BEng. (Hons) in Software Engineering
Level 6

Mask Detection Attendance System

Prepared By: Dilupa Bandara Chandrasekara (CB007729) [HF20A1SEENG]

Date of Submission: 30th June, 2021.

Supervisor: Umanga Pilapitiya

Abstract

The objective of this research paper is to document the development of a university attendance system which can detect whether the student/lecturer is wearing a mask before entering to the premise. Mask Detection model is developed using open cv, mobilenet, TensorFlow and keras in Python. First a student/lecturer is searched for whether he/she is wearing a mask. Then the result is sent to the system. If the system has proven that the user is wearing a mask, then the user needs to input the RFID card into Reader to mark the attendance. The user can check the attendance percentage also. Then the admin handles all the student administration activities such as Adding, Deleting and Updating Students/Lecturers.

This analysis contains the literature review which includes a thorough domain analysis, technology comparisons as well as an analysis on mask detection models.

The chapters solution concept and the methodology give a deeper understanding of how the solution is proposed, the methodologies used in implementation and SLDC model followed.

Acknowledgement

I would like to take this opportunity to thank my supervisor Mrs. Umanga Pilapitya for successfully guiding me through this project and Mr. Nipunu Wijesinghe for providing insights, guidance, and new ideas to make this project a success.

Last, I would like to thank my family, batchmates and schoolmates who have supported me throughout this journey.

Table of Contents

Contents

Table of Figures

Chapter 1- Introduction

1.1 Problem Context

Plagues are a recurring phenomenon happening throughout the history of mankind. The newest plague that we are facing is the coronavirus. The first reported case is in December 2019. From that day onwards we are facing this issue in the present as well. Different rules have been imposed by the government to reduce the spread of this disease. One main important rule is to wear mask every time if you are in a public area. It has been more nearly two years after the outbreak of the virus.

So, organizations, institutes and workplaces also suffer deeply because of this situation. And they can not just sit and wait for the government to do all the things. If they want to be successful during this global pandemic, they need to make some changes to how they ran the operations before the covid outbreak

So, in this project we will be discussing about the solution for an issue at a university due to the corona virus. On-Campus lectures will be starting in next month in this university. A study led by Japanese researchers found masks block about 60% of the amount of virus that comes out of an infected person. Wearing the mask always will the reduce chance of getting the corona virus (The Economic Times, 2020). So how can you check whether a student/lecturer is wearing a mask before he /she enters to the university premises. For that the university needs an automated system to check for masks before entering to the university.

1.2 Problem Statement

There are number of benefits of having an automated system to check for masks. One of the main benefits is that we can check whether a Student/Lecturer is wearing mask at rapid rate. Therefore, we can save the time that will be taken if we use a manual method. Some students may skip this search somehow due to some reasons if you are using eyes as the mask detection mechanism. You can never be sure if all the lecturers/students are wearing masks. We need to overcome this issue as well. So, when we investigated some of the systems available in the university, we have found that, there is no automated way for marking the attendance of students. The attendance has been marked manually by this university. Therefore, we have come up with a plan to overcome both these problem

1.3 Aim and Objectives

1.3.1 Aim

Considering the above-mentioned problems, the aim is to develop an automated Attendance System with the mask detection feature up front. By implementing this solution, we can check for masks at a rapid rate and no student will be able skip the search and finally attendance marking will be automated also.

1.3.2 Objectives

The approach that this document suggest is a mask detector with an automated attendance system. The initial step is to develop a mask detector model using an algorithm. It will not use any morphed masked images dataset; the model will be accurate. After the mask detector we need to develop the attendance marking and management process. For that we will be using firebase as the backend and JavaScript as the front end. This project does not only focus on detecting the mask but focuses on developing a fully automated attendance system with the additional feature of detecting whether student/lecturer is wearing a mask.

Main objectives of this document are,

- Developing a mask detector which uses live feed to detect masks.
- Developing a system and training the model to identify the mask accurately.
- Developing a server connected to a database to store and process data.
- Integrate the mask detection system with the attendance system

To mark the attendance student/lecturer is provided with a RFID card for each person. If the system identifies whether the student/lecturer is wearing a mask, system permits the student/lecturer to input the RFID card to mark the attendance. Then admin can view all the attendance of students and lecturers in the admin panel.

1.4 Solution Scope

The scope of this project is narrowed down to mainly implementing the mask detection system and attendance marking and attendance management systems. As the focus of this project is to only develop that.

The proposed system has two main stages. Extracting info from student/lecturer's face whether he/she is wearing a mask and extracting info from RFID card to mark the attendance.

Chapter 2 – Literature Review

Taking precautions is the best way of surviving a pandemic. The latest pandemic the whole world is facing is the corona virus. More than 100 countries are suffering from this virus and more than 150,000,000 people have died from this. There is no proper cure for this. So, the best way to avoid this pandemic is to follow precautionary methods. Some of them are using Sanitizers and wearing mask. These methods have become more than precautionary methods. Government has made wearing the mask in public a rule.

