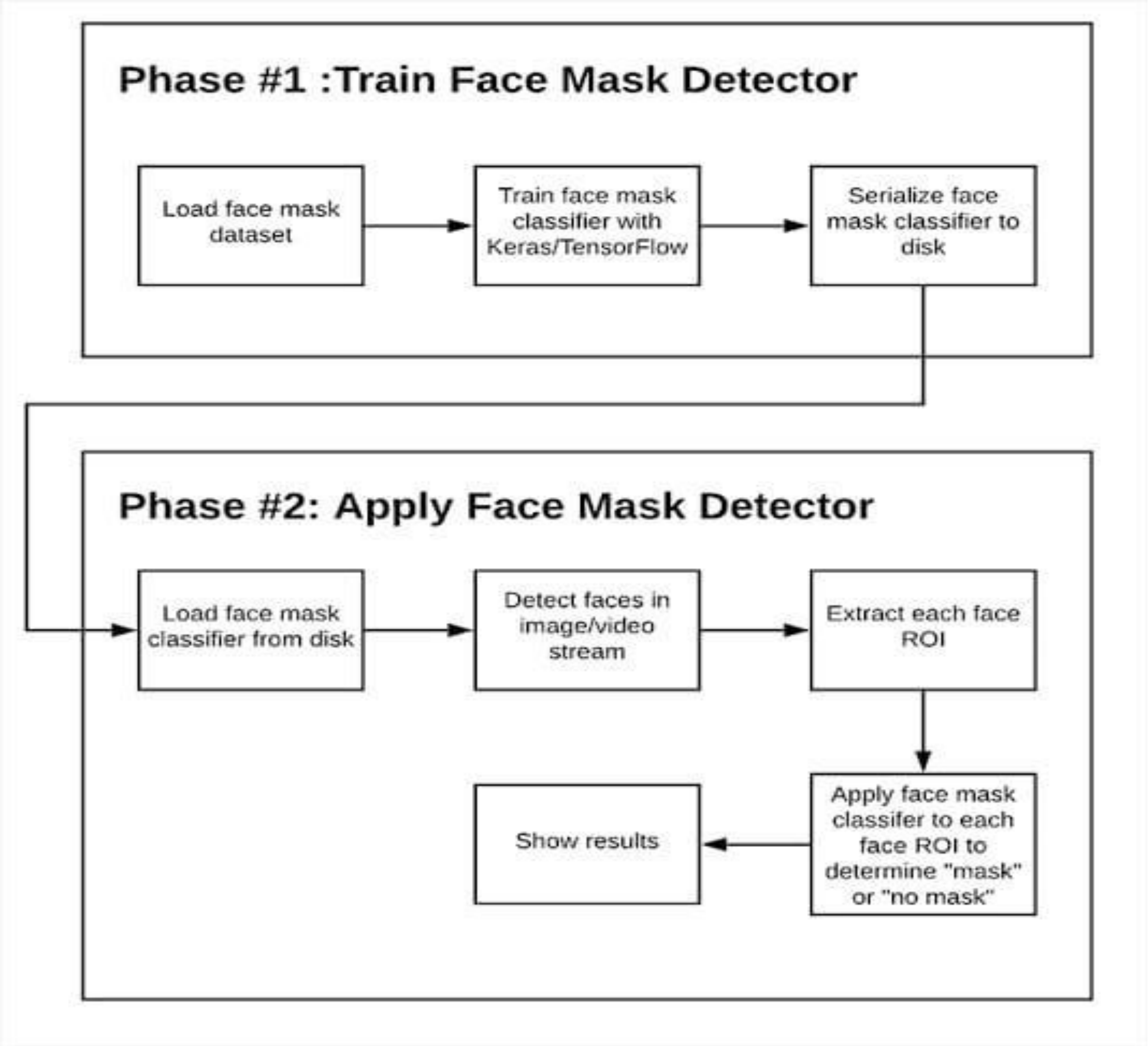
Software Projects	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

E = 2.4*(150)^1.05 = 2.4*192.706 = 462.495MM

Time = 2.5*(462.495)^0.38 = 2.5 * 10.297= 25.74~ 26 Months

Person required = E/Time = 462.495/26 = 17.78~ 18 Persons

4.4 Work breakdown structure


Phase #1 :Train Face Mask Detector
Load face mask dataset
Train face mask classifier with Keras/TensorFlow
Serialize face mask classifier to disk
Phase #2: Apply Face Mask Detector
Load face mask classifier from disk
Detect faces in image/video stream
Extract each face ROI
Apply face mask classifer to each face ROI to determine "mask" or "no mask"
Show results

4.5 Elicitation Technique-

For the Survey, a set of questions is given to stakeholders to quantify their thoughts. After collecting the responses from stakeholders, data is analysed to identify the area of interest of stakeholders.

Questions should be based on high-priority risks. Questions should be direct and unambiguous. Once the survey is ready, notify the participants and remind them to participate.

Two types of questions can be used here:

- **Open-Ended:** Respondent is given the freedom to provide answers in their own words rather than selecting from predefined responses. This is useful but at the same time, this is timeconsuming as interpreting the responses is difficult.

- **Close Ended:** It includes a predefined set of answers for all the questions and the respondent has to choose from those answers. Questions can be multiple choice or can be ranked from not important to very important.

Benefits:

- Easy to get data from a large audience.
- Less time is required for the participants to respond.
- You can get more accurate information as compared to interviews.

Drawback:

- All the Stakeholders might not participate in the surveys.
- Questions may not be clear to all the participants.
- Open-ended questions require more analysis.
- Follow up surveys might be required based on the responses provided by participants.

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	5
Title of Experiment	Ṭimeline Scheduling and Gantt chart, Risk Analysis
Name of the candidate	Ashish Prakash Singh
Team Members	Karan Keshri, Nilay
Register Number	RA2011033010060
Date of Experiment	13/04/2022

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

Aim

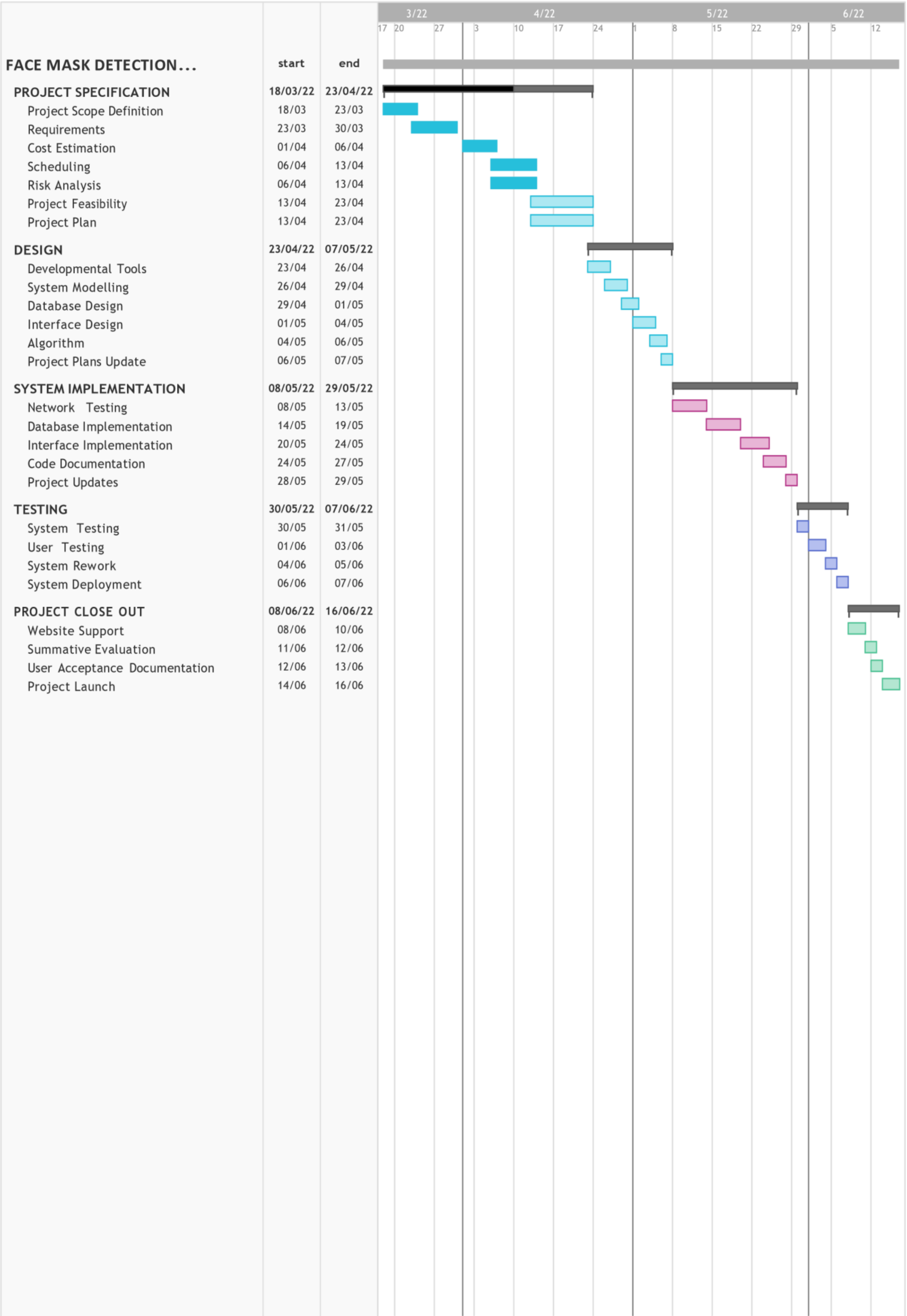
To create time scheduling, gantt chart, risk analysis

Team Members:

S No	Register No	Name	Role
1	RA2011033010060	Ashish Prakash Singh	**Rep**
2	RA2011033010044	Karan Keshri	**Member**
3	RA2011033010056	Nilay	**Member**

Result:

Time scheduling, gantt chart, risk analysis model was created successfully.



FACE MASK DETECTION...	start	end
PROJECT SPECIFICATION	**18/03/22**	**23/04/22**
Project Scope Definition	18/03	23/03
Requirements	23/03	30/03
Cost Estimation	01/04	06/04
Scheduling	06/04	13/04
Risk Analysis	06/04	13/04
Project Feasibility	13/04	23/04
Project Plan	13/04	23/04
DESIGN	**23/04/22**	**07/05/22**
Developmental Tools	23/04	26/04
System Modelling	26/04	29/04
Database Design	29/04	01/05
Interface Design	01/05	04/05
Algorithm	04/05	06/05
Project Plans Update	06/05	07/05
SYSTEM IMPLEMENTATION	**08/05/22**	**29/05/22**
Network Testing	08/05	13/05
Database Implementation	14/05	19/05
Interface Implementation	20/05	24/05
Code Documentation	24/05	27/05
Project Updates	28/05	29/05
TESTING	**30/05/22**	**07/06/22**
System Testing	30/05	31/05
User Testing	01/06	03/06
System Rework	04/06	05/06
System Deployment	06/06	07/06
PROJECT CLOSE OUT	**08/06/22**	**16/06/22**
Website Support	08/06	10/06
Summative Evaluation	11/06	12/06
User Acceptance Documentation	12/06	13/06
Project Launch	14/06	16/06

Time Scheduling

Task	**Start Date`**	**End Date**
Business Case	18th March 2022	23rd March 2022
Stakeholder and User Description	23rd March 2022	31st March 2022
Identify the Requirements	31st March 2022	6th April 2022
Cost Estimation and WBS	6th April 2022	13th April 2022
Scheduling Gantt Chart and Risk Analysis	13th April 2022	20th April 2022
Design system architecture and UML diagrams	20th April 2022	4th May 2022

Module implementation	4th May 2022	18th May 2022e
Testing	18th May 2022	2nd June 2022
Demo	8th June 2022	16th June 2022

Risk management

Risk from Outer world

- Inappropriate Distance between face and camera.
- Dim lightning on the face will create problems on case recognition and percentage calculation .
- The lower camera quality or dull lens of the camera .