So, universities and educational institutes also need to follow these rules and cooperate the government. One of the main problems that the universities face is how to check for masks from each student or staff member when entering the university premises. To overcome this problem, we can think of few solutions. One of the solutions is to check each student and staff member at the entrance for masks by the security personnel. It will be an overwhelming task for the security to manually check each student. What if this process can be automated? It would be a solution that will release a headache from the university administration.

So, let us dive further into how to make this solution possible. In the next part of this document, we will be discussing the steps that we have taken to automate the mask checking process. There is no automated way to mark the attendance of the students for many universities in Sri Lanka. Calling the name of a student is the method most of the universities use. So, by looking at that problem and the mask checking problem we have come up with a solution to overcome both these issues. An automated attendance system with mask detection feature is the solution.

If you look locally, APIIT, SLIIT uses the "Calling the name of the student" method to mark the attendance and there is no system detect the masks. The threat that comes with this is huge. What if the security misses one student that did not wear the mask and he is also a covid patient? Thousands will be in danger and will cause chaos to whole the country. So, taking precautions is a must currently. Before the corona virus there were different attendance systems which used face detection or fingerprint to overcome the traditional attendance methods. One of the main concerns with these automated systems is that we can never be sure. But that can not be the case with regarding attendance. Student life and future of a student depends on the attendance. Many universities do not allow students to sit exams if their attendance is low. So, what if our system makes a mistake and that results in not been eligible for the exam. Lot of issues comes with that. So, we need to make an accurate attendance system to overcome that issue.

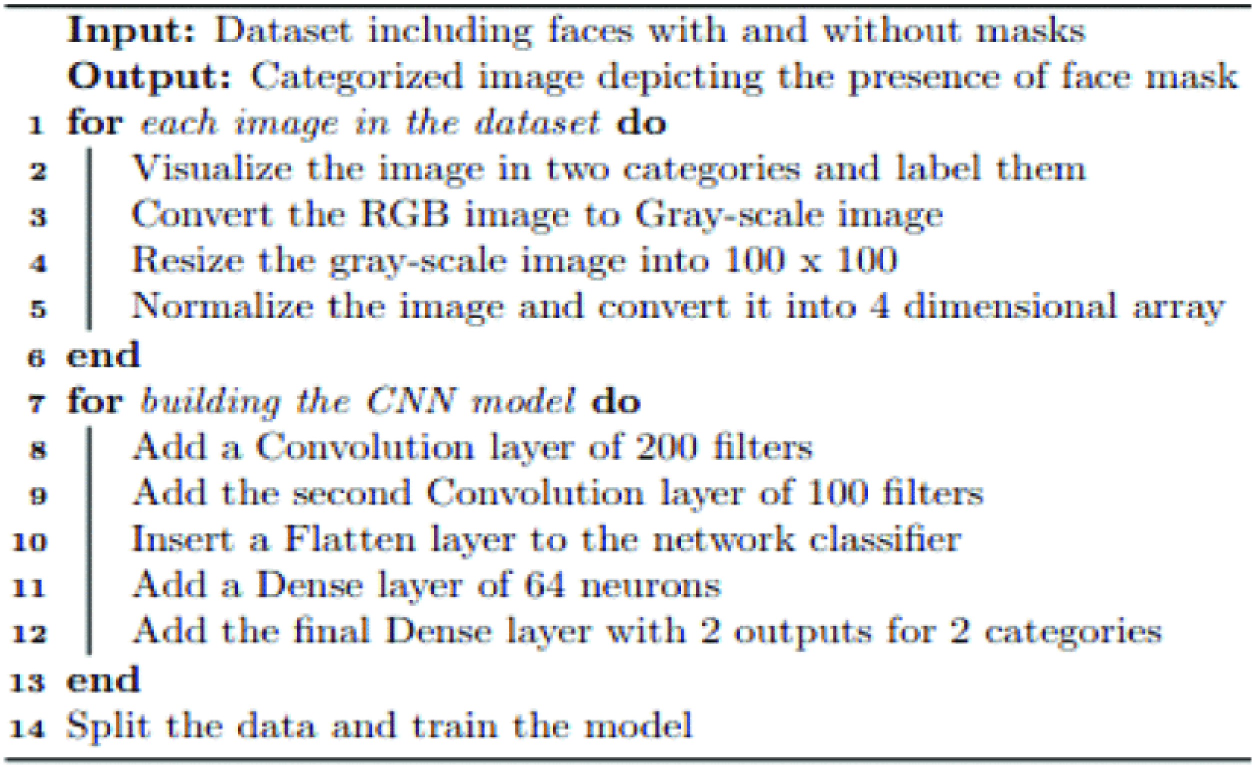
Some people might think detecting a mask and detecting a face is similar. But it is not the case. Attendance systems with face recognition uses a student's face. To recognize a student's face the system must first take and save a picture of the student as a reference in a database. During the attendance check, the web camera takes face pictures for a student to be recognized, and then the computer automatically detects the face and identifies a student name who most likely matches the pictures, and finally an excel file will be updated for attendance record based on the face recognition results (Weidong Kuang, 2020). In this system, a haar cascade model is used to detect faces from the web camera. A FaceNet, which has been trained by minimizing the triplet loss, is used to generate a 128- dimensional encoding for a face image. The similarity between the encodings of two face images determines whether the two face images coming from the same students. The system has been used for a class, and the results are very satisfactory (Weidong Kuang, 2020). So, in this face detection attendance system facenet and haar cascade algorithms are used. There are different varieties of ways of face recognition. Following are some Algorithms and their way of detecting the face and their differences.

Algorithm	Method	Difference
Haar Cascade	uses edge or line detection features	Haar Features have to be determined manually, there is a certain limit to the types of things it can detect. If you give classifier (a network, or any algorithm that detects faces) edge and line features, then it will only be able to detect objects with clear edges and lines
Principal Component Analysis	It carries out orthogonal transformation according to the features that have been censured to eliminate the correlation between original vectors. Characteristic values of the vectors descend in turn, namely characteristic face.	PCA deals with the data in its entirety for the principal components analysis without paying any particular attention to the underlying class structure
Linear Discriminate Analysis (LDA)	This is also a common approach, which searches for the best projection, using Fisher criterion function derivation to get extreme vector. Then, the samples are projected based on the vector, which minimizes the within-	LDA deals directly with discrimination between classes

	class scatter, maximizes between-class scatter in the way of classification but without losing useful information	

So, these are some algorithms that have been used in Face Detection systems. Same steps will be followed for detecting a mask as well. In face detection we use an algorithm to detect the faces and then we train a dataset and make the model and then we can detect faces instantaneously. That will be the approach for mask detection as well.

I have chosen Keras, Tensorflow, MobileNet and OpenCV in Python language as the incorporated packages that will help me to build this program. The method that I have used consists of a cascade classifier and a pre-trained CNN (convolutional neural network) which contains two 2D convolution layers connected to layers of dense neurons. The algorithm for face mask detection is as follows:



```
Input: Dataset including faces with and without masks
Output: Categorized image depicting the presence of face mask
for each image in the dataset do
    Visualize the image in two categories and label them
    Convert the RGB image to Gray-scale image
    Resize the gray-scale image into 100 x 100
    Normalize the image and convert it into 4 dimensional array
end
for building the CNN model do
    Add a Convolution layer of 200 filters
    Add the second Convolution layer of 100 filters
    Insert a Flatten layer to the network classifier
    Add a Dense layer of 64 neurons
    Add the final Dense layer with 2 outputs for 2 categories
end
Split the data and train the model
```

Figure 1: Algorithm

So, detecting the mask is only the first step in this system. Attendance marking is also a major part in this solution. I have chosen inputting a RFID card as the method of marking the attendance. Detecting the mask will be the pass for inputting the RFID card. Only if you wear mask as the next step you can input the RFID card and mark yourself present. Then the admin can manage the students and staff members after that.

There are few reasons for why I chose Keras, TensorFlow, MobileNet and OpenCV as the packages for this solution. I had start from the beginning with neural networks. So, I choosed Keras as the framework for working with neural networks because it has a simple architecture making it more readable and easy to use. Keras is a high-level framework that hides the backend computation and allows us to quickly build a neural network mode (Data Flair, 2021). Advantages of keras are

1. User-Friendly and Fast Deployment - Keras is a user-friendly API and it is very easy to create neural network models with Keras. It is good for implementing deep learning algorithms and natural language processing. We can build a neural network model in just a few lines of code (Data Flair, 2021).
2. Quality Documentation and Large Community Support
3. Multiple Backend and Modularity - Keras provides multiple backend support, where TensorFlow, Theano and CNTK being the most common backends. We can choose any of these backends according to the needs for different projects (Data Flair, 2021).
   We can also train the Keras model on one backend and test its results on other. It is very easy to change a backend in Keras, you just must write the name of the backend in the configuration file (Data Flair, 2021).
4. Pretrained models - Keras provides some deep learning models with their pre-trained weights. We can use these models directly for making predictions or feature extraction. These models have built-in weights, these weights are the results of training the model .
   Some pretrained models are - InceptionResNetV2 , MobileNet ,MobileNetV2,DenseNet
   NASNet

As you can Keras is the foundation of this program. We use Tensorflow as the backend for keras and MobileNetV2 as the pretrained model in this solution. There are different frameworks in usage in the modern world. One of the main framework is Pytorch. Down below is a comparison of the two different libraries.

S.No	Keras	PyTorch
1.	Keras was released in March 2015.	While PyTorch was released in October 2016.
2.	Keras has a high level API.	While PyTorch has a low level API.
3.	Keras is comparatively slower in speed.	While PyTorch has a higher speed than Keras, suitable for high performance.
4.	Keras has a simple architecture,making it more readable and easy to use.	While PyTorch has very low readablility due to a complex architecture.
5.	Keras has a smaller community support.	While PyTorch has a stronger community support.