Server end

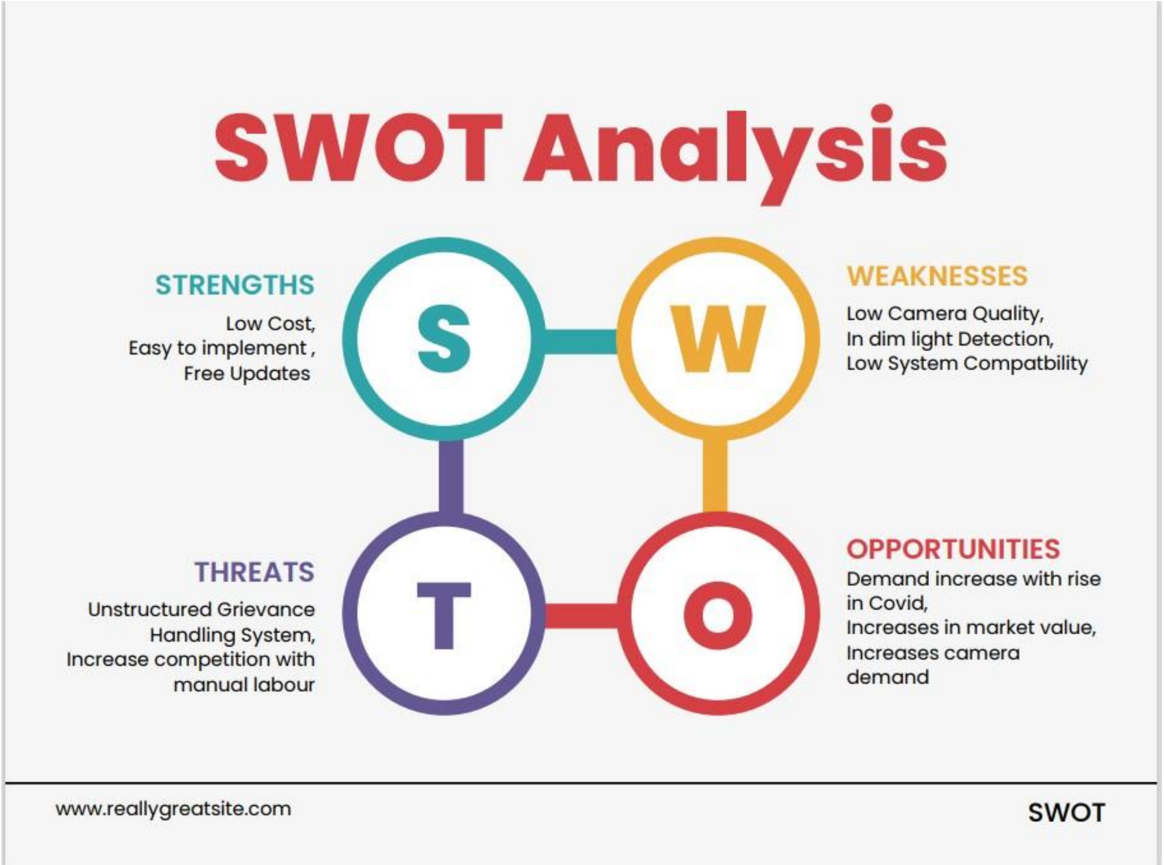
- Failure in data management.
- Failure to calculate percentage graph.
- Failure in connecting with the server.
- Failure in the coding part.

Solution

- Distance between face and camera must be appropriate.
- There must be proper lighting on the face so the face should be clearly visible.
- The camera quality must be high(1080px,720px).

- Proper maintenance of code and review it properly.
- Be precise with percentage calculation.

Swot Analysis



SWOT Analysis

STRENGTHS

Low Cost,
Easy to implement ,
Free Updates

WEAKNESSES

Low Camera Quality,
In dim light Detection,
Low System Compatbility

THREATS

Unstructured Grievance
Handling System,
Increase competition with
manual labour

OPPORTUNITIES

Demand increase with rise
in Covid,
Increases in market value,
Increases camera
demand

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	6
Title of Experiment	System Architecture, Use Case and Class Diagram
Name of the candidate	Ashish Prakash Singh
Team Members	Karan Keshri, Nilay
Register Number	RA2011033010060
Date of Experiment	11/05/2022

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

Aim

To create system architecture, use case and class diagram.

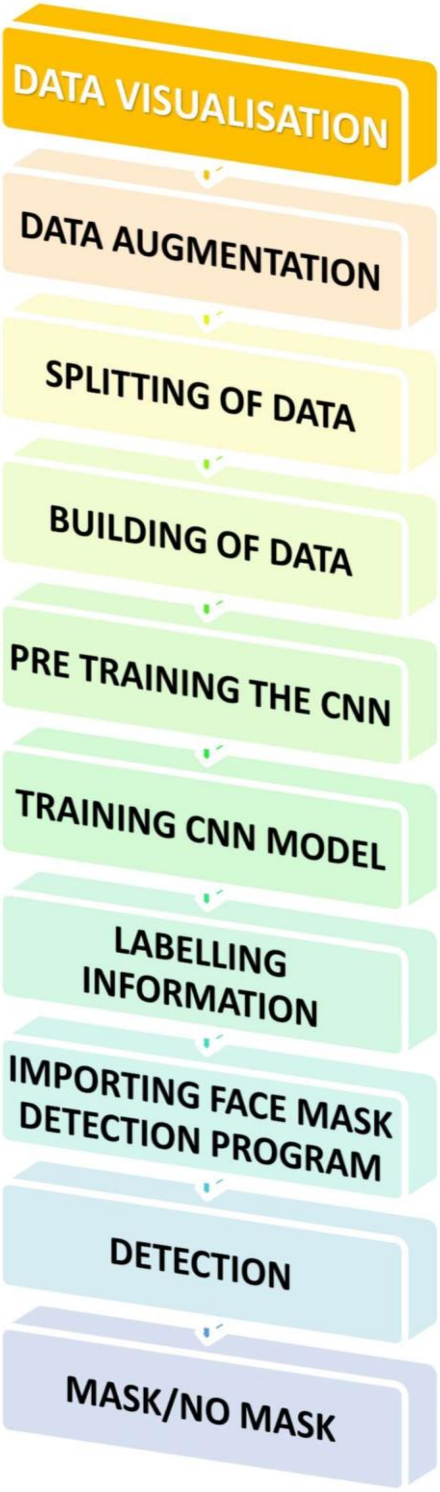
Team Members:

S No	Register No	Name	Role
1	RA2011033010060	Ashish Prakash Singh	**Rep**
2	RA2011033010044	Karan Keshri	**Member**
3	RA2011033010056	Nilay	**Member**

Result:

System architecture, use case and class diagram was created successfully.

6.1 SYSTEM ARCHITECTURE



- Data Visualization.
- Data Augmentation.
- Splitting the data.
- Labeling the Information.
- Importing the Face detection.
- Detecting the Faces with and without Masks.

Data Visualization

In the first step, let us visualize the total number of images in our dataset in both categories. We can see that there are 690 images in the 'yes' class and 686 images in the 'no' class.

Data Augmentation

In the next step, we augment our dataset to include more number of images for our training. In this step of data augmentation, we rotate and flip each of the images in our dataset.

Splitting the data

In this step, we split our data into the training set which will contain the images on which the CNN model will be trained and the test set with the images on which our model will be tested.

Building the Model

In the next step, we build our Sequential CNN model with various layers such as Conv2D, MaxPooling2D, Flatten, Dropout and Dense

Pre-Training the CNN model

After building our model, let us create the 'train_generator' and 'validation_generator' to fit them to our model in the next step.

Training the CNN model

This step is the main step where we fit our images in the training set and the test set to our Sequential model we built using keras library. I have trained the model for 30 epochs (iterations). However, we can train for more number of epochs to attain higher accuracy lest there occurs over-fitting.

Labeling the Information

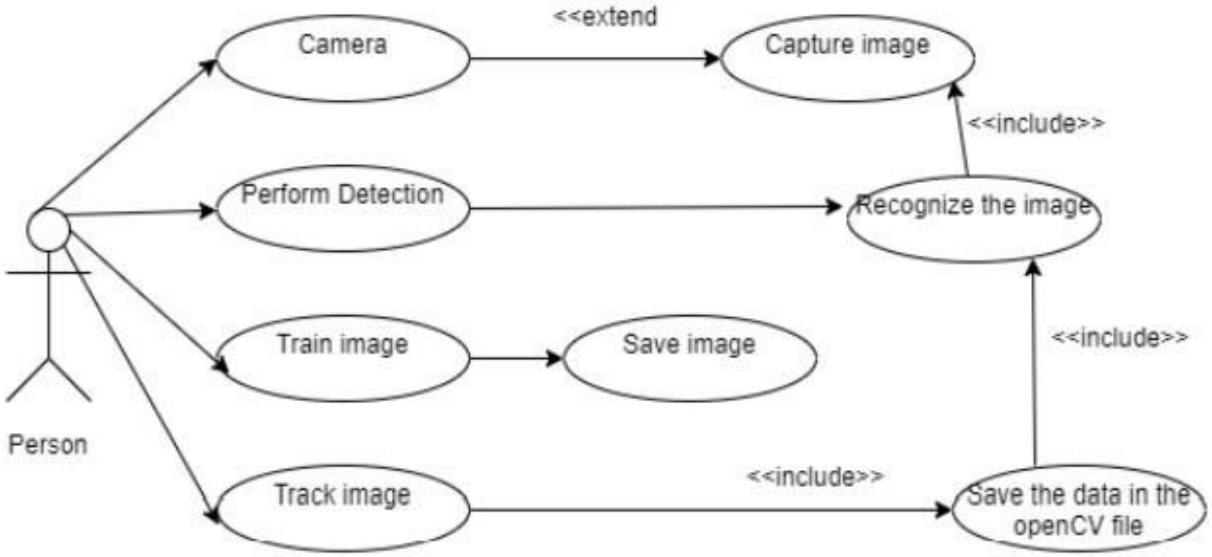
After building the model, we label two probabilities for our results. [‘0’ as ‘without_ ask’ and ‘1’ as ‘with_ mask’]. I am also setting the boundary rectangle color using the RGB values.

6.2 USE CASE DIAGRAMS

In the Unified Modelling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent: ● Scenarios in which your system or application interacts with people, organizations, or external systems.

● Goals that your system or applica on helps those en es (known as actors) achieve.

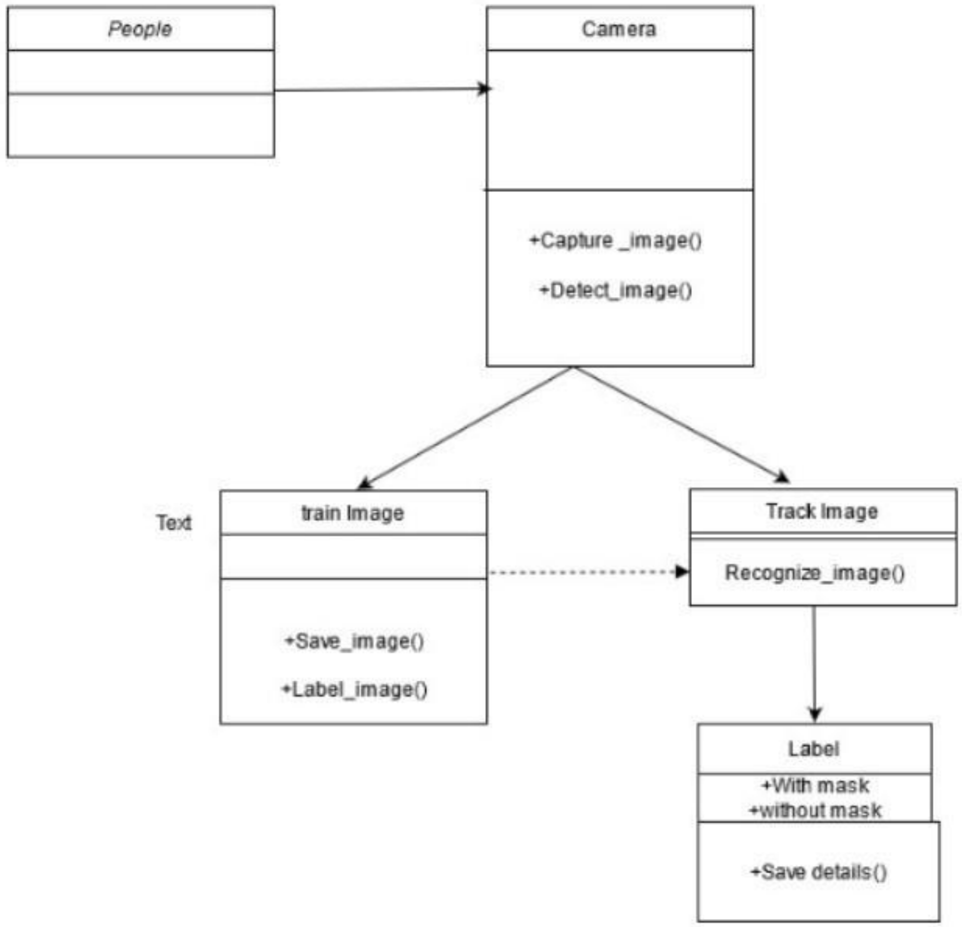
● The scope of your system



6.3 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application.

Class diagrams are the only diagrams which can be directly mapped with object oriented languages and thus widely used at the time of construction.


People
Camera
+Capture _image()
+Detect_image()
Text
train Image
+Save_image()
+Label_image()
Track Image
Recognize_image()
Label
+With mask
+without mask
+Save details()

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	7
Title of Experiment	Design a Entity relationship diagram
Name of the candidate	Ashish Prakash Singh
Team Members	Karan Keshri, Nilay
Register Number	RA2011033010060
Date of Experiment	23/05/2022

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

Aim

To create the Entity Relationship Diagram

Team Members:

S No	Register No	Name	Role
1	RA2011033010060	Ashish Prakash Singh	**Rep**
2	RA2011033010044	Karan Keshri	**Member**
3	RA2011033010056	Nilay	**Member**

Result:

Thus, the entity relationship diagram was created successfully.

<u>*/ ER Diagram, Notation and Example</u> What is ER Diagram?

- ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.
- ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.
- At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique. The purpose of ER Diagram is to represent the entity framework infrastructure.

What is ER Model?

- ER Model stands for Entity Relationship Model is a high-level conceptual data model diagram. ER model helps to systematically analyze data requirements to produce a well-designed database.
- ER Model represents real-world entities and the relationships between them. Creating an ER Model in DBMS is considered as a best practice before implementing your database. - ER Modeling helps you to analyze data requirements systematically to produce a well- designed database. So, it is considered a best practice to complete ER modeling before implementing your database.

Why use ER Diagrams?

Here, are prime reasons for using the ER Diagram

- Helps you to define terms related to entity relationship modeling

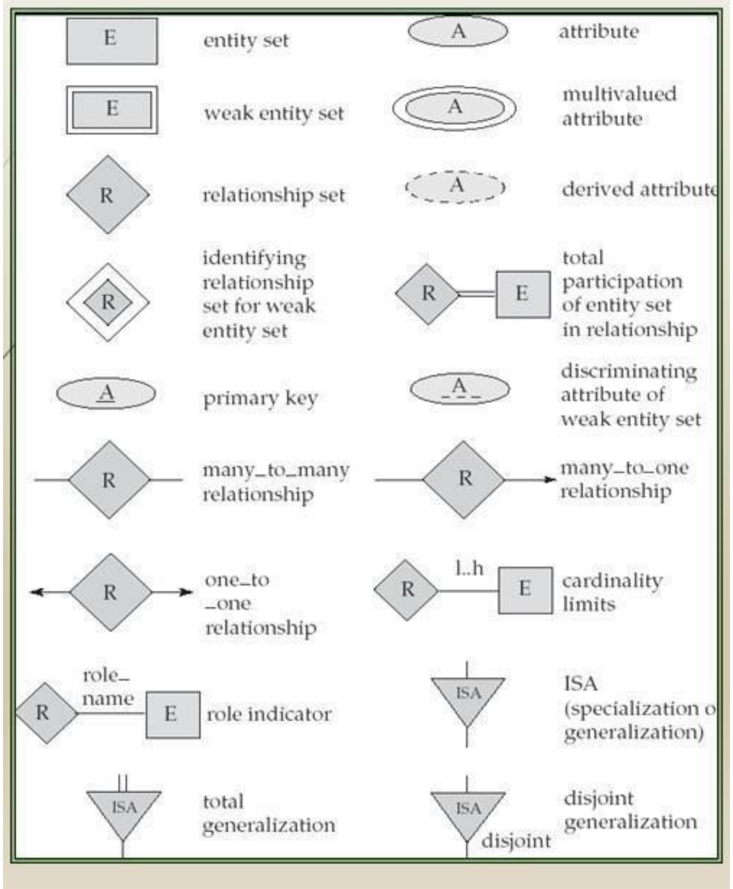
- Provide a preview of how all your tables should connect, what fields are going to be on each table
- Helps to describe entities, attributes, relationships
- ER diagrams are translatable into relational tables which allows you to build databases quickly
- ER diagrams can be used by database designers as a blueprint for implementing data in specific software applications
- The database designer gains a better understanding of the information to be contained in the database with the help of ERP diagram
- ERD Diagram allows you to communicate with the logical structure of the database to users

Components of the ER Diagram

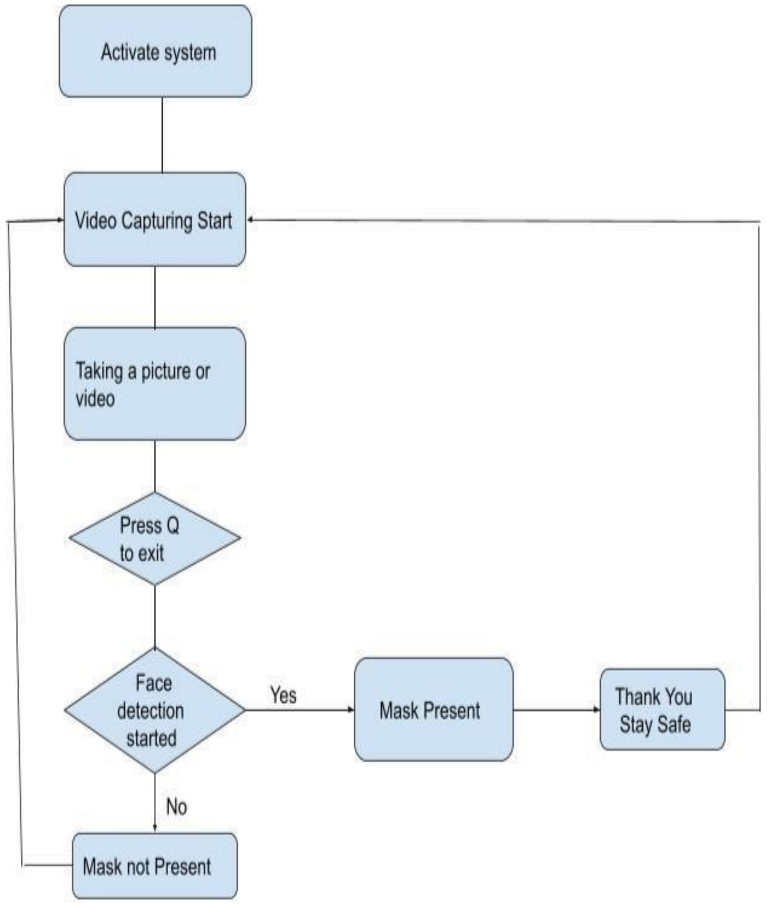
This model is based on three basic concepts: Entities, Attributes, Relationships

ER Diagram – Notations

- Rectangles represent entity sets.
- Diamonds represent relationship sets.
- Lines link attributes to entity sets and entity sets to relationship sets.
- Ellipses represent attributes
- Double ellipses represent multivalued attributes. - Dashed ellipses denote derived attributes.
- Underline indicates primary key attributes



ER Diagram of Face Mask Detection



ADDITIONAL NOTES

- A database can be modeled as a collection of entities, relationship among entities.
- An entity is an object that exists and is distinguishable from other objects. Example: specific person, company, event, plant - Entities have attributes.

Example: people have names and addresses

- An entity set is a set of entities of the same type that share the same properties. Example: set of all persons, companies, trees, holidays

- Express the number of entities to which another entity can be associated via a relationship set.
- Most useful in describing binary relationship sets.
- We express cardinality constraints by drawing either a directed line (->), signifying "one," or an undirected line (—), signifying "many," between the relationship set and the entity set.

- An entity is represented by a set of attributes, that is descriptive properties possessed by all members of an entity set.

Example: customer = (customer-id, customer-name, customer-street, customer-city) loan
= (loan-number, amount)

- Domain – the set of permitted values for each attribute - Attribute types:

1. Simple and composite attributes.
2. Single-valued and multi-valued attributes

E.g. multivalued attribute: phone-numbers

3. Derived attributes-Can be computed from other attributes E.g. age, given date of birth

Cardinality

- For a binary relationship set the mapping cardinality must be one of the following types:

1. One to one

A customer is associated with at most one loan via the relationship borrower. A loan is associated with at most one customer via borrower

2. One to many

A loan is associated with at most one customer via borrower, a customer is associated with several (including 0) loans via borrower

3. Many to one

A loan is associated with several (including 0) customers via borrower, a customer is associated with at most one loan via borrower

4. Many to many

A loan is associated with several (including 0) customers via borrower, a customer is associated with several loans (including 0) via borrower

Weak Entity Set

- An entity set that does not have a primary key is referred to as a weak entity set and represented by double outlined box in E-R diagram.

Example : Consider the entity set payment which got three attributes : payment_number, payment_date and payment_amount. Payment numbers are sequential starting from 1 generally separately for each loan. Although each payment entity is distinct, payments for different loans may share the same payment number. Thus this entity set does not have a primary key.

Discriminator

- The discriminator (or partial key) of a weak entity set is the set of attributes that distinguishes among all the entities of a weak entity set

Example: discriminator of weak entity set payment is the attribute payment_number since for each loan a payment number uniquely identifies one single payment for that loan.

Specialization-Generalization-ISA

- E-R model provides means of representing these distinctive entity groupings
- Process of designating subgroupings within an entity set is called specialization depicted by triangle component labelled ISA (“is a”)
- Bottom up design process in which multiple entity sets are synthesized into higher level entity set - Generalization
- ISA relationship may also be referred to as superclass-subclass relationship
- Higher and lower level entity sets are designated by the terms superclass and subclass. - Specialization and generalization are simple inversions of each other; they are represented in an E-R diagram in the same way.

Total & Partial Participation

- Total participation (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship set

E.g. participation of loan in borrower is total, every loan must have a customer associated to it via borrower

- Partial participation: some entities may not participate in any relationship in the relationship set

Example: participation of customer in borrower is partial

Cardinality limits

- Cardinality limits can also express participation constraints
- Minimum and maximum cardinality is expressed as l..h where l is the minimum and h is the maximum cardinality
- Minimum value of 1 indicates total participation of entity set in relationship set - Maximum value of 1 indicates entity participates in atmost one relationship set. - Maximum value of * indicates no limit

Role indicator

- Entity sets of a relationship need not be distinct
- The labels “manager” and “worker” are called roles; they specify how employee entities interact via the works-for relationship set.
- Roles are indicated in E-R diagrams by labeling the lines that connect diamonds to rectangles.
- Role labels are optional, and are used to clarify semantics of the relationship

Disjoint Generalization

- Disjointness constraint requires that an entity belong to more than one lower level entity set.