S.No	Keras	PyTorch
6.	Keras is mostly used for small datasets due to its slow speed.	While PyTorch is preferred for large datasets and high performance.
7.	Debugging in Keras is difficult due to presence of computational junk.	While debugging in PyTorch is easier and faster.
8.	Keras provides static computation graphs.	While PyTorch provides dynamic computation graphs.
9.	Backend for Keras include:TensorFlow, Theano and Microsoft CNTK backend.	While PyTorch has no backend implementation.

As you can see you can have a backend in Keras and PyTorch doesn't support a backend implementation. This is also another main reason for why I chose keras over PyTorch.

Then when comes to TensorFlow, the main reason I chose TensorFlow over any other is the keras friendly features TensorFlow has. TensorFlow has compatibility with Keras, which allows its users to code some high-level functionality sections in it. Keras provides system-specific functionality to TensorFlow, such as pipelining, estimators, and eager execution. The Keras functional API supports a variety of topologies with different combinations of inputs, output, and layers (Techvidvan, n.d.). So the first choice was going with TensorFlow as the backend along with Keras.

After that OpenCV became the choice of library for differentiate and recognize faces and recognize objects in this case the mask. OpenCV library helps to group movements in recordings, trace progressive modules, follow eye gesture, track camera actions, expel red eyes from pictures taken utilizing flash, find comparative pictures from an image database, perceive landscape, and set up markers to overlay it with increased reality and so forth (Arjya Das, 2021). This solution makes use of these features in resizing and color conversion of data images. As I chose Python as the language for the mask detector it is very easy to code in python using OpenCV and it is a python wrapper. As we are doing image processing in this solution OpenCV has more than 400 free image processing functions. Those were reasons behind the choice of using OpenCV.

After the mask detection inserting a RFID card was the next step. The reason for Using RFID card as the attendance marking method is because it is easy to provide an RFID card each student and staff member. It cost around Rs.85 rupees in Sri Lanka. So not much money will be spent on RFIDs as well because it is cheap and can be found in any part in Sri Lanka. Finally by looking at the facts above we can build a system that reaches the goals and aims of the proposed method.

Chapter 3 – Requirement Specification

3.1 Functional Requirements

3.1.1 Student/Lecturer Management

Admins should be able to Register, Delete, Update and View Lecturers and Students. They can view specific details of each student's profiles also. Admins have the full authority in this system

The initial Details that required to register a student are:

- RFID card number
- Student ID
- Full Name
- DOB
- Address
- Email
- Batch
- Contact Number

The initial Details that required to register a lecturer are:

- RFID card number
- Staff ID
- Full Name
- DOB
- Address
- Email
- Contact Number

Only admins can register students and lecturers.

3.1.3 User Login

Admins can log in to the system through their given credentials. Students and lecturers can not log into the system they can only view their attendance percentage from the system.

3.1.2 Request Attendance

3.1.4 Mask Detection

The system should identify whether a Student/Lecturer is wearing a mask before entering to the premise. A machine learning model will be used for the object detection in the face. Then the results are passed into the server through an API.

3.1.5 RF card reading process

Every student/lecturer will be given a RF card. This process comes after passing the mask detection process. A RFID scanner will be used to scan this card. So when the student input the card, details will be read and passed in to backend. After that student or lecturer will be marked as present.

3.2 Non-Functional Requirements

1. **Speed-** Attendance Marking process should be fast. Otherwise, students and lecturers will not be happy.
2. **Accuracy** – Mask Detection model should be accurate. We cannot pass students and lecturers with no masks
3. **Security** – As personal information of students and lecturers are stored in this system, data must be highly secured.

\

Chapter 4- Methodology

4.1 Introduction

Down below will be an explanation of the methods that took to develop the mask detection attendance system. Steps that took to develop the Mask Detector and RFID card supported backend will be discussed.