Example: account entity can satisfy only one condition for account_type attribute ; entity can either be savings or chequing account but not both.

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	8
Title of Experiment	Develop a Data Flow Diagram (Process-Up to Level 1)
Name of the candidate	Ashish Prakash Singh
Team Members	Karan Keshri, Nilay
Register Number	RA2011033010060
Date of Experiment	23/5/2022

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

Aim

To develop the data flow diagram up to level 1 for the Face Mask Detection.

Team Members:

S No	**Register No**	**Name**	**Role**
1	RA2011033010060	Ashish Prakash Singh	**Rep**
2	RA2011033010044	Karan Keshri	**Member**
3	RA2011033010056	Nilay	**Member**

<DFD >

Result:

Thus, the data flow diagrams have been created for the Face Mask Detection.

Data Flow Diagram

The DFD takes an input-process-output view of a system. That is, data objects flow into the software, are transformed by processing elements, and resultant data objects flow out of the software. Data objects are represented by labeled arrows, and transformations are represented by circles (also called bubbles). The DFD is presented in a hierarchical fashion. That is, the first data flow model (sometimes called a level 0 DFD or context diagram) represents the system as a whole. Subsequent data flow diagrams refine the context diagram, providing increasing detail with each subsequent level.

The data flow diagram enables you to develop models of the information domain and functional domain. As the DFD is refined into greater levels of detail, you perform an implicit functional decomposition of the system. At the same time, the DFD refinement results in a corresponding refinement of data as it moves through the processes that embody the application.

A few simple guidelines can aid immeasurably during the derivation of a data flow diagram:

(1) Level 0 data flow diagram should depict the software/system as asingle bubble;

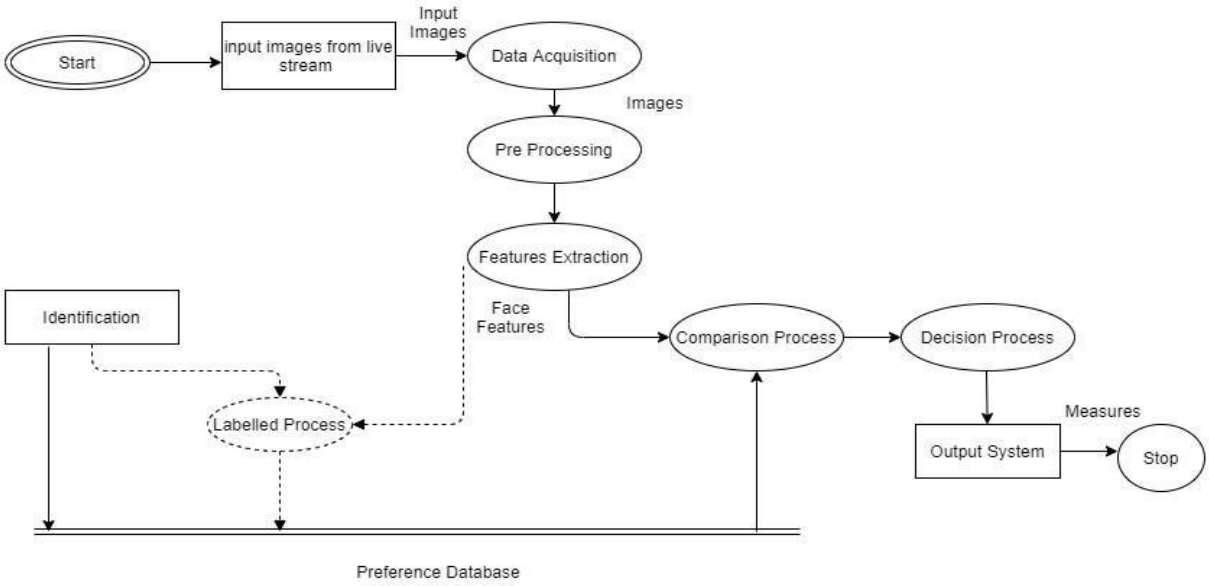
(2) Primary input and output should be carefully noted;

(3) Refinement should begin by isolating candidate processes, data objects, and data stores to be represented at the next level;

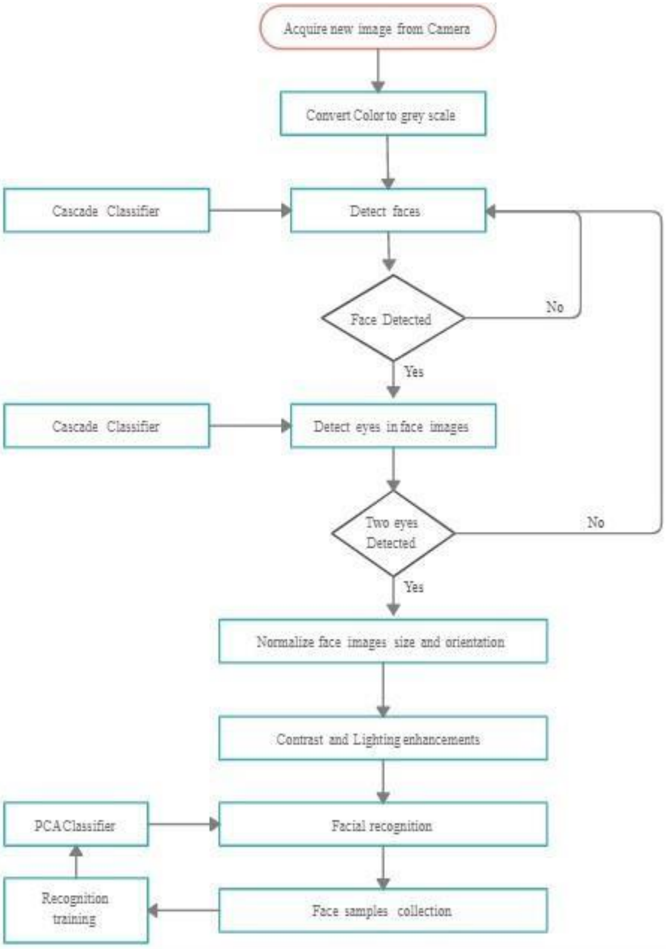
(4) All arrows and bubbles should be labeled with meaningful names;
(5) Information flow continuity must be maintained from level to level and
(6) One bubble at a time should be refined. There is a natural tendency to overcomplicate the data flow diagram. This occurs when you attempt to show too much detail too early or represent procedural aspects of the software in lieu of information flow.

DATA FLOW DIAGRAM

Level 0



Level 1


Acquire new image from Camera
Convert Color to grey scale
Cascade Classifier
Detect faces
Face Detected
No
Yes
Cascade Classifier
Detect eyes in face images
Two eyes Detected
No
Yes
Normalize face images size and orientation
Contrast and Lighting enhancements
PCA Classifier
Facial recognition
Recognition training
Face samples collection

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	9
Title of Experiment	Design a Sequence and Collaboration Diagram
Name of the candidate	ASHISH PRAKASH SINGH
Team Members	NILAY, KARAN KESHRI
Register Number	RA2011033010060
Date of Experiment	30/05/2022

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	10	

Staff Signature with date

Aim

To create the sequence and collaboration diagram for the Face Mask Detection

Team Members:

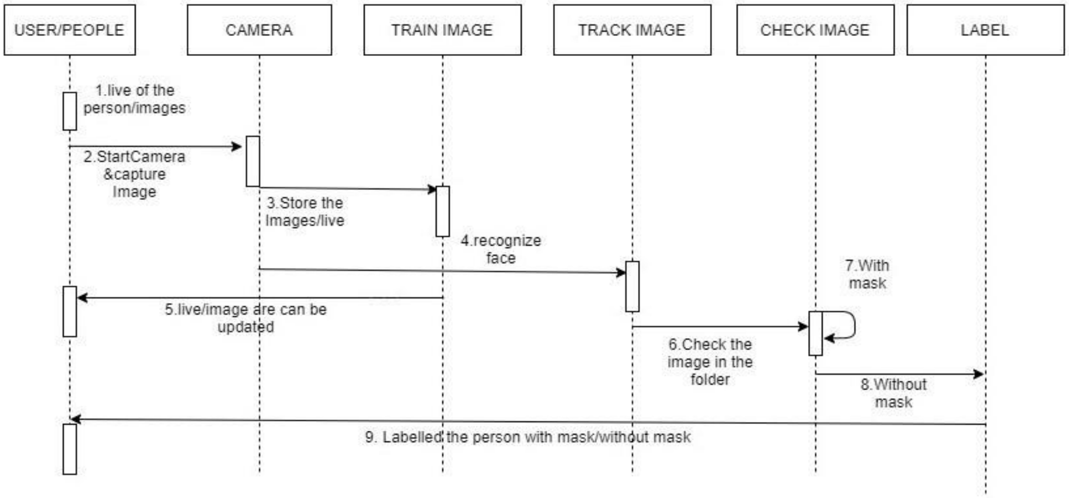
S No	Register No	Name	Role
1	RA2011033010060	ASHISH PRAKASH SINGH	Rep/Member
2	RA2011033010056	NILAY	Member
3	RA2011033010044	KARAN KESHRI	Member

<Sequence and Collaboration Diagram>

Result:

Thus, the sequence and collaboration diagrams were created for the Face Mask Detection.

Sequence Diagram


USER/PEOPLE
CAMERA
TRAIN IMAGE
TRACK IMAGE
CHECK IMAGE
LABEL
1.live of the person/images
2.StartCamera &capture Image
3.Store the Images/live
4.recognize face
5.live/image are can be updated
6.Check the image in the folder
7.With mask
8.Without mask
9. Labelled the person with mask/without mask

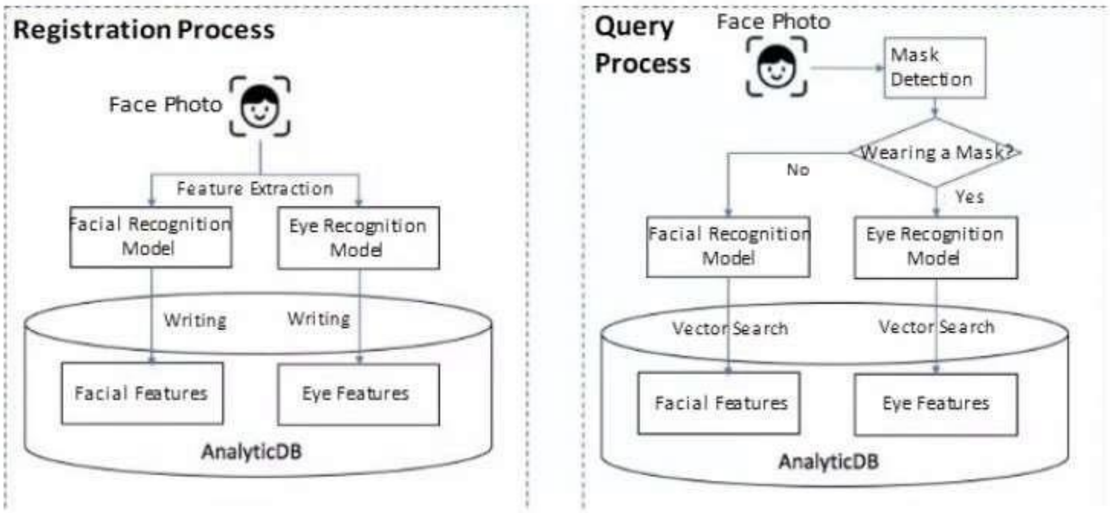
Justification –

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

Sequence diagrams can be useful references for businesses and other organizations. Try drawing a sequence diagram to:

- Represent the details of a UML use case.
- Model the logic of a sophisticated procedure, function, or operation.
- See how objects and components interact with each other to complete a process.
- Plan and understand the detailed functionality of an existing or future scenario.

Collaboration Diagram-



Justification-

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.

Collaboration diagrams are created by first identifying the structural elements required to carry out the functionality of an interaction. A model is then built using the relationships between those elements. Several vendors offer software for creating and editing collaboration diagrams.

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	10
Title of Experiment	Develop a Testing Framework/User Interface
Name of the candidate	Ashish Prakash Singh
Team Members	Karan Keshri, Nilay
Register Number	RA2011033010060
Date of Experiment	06/06/2022

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		**10**	

Staff Signature with date

Aim

To develop a testing framework/user interface

Team Members:

S No	**Register No**	**Name**	**Role**
1	RA2011033010060	Ashish Prakash Singh	**Rep**
2	RA2011033010044	Karan Keshri	**Member**
3	RA2011033010056	Nilay	**Member**

Result:

Testing framework/user interface was developed successfully.

Executive Summary

Scope –

Scope of this project is detecting whether person is wearing facemask or not, if they are not wearing mask then it will through the error message (Please wear mask) It will also beep and through red light.

Objective –

Scope of this project is detecting whether person is wearing facemask or not. It is very important for our daily life.

Approach –

I. Unit testing – Here we are testing each module and if error comes it will be solve at unit level.
II. Integration testing – Here we are combining different modules and testing there compatibility.
III. System testing – Here we are going to test its compatibility with different system.
IV. Acceptance testing – Here we are going to test it with whether it will be accepting by user or whether fulfil users requirement or not. Modules –

1. cv2 module – Here we are using cv2 module of python library. It will detect device surrounding and open camera.
2. tensorflow module – This will use for calculation of numeric values and it will return the value in some percentage.
3. os module – This will use to interact with operating system.
4. matplotlib.pyplot – This module is used to create 2D graphs and plots by using python script. It makes thing easy for easy plotting by providing features to control line style, font properties, formatting axes.
5. numpy – Here it is use for performing mathematical operation like percentage calculator.

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	11
Title of Experiment	Test Cases
Name of the candidate	Ashish Prakash Singh
Team Members	Karan Keshri, Nilay
Register Number	RA2011033010060
Date of Experiment	4/6/2022

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

Aim

To develop the test cases manual for the face mask detection project.

Team Members:

S No	Register No	Name	Role
1	Ashish Prakash Singh	RA2011033010060	**Rep**
2	Karan Keshri	RA2011033010044	**Member**
3	Nilay	RA2011033010056	**Member**

<Utilize the templates below and incorporate the project's test cases - Manual Test case to be written for at least one module >

Result:

Thus, the test case manual has been created for the face mask detection.

Test Case

Functional Test Cases

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
	Device Camera Detection	Checking background	It will detect camera quality first. It will check proper lightning background.	It will check the environment and then detect face.	After checking the environment it starts detecting face.	Pass	Success

	Face Detection	Face Detect	Whenever face appears it will automatically detect.	Face will be detected automatically and make a box.	It is detecting automatically and make a box.	Pass	Success
	Picture Grayscale	Picture conversion into grayscale	After detecting Picture it will convert it into grayscale.	It will convert picture into grayscale.	After detecting it convert picture into grayscale.	Pass	Success
	Matching with Given Data	It match picture with given data	After matching the picture it will compare with given module	It will compare with given result.	After comparison with gives result.	Pass	Success
	Mask Showing	Mask Detection	It will Detect mask.	Showing Mask Wearing.	It is showing Whether person is wearing mask or not.	Pass	Success

Non-Functional Test Cases

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
	Number of user	Detecting Face	It will Start Detecting faces.	It will detect multiple faces at a time	It is detecting multiple faces at a time	Pass	Success

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	12
Title of Experiment	Manual Test Case Reporting
Name of the candidate	ASHISH PRAKASH SINGH
Team Members	NILAY, KARAN KESHRI
Register Number	RA2011033010056
Date of Experiment	06/06/2022

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

Aim

To prepare the manual test case report for the face mask detection project.

Team Members:

S No	Register No	Name	Role
1	RA2011033010060	ASHISH PRAKASH SINGH	**Rep/Member**
2	RA2011033010056	NILAY	**Member**
3	RA2011033010044	KARAN KESHRI	**Member**

<Manual Test Case Report to be incorporated >
<< Summarize the current status of the Testing>
<<present obstacles to proceed further >>
<< Seek help from stakeholders to remove obstacles/constraints>>

Category	Progress Against Plan	Status
Functional Testing	Green / Amber / Red	Not-Started / In-Progress / Completed
Device Camera Detection	Green	Completed
Face Detection	Green	Completed
Picture Grayscale	Green	Completed
Matching with Given Data	Green	Completed
Mask Showing	Amber	In-Progress
Non-Functional Testing		
Number of user	Green	Completed

Functional	Test Case Coverage (%)	Status
Module ID	30%	Not-Started / In-Progress / Completed
Face mask detecting	70%	In Progress

Result: Thus, the test case report has been created for the face mask detection.

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	13
Title of Experiment	Architecture design and framework implementation
Name of the candidate	ASHISH PRAKASH SINGH
Team Members	NILAY, KARAN KESHRI
Register Number	RA2011033010056
Date of Experiment	13/06/2022

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

Aim

To prepare architecture design and framework implementation.

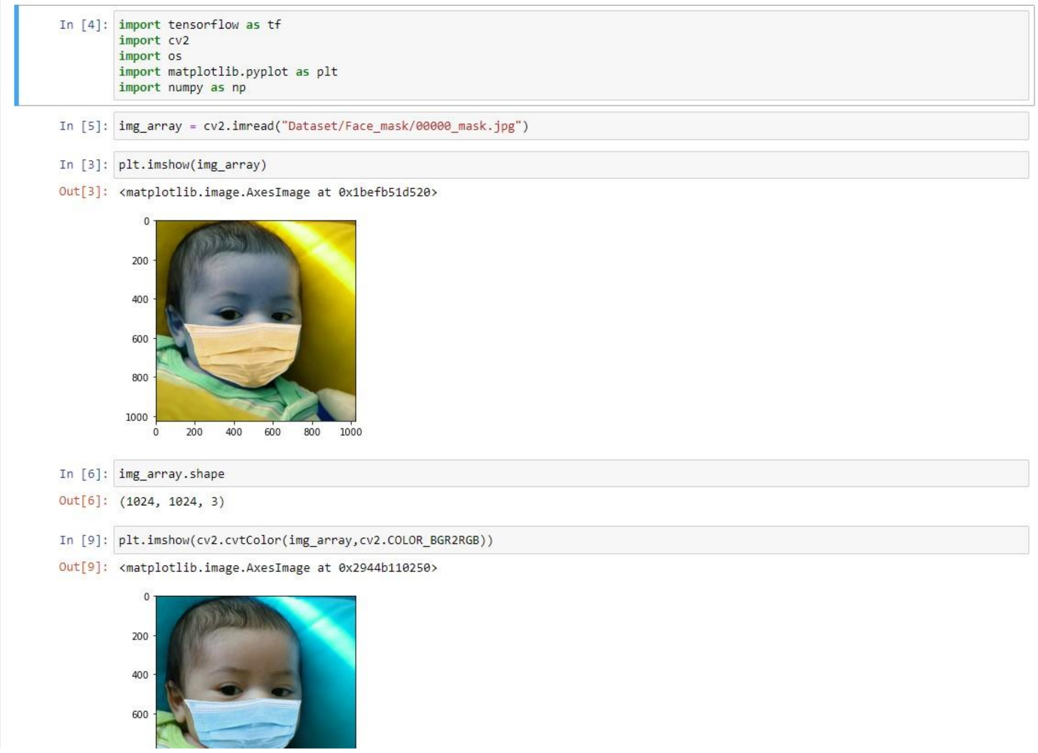
Team Members:

S No	Register No	Name	Role
1	RA2011033010060	ASHISH PRAKASH SINGH	**Rep/Member**
2	RA2011033010056	NILAY	**Member**
3	RA2011033010044	KARAN KESHRI	**Member**

PROJECT TITLE- FACE MASK DETECTION

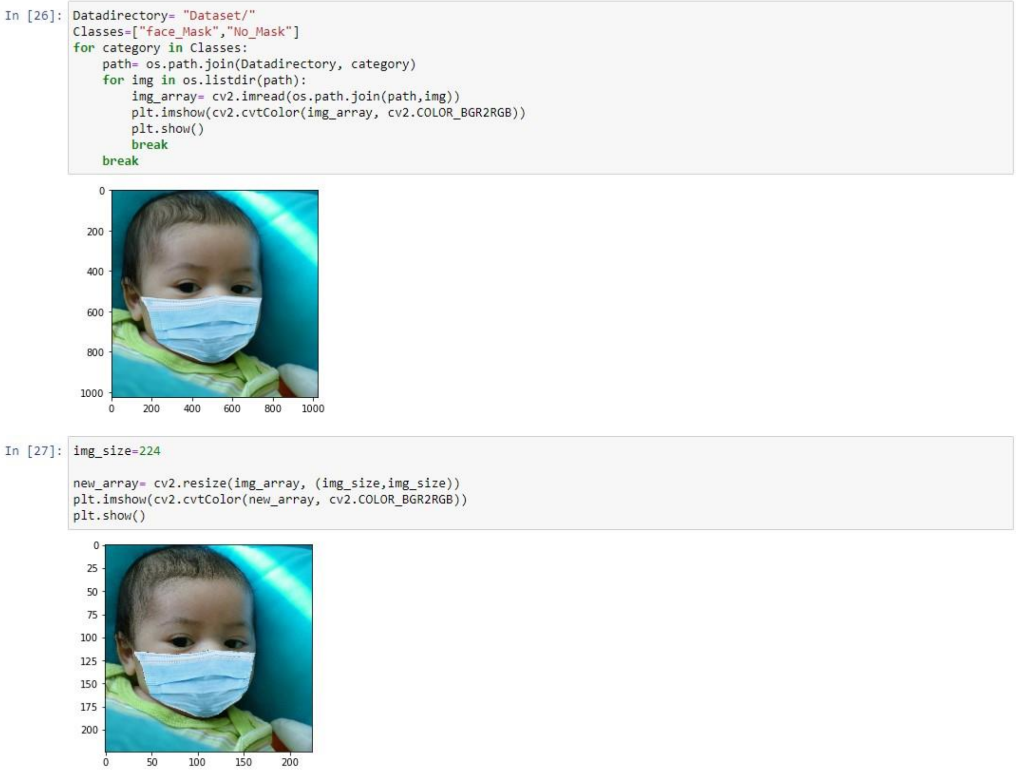
CODING PART-

To load images using cv and convert them into array, to train model and detect values for face mask or without mask(with output)