4.2 System Architecture

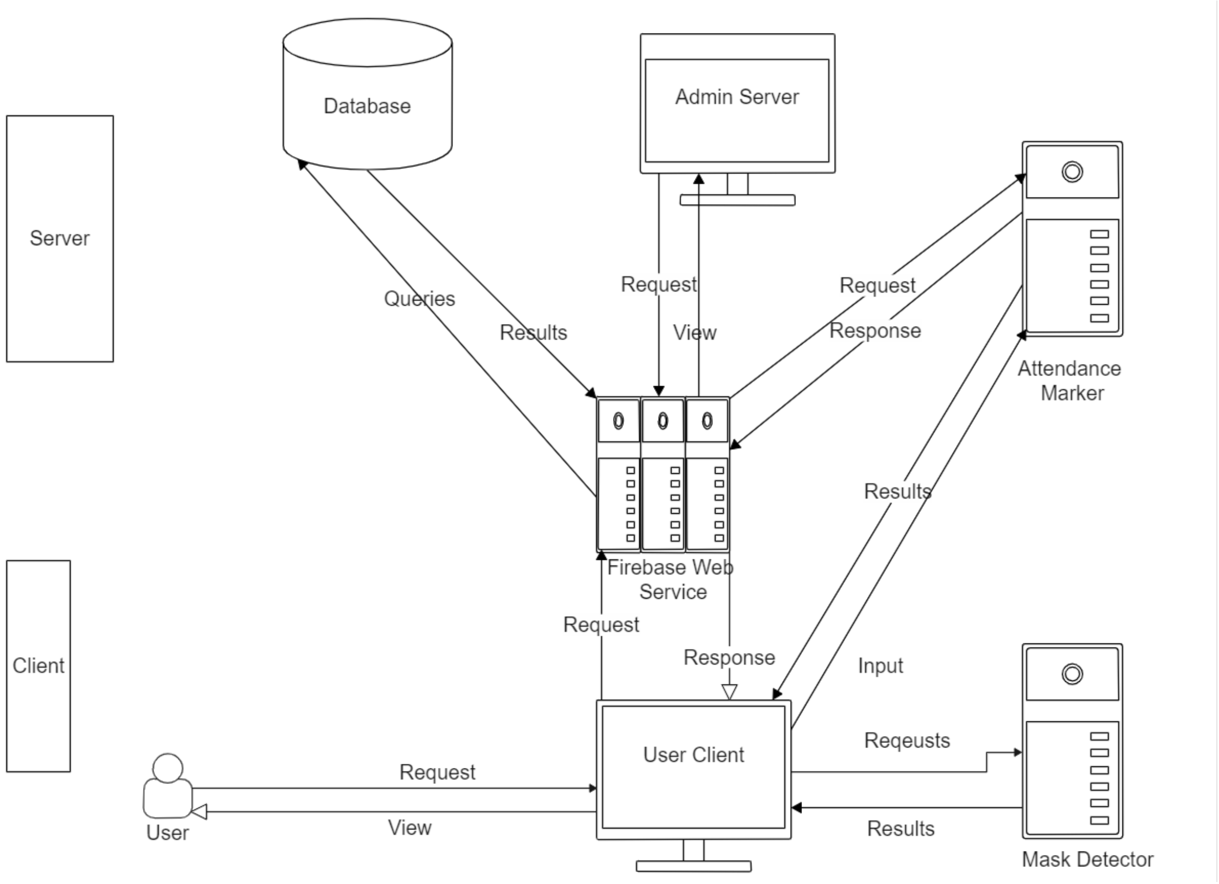


Figure 2:System Architecture