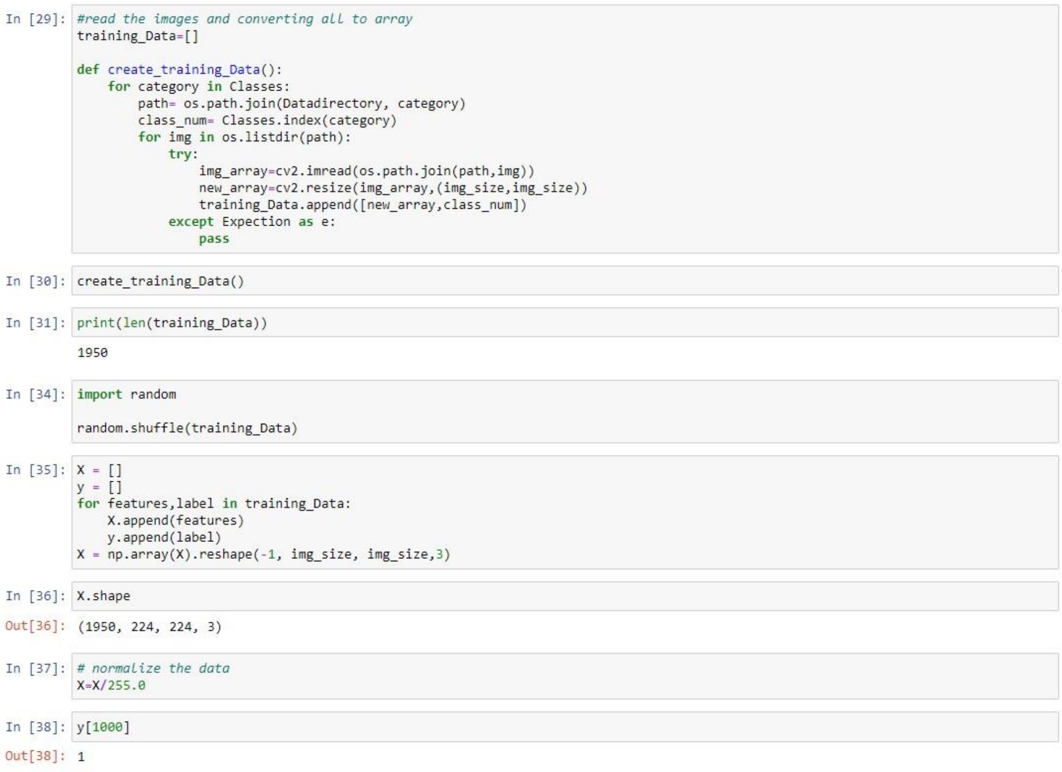
```
In [4]: import tensorflow as tf
        import cv2
        import os
        import matplotlib.pyplot as plt
        import numpy as np

In [5]: img_array = cv2.imread("Dataset/Face_mask/00000_mask.jpg")

In [3]: plt.imshow(img_array)

Out[3]: <matplotlib.image.AxesImage at 0x1befb51d520>
```

```
In [6]: img_array.shape

Out[6]: (1024, 1024, 3)

In [9]: plt.imshow(cv2.cvtColor(img_array,cv2.COLOR_BGR2RGB))

Out[9]: <matplotlib.image.AxesImage at 0x2944b110250>
```

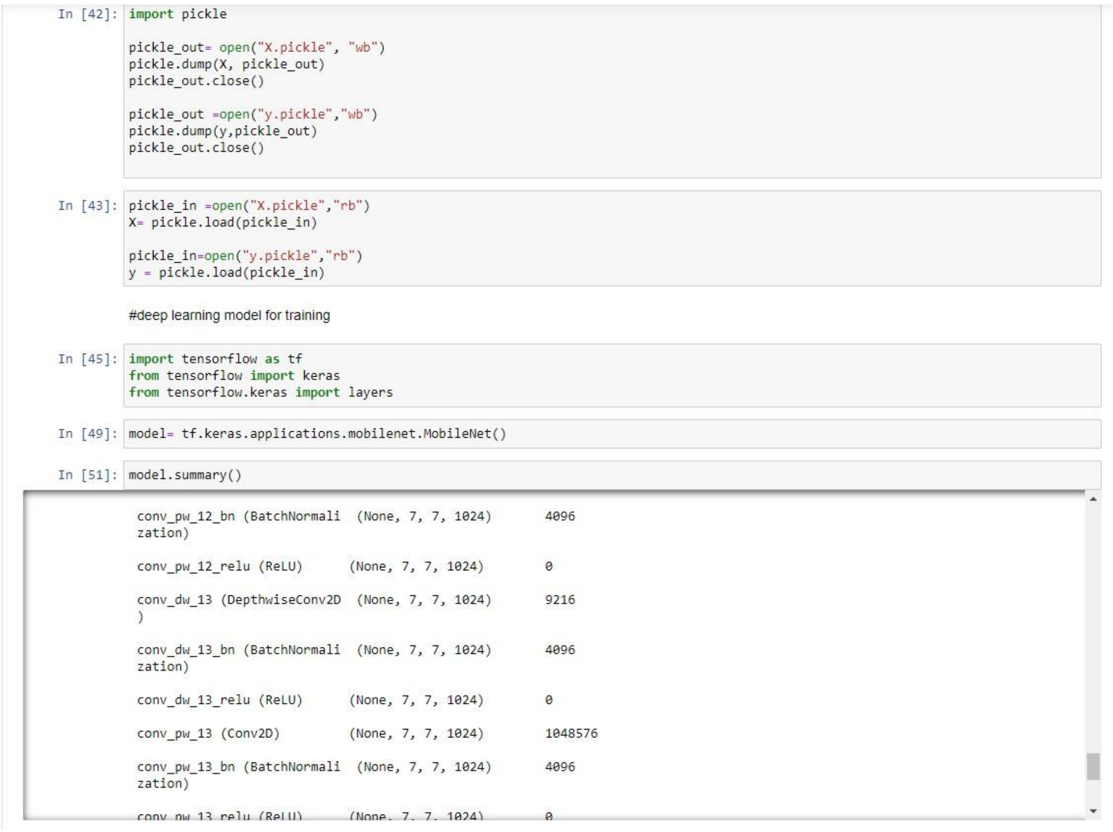


```
In [26]: Datadirectory= "Dataset/"
         Classes=["face_Mask","No_Mask"]
         for category in Classes:
             path= os.path.join(Datadirectory, category)
             for img in os.listdir(path):
                 img_array= cv2.imread(os.path.join(path,img))
                 plt.imshow(cv2.cvtColor(img_array, cv2.COLOR_BGR2RGB))
                 plt.show()
                 break
             break
```

```
In [27]: img_size=224

         new_array= cv2.resize(img_array, (img_size,img_size))
         plt.imshow(cv2.cvtColor(new_array, cv2.COLOR_BGR2RGB))
         plt.show()
```



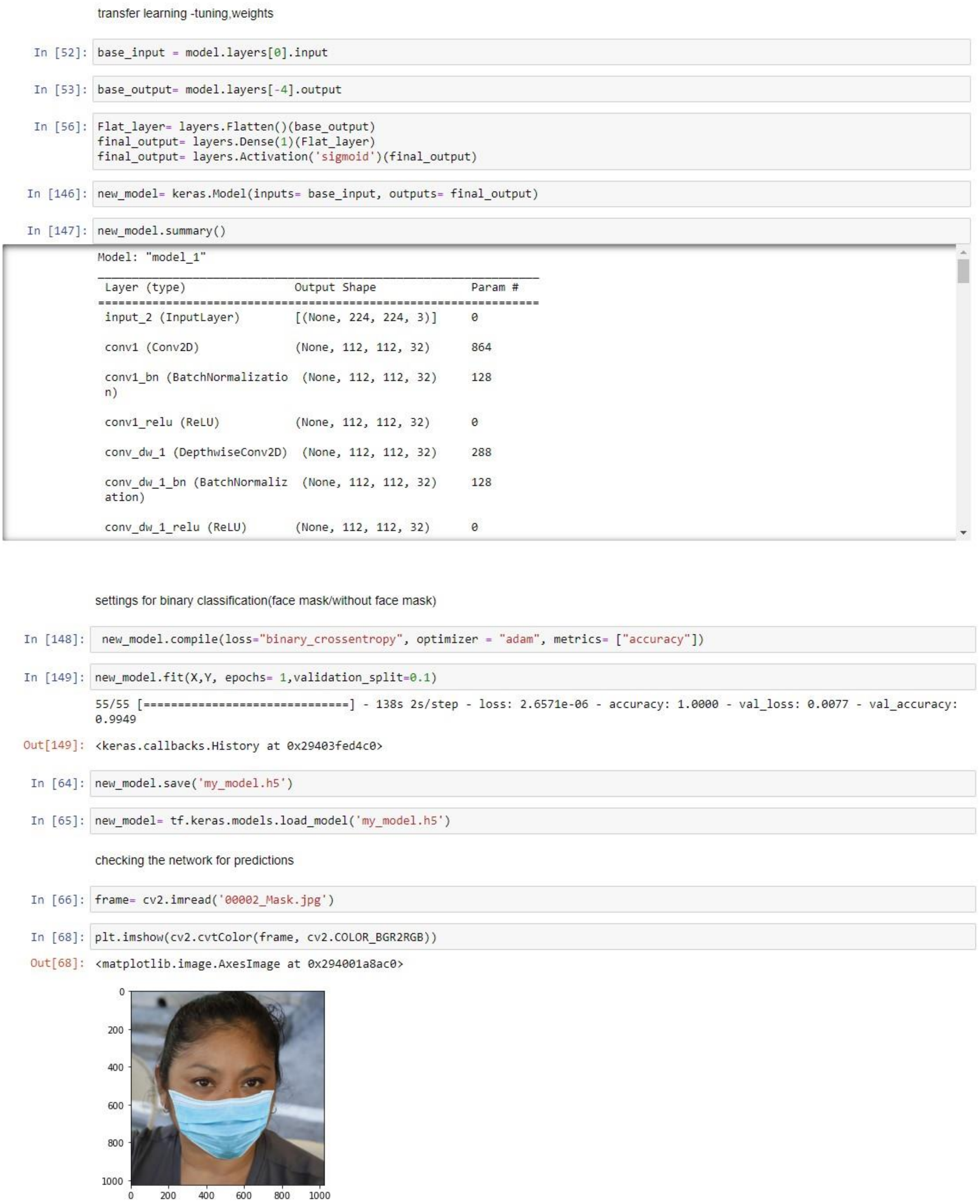
```
In [29]: #read the images and converting all to array
         training_Data=[]

         def create_training_Data():
             for category in Classes:
                 path= os.path.join(Datadirectory, category)
                 class_num= Classes.index(category)
                 for img in os.listdir(path):
                     try:
                         img_array=cv2.imread(os.path.join(path,img))
                         new_array=cv2.resize(img_array,(img_size,img_size))
                         training_Data.append([new_array,class_num])
                     except Expection as e:
                         pass

In [30]: create_training_Data()

In [31]: print(len(training_Data))

         1950

In [34]: import random

         random.shuffle(training_Data)

In [35]: X = []
         y = []
         for features,label in training_Data:
             X.append(features)
             y.append(label)
         X = np.array(X).reshape(-1, img_size, img_size,3)

In [36]: X.shape

Out[36]: (1950, 224, 224, 3)

In [37]: # normalize the data
         X=X/255.0

In [38]: y[1000]

Out[38]: 1
```



```
In [42]: import pickle

         pickle_out= open("X.pickle", "wb")
         pickle.dump(X, pickle_out)
         pickle_out.close()

         pickle_out =open("y.pickle","wb")
         pickle.dump(y,pickle_out)
         pickle_out.close()

In [43]: pickle_in =open("X.pickle","rb")
         X= pickle.load(pickle_in)

         pickle_in=open("y.pickle","rb")
         y = pickle.load(pickle_in)
```