4.3 System Flow

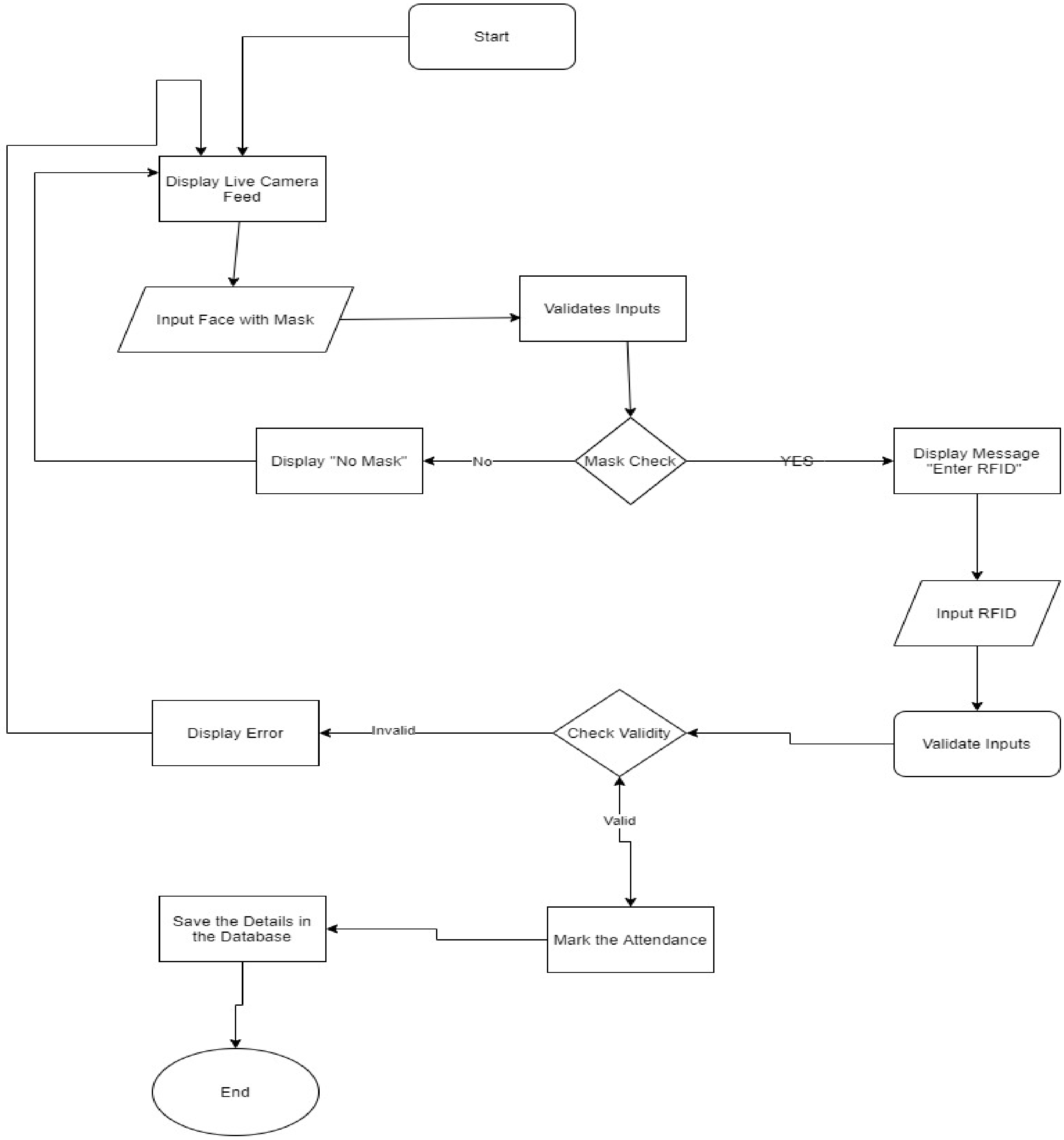
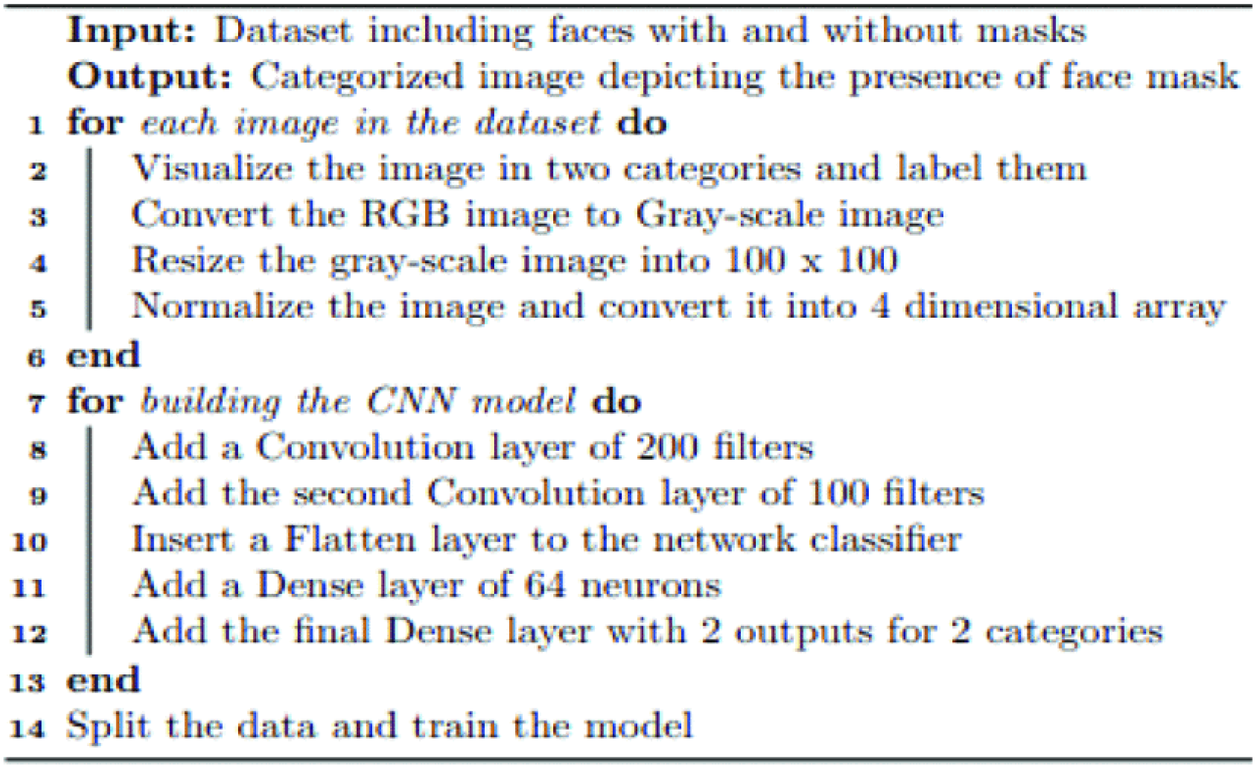


Figure 3: System Flow

4.4 The Mask Detector

This mask detector contains a cascade classifier and pre-trained convolutional neural network which has two 2D convolution layers connected to layers of numerous neurons. The algorithm for this was mentioned above in the Literature Review.



```
Input: Dataset including faces with and without masks
Output: Categorized image depicting the presence of face mask
for each image in the dataset do
    Visualize the image in two categories and label them
    Convert the RGB image to Gray-scale image
    Resize the gray-scale image into 100 x 100
    Normalize the image and convert it into 4 dimensional array
end
for building the CNN model do
    Add a Convolution layer of 200 filters
    Add the second Convolution layer of 100 filters
    Insert a Flatten layer to the network classifier
    Add a Dense layer of 64 neurons
    Add the final Dense layer with 2 outputs for 2 categories
end
Split the data and train the model
```

Figure 4: Algorithm in Methodology

First step of this is Data Pre-Processing. Data Pre-Processing involves conversion of data from a given format to much user friendly desired and meaningful format. It can be in any form like tables, images, videos, graphs, etc. This organized information fit in with an information model or composition and captures relationship between different entities. The proposed method deals with image and video data using NumPy and OpenCV.

In the data pre-processing part in our solution we change all the images we have in our folders the “with masks” and “without” masks folders into arrays. From those arrays we will be creating the deep learning model.

Next,

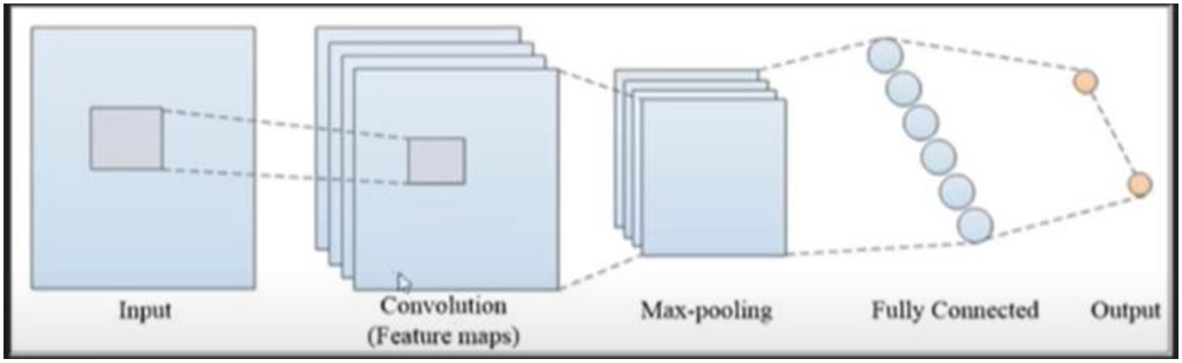


Figure 5: Mask Detector Model Process

we Train the model.

As the first step to our solution, we insert images from the folders. After the input image is processed as an array, we will send that to the mobilnet and then we do the Max-pooling and flatten it and create a fully connected layer and then get the output. As you can see in the traditional approach to image processing was to use Convolutional neural networks. But in this solution we have made a small change. Still we will be following the convolutional neural networks but we will be neglecting the convolution that happens in image processing and introducing mobilnets in place of that. Mobilnets are very faster in process than CNN and uses lesser parameters. But they are less accurate than CNNs. But we have taken steps to not to make that problem in this solution by having the learning rate a less value. That means we can calculate the loss properly. Which means we can get better accuracy soon.

By using mobilent we are generating two types of models. First one will be the mobilnet model of whose output will be putting into the normal model we are going to develop. We call that the head model and other model base model. First we will be creating the base model and then we will be crating the fully connected layer using pooling store. For that we will be creating a head model object in which we will be passing the base model output and then creates the pool and then flatten the layers after that dense layer will be added with 128 neurons. The output of this has 2 layers one for with masks and one for without.

Then we need to freeze the layers in the base model so they will not be updated during the first training process because they are just a replacement for convolutional neural networks. So after the training process we will be saving the model.

4.5 RFID card supported backend

All the server functions will be done in the backend. To use the backend we need to have a front end as well. Users are the ones who use the front end. React.js and material ui have been used for the front end. Then axios api to capture the data from the mask detector.

For the database we have used a nosql database firebase firestore. And firebase authentication have been used for login for the admin.

4.6 Software Development Methodology

For this solution we can use the waterfall model.