#deep learning model for training

```
In [45]: import tensorflow as tf
         from tensorflow import keras
         from tensorflow.keras import layers

In [49]: model= tf.keras.applications.mobilenet.MobileNet()

In [51]: model.summary()

 conv_pw_12_bn (BatchNormali  (None, 7, 7, 1024)       4096
 zation)

 conv_pw_12_relu (ReLU)      (None, 7, 7, 1024)        0

 conv_dw_13 (DepthwiseConv2D  (None, 7, 7, 1024)       9216
 )

 conv_dw_13_bn (BatchNormali  (None, 7, 7, 1024)       4096
 zation)

 conv_dw_13_relu (ReLU)      (None, 7, 7, 1024)        0

 conv_pw_13 (Conv2D)         (None, 7, 7, 1024)        1048576

 conv_pw_13_bn (BatchNormali  (None, 7, 7, 1024)       4096
 zation)

 conv_pw_13_relu (ReLU)      (None, 7, 7, 1024)        0
```



transfer learning -tuning,weights

```
In [52]: base_input = model.layers[0].input
```

```
In [53]: base_output= model.layers[-4].output
```

```
In [56]: Flat_layer= layers.Flatten()(base_output)
         final_output= layers.Dense(1)(Flat_layer)
         final_output= layers.Activation('sigmoid')(final_output)
```

```
In [146]: new_model= keras.Model(inputs= base_input, outputs= final_output)
```

```
In [147]: new_model.summary()
```

```
Model: "model_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 224, 224, 3)]     0

 conv1 (Conv2D)              (None, 112, 112, 32)      864

 conv1_bn (BatchNormalizatio  (None, 112, 112, 32)     128
 n)

 conv1_relu (ReLU)           (None, 112, 112, 32)      0

 conv_dw_1 (DepthwiseConv2D)  (None, 112, 112, 32)     288

 conv_dw_1_bn (BatchNormaliz  (None, 112, 112, 32)     128
 ation)

 conv_dw_1_relu (ReLU)       (None, 112, 112, 32)      0
```

settings for binary classification(face mask/without face mask)

```
In [148]:  new_model.compile(loss="binary_crossentropy", optimizer = "adam", metrics= ["accuracy"])
```

```
In [149]: new_model.fit(X,Y, epochs= 1,validation_split=0.1)
          55/55 [==============================] - 138s 2s/step - loss: 2.6571e-06 - accuracy: 1.0000 - val_loss: 0.0077 - val_accuracy:
          0.9949
Out[149]: <keras.callbacks.History at 0x29403fed4c0>
```

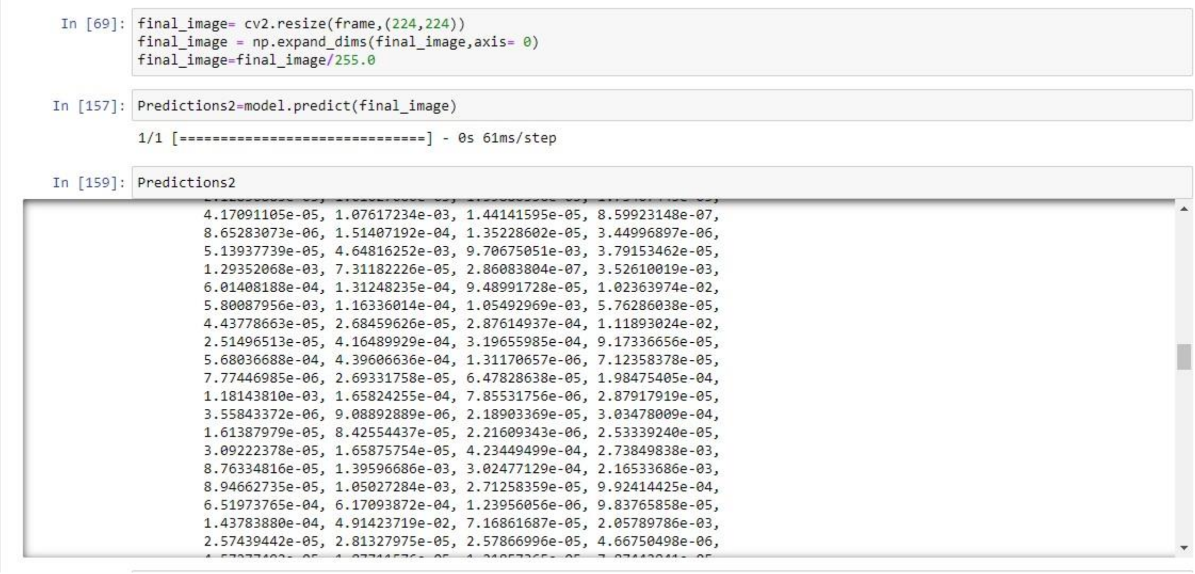
```
In [64]: new_model.save('my_model.h5')
```

```
In [65]: new_model= tf.keras.models.load_model('my_model.h5')
```

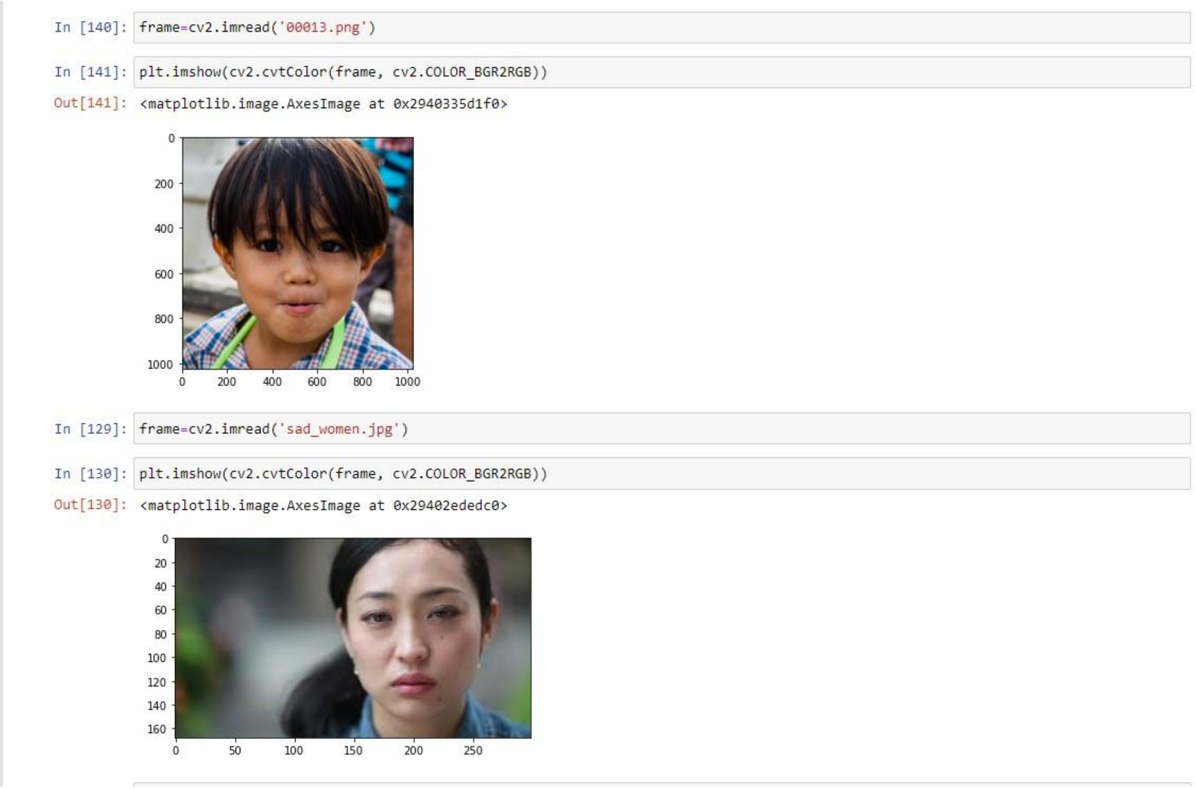
checking the network for predictions

```
In [66]: frame= cv2.imread('00002_Mask.jpg')
```

```
In [68]: plt.imshow(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
Out[68]: <matplotlib.image.AxesImage at 0x294001a8ac0>
```

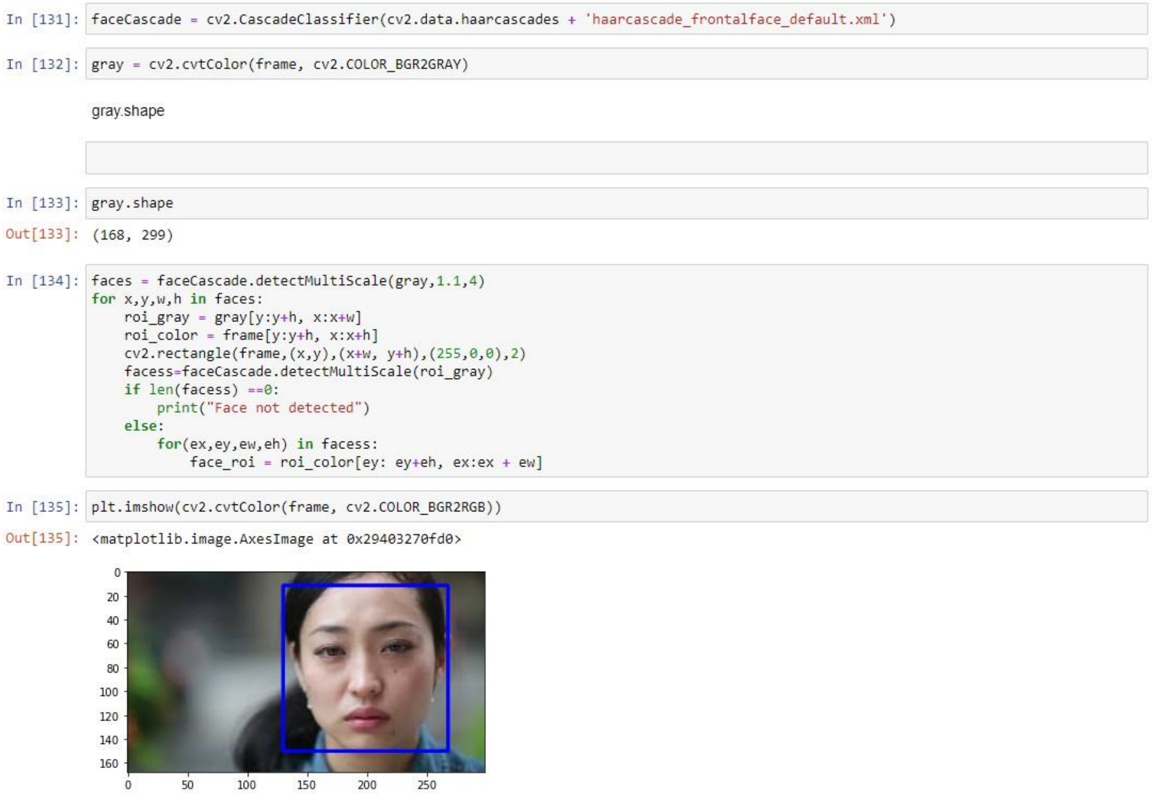


```
In [69]: final_image= cv2.resize(frame,(224,224))
         final_image = np.expand_dims(final_image,axis= 0)
         final_image=final_image/255.0

In [157]: Predictions2=model.predict(final_image)

          1/1 [==============================] - 0s 61ms/step

In [159]: Predictions2

                4.17091105e-05, 1.07617234e-03, 1.44141595e-05, 8.59923148e-07,
                8.65283073e-06, 1.51407192e-04, 1.35228602e-05, 3.44996897e-06,
                5.13937739e-05, 4.64816252e-03, 9.70675051e-03, 3.79153462e-05,
                1.29352068e-03, 7.31182226e-05, 2.86083804e-07, 3.52610019e-03,
                6.01408188e-04, 1.31248235e-04, 9.48991728e-05, 1.02363974e-02,
                5.80087956e-03, 1.16336014e-04, 1.05492969e-03, 5.76286038e-05,
                4.43778663e-05, 2.68459626e-05, 2.87614937e-04, 1.11893024e-02,
                2.51496513e-05, 4.16489929e-04, 3.19655985e-04, 9.17336656e-05,
                5.68036688e-04, 4.39606636e-04, 1.31170657e-06, 7.12358378e-05,
                7.77446985e-06, 2.69331758e-05, 6.47828638e-05, 1.98475405e-04,
                1.18143810e-03, 1.65824255e-04, 7.85531756e-06, 2.87917919e-05,
                3.55843372e-06, 9.08892889e-06, 2.18903369e-05, 3.03478009e-04,
                1.61387979e-05, 8.42554437e-05, 2.21609343e-06, 2.53339240e-05,
                3.09222378e-05, 1.65875754e-05, 4.23449499e-04, 2.73849838e-03,
                8.76334816e-05, 1.39596686e-03, 3.02477129e-04, 2.16533686e-03,
                8.94662735e-05, 1.05027284e-03, 2.71258359e-05, 9.92414425e-04,
                6.51973765e-04, 6.17093872e-04, 1.23956056e-06, 9.83765858e-05,
                1.43783880e-04, 4.91423719e-02, 7.16861687e-05, 2.05789786e-03,
                2.57439442e-05, 2.81327975e-05, 2.57866996e-05, 4.66750498e-06,
```



```
In [140]: frame=cv2.imread('00013.png')

In [141]: plt.imshow(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))

Out[141]: <matplotlib.image.AxesImage at 0x2940335d1f0>
```

```
In [129]: frame=cv2.imread('sad_women.jpg')

In [130]: plt.imshow(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))

Out[130]: <matplotlib.image.AxesImage at 0x29402ededc0>
```



```
In [131]: faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
```

```
In [132]: gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

gray.shape

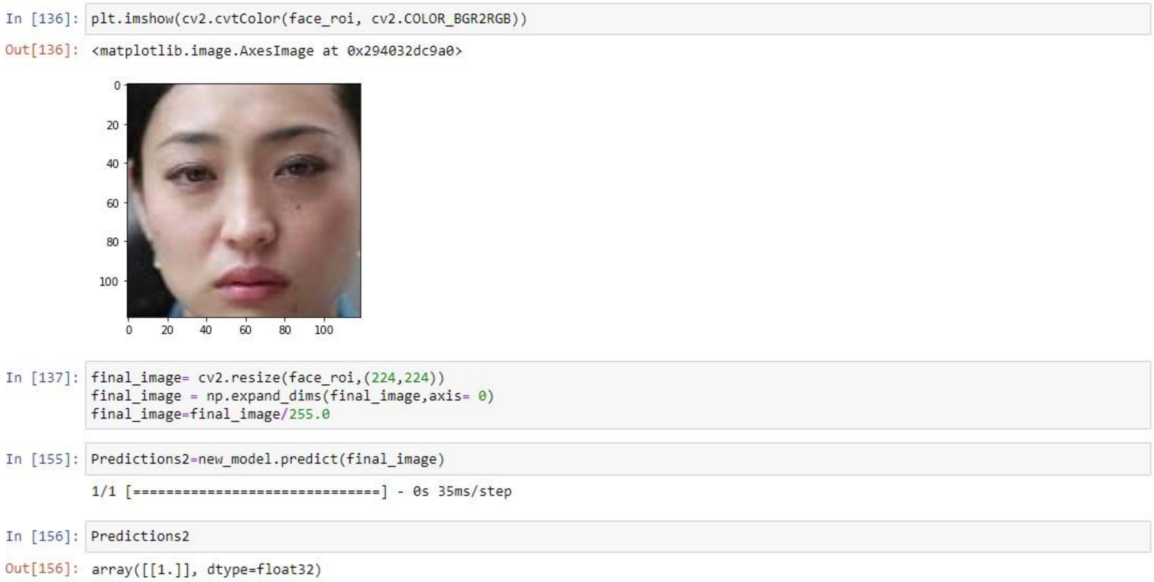
```
In [133]: gray.shape
```

```
Out[133]: (168, 299)
```

```
In [134]: faces = faceCascade.detectMultiScale(gray,1.1,4)
          for x,y,w,h in faces:
              roi_gray = gray[y:y+h, x:x+w]
              roi_color = frame[y:y+h, x:x+h]
              cv2.rectangle(frame,(x,y),(x+w, y+h),(255,0,0),2)
              facess=faceCascade.detectMultiScale(roi_gray)
              if len(facess) ==0:
                  print("Face not detected")
              else:
                  for(ex,ey,ew,eh) in facess:
                      face_roi = roi_color[ey: ey+eh, ex:ex + ew]
```

```
In [135]: plt.imshow(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
```

```
Out[135]: <matplotlib.image.AxesImage at 0x29403270fd0>
```



```
In [136]: plt.imshow(cv2.cvtColor(face_roi, cv2.COLOR_BGR2RGB))
```

```
Out[136]: <matplotlib.image.AxesImage at 0x294032dc9a0>
```

```
In [137]: final_image= cv2.resize(face_roi,(224,224))
          final_image = np.expand_dims(final_image,axis= 0)
          final_image=final_image/255.0
```

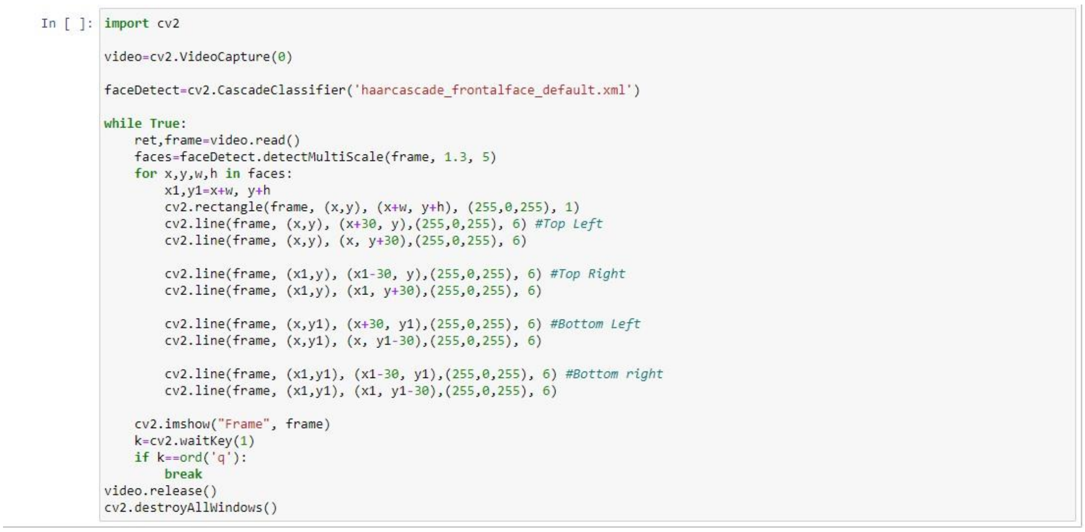
```
In [155]: Predictions2=new_model.predict(final_image)

          1/1 [==============================] - 0s 35ms/step
```

```
In [156]: Predictions2
```

```
Out[156]: array([[1.]], dtype=float32)
```

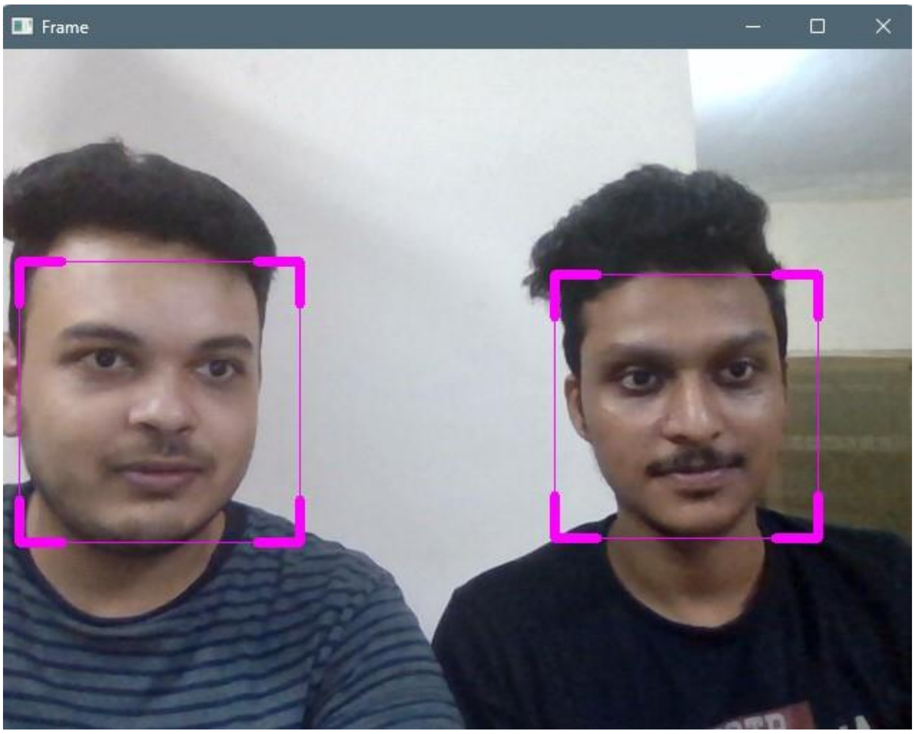
To open camera and detect multiple faces-

Code-



```
import cv2

video=cv2.VideoCapture(0)

faceDetect=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

while True:
    ret,frame=video.read()
    faces=faceDetect.detectMultiScale(frame, 1.3, 5)
    for x,y,w,h in faces:
        x1,y1=x+w, y+h
        cv2.rectangle(frame, (x,y), (x+w, y+h), (255,0,255), 1)
        cv2.line(frame, (x,y), (x+30, y),(255,0,255), 6) #Top Left
        cv2.line(frame, (x,y), (x, y+30),(255,0,255), 6)

        cv2.line(frame, (x1,y), (x1-30, y),(255,0,255), 6) #Top Right
        cv2.line(frame, (x1,y), (x1, y+30),(255,0,255), 6)

        cv2.line(frame, (x,y1), (x+30, y1),(255,0,255), 6) #Bottom Left
        cv2.line(frame, (x,y1), (x, y1-30),(255,0,255), 6)

        cv2.line(frame, (x1,y1), (x1-30, y1),(255,0,255), 6) #Bottom right
        cv2.line(frame, (x1,y1), (x1, y1-30),(255,0,255), 6)

    cv2.imshow("Frame", frame)
    k=cv2.waitKey(1)
    if k==ord('q'):
        break
video.release()
cv2.destroyAllWindows()
```

Output-



Result- Thus, the details of architectural design/framework/implementation along with the screenshots were provided

Conclusion

As the technology are blooming with emerging trends the availability so we have novel face mask detector which can possibly contribute to public healthcare. The architecture consists of Mobile Net as the backbone it can be used for high and low computation scenarios. In order to extract more robust features, we utilize transfer learning to adopt weights from a similar task face detection, which is trained on a very large dataset. We used OpenCV, tensor flow, and NN to detect whether people were wearing face masks or not. The models were tested with images and real-time video streams. The accuracy of the model is achieved and, the optimization of the model is a continuous process and we are building a highly accurate solution by tuning the hyper parameters. This specific model could be used as a use case for edge analytics. Furthermore, the proposed method achieves state-of-the-art results on a public face mask dataset. By the development of face mask-detection we can detect if the person is wearing a face mask and allow their entry would be of great help to the society

Reference

M. M. Rahman, M. M. H. Manik, M. M. Islam, S. Mahmud and J. -H. Kim, "An Automated System to Limit COVID-19 Using Facial Mask Detection in Smart City Network," 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), 2020

D. Yi, Z. Lei, S. Liao, and S. Z. Li, “Learning face representation from scratch,” CoRR abs/1411.7923, 2014.

X. Cao, D. Wipf, F. Wen, G. Duan, and J. Sun, “A practical transfer learning algorithm for face verification,” in Computer Vision (ICCV), 2013 IEEE International Conference on, pp. 3208–3215, IEEE, 2013.

M. R. Bhuiyan, S. A. Khushbu and M. S. Islam, "A Deep Learning Based Assistive System to Classify COVID-19 Face Mask for Human Safety with YOLOv3," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)