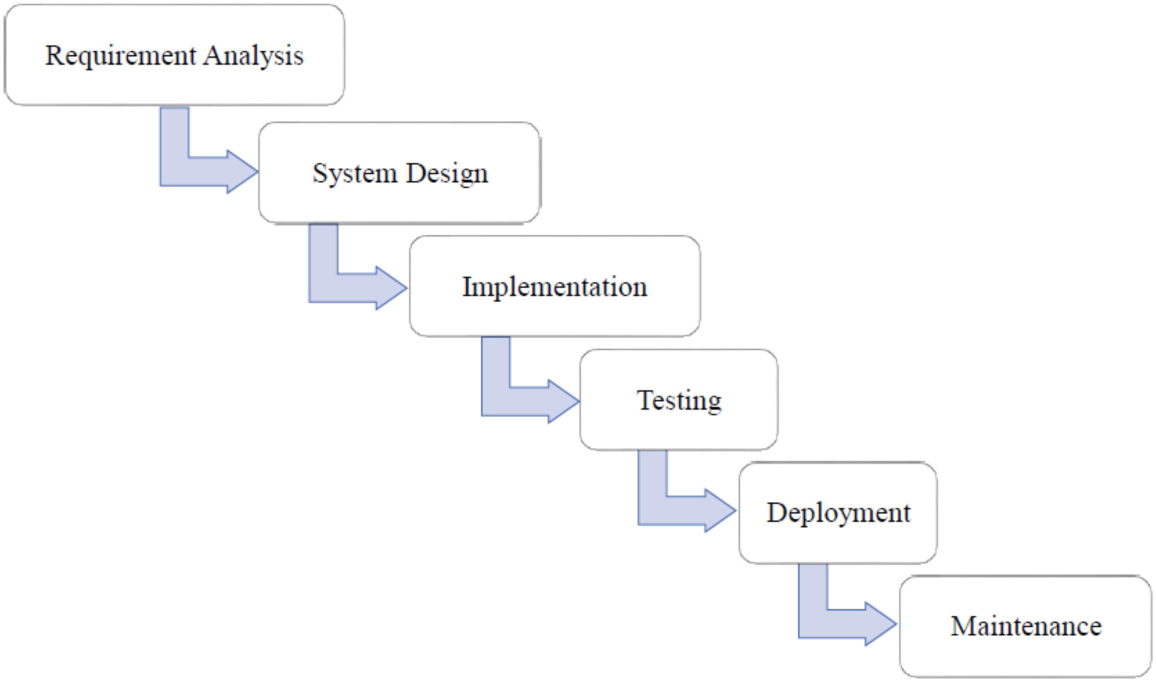


Figure 6: SDLC

If this process is followed each step must completed before moving into the next step. Overlapping of the steps is not allowed. Main steps of this are,

1. Requirement Gathering – Analysis of functional and non functional requirements are done. Before going to the next step requirements must be finalized and confirmed.
2. System Design- Structure of classes ,databases and high level functionality should be proposed using UML concepts
3. Implementation – Developing the proposed solution
4. Testing – Testing the developed solution
5. Deployment – Solution is ready to use
6. Maintenance – Fix the errors and improve the solution

The main advantage of this solution we can finish each task according to our deadline. Because we must complete each step before going to the next one. Main disadvantage of this methodology is that if any changes happen making those changes will be tough task.

Chapter 5 – Solution Concept

5.1 Component Diagram

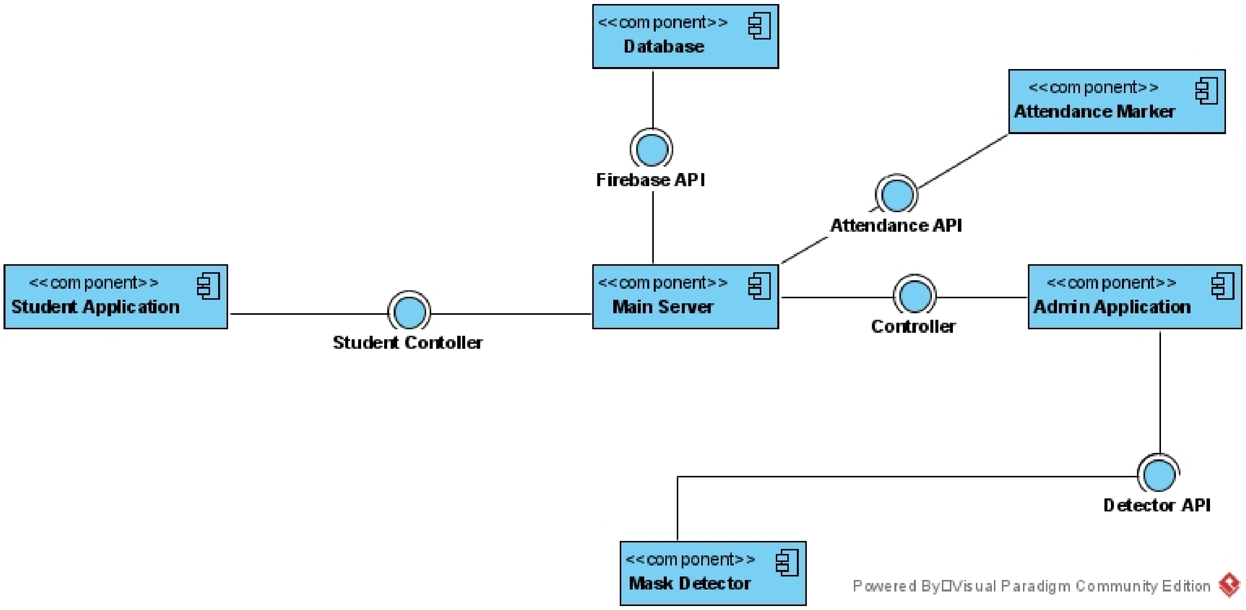


Figure 7: Component Diagram

Chapter 6- Requirement Analysis

6.1 Use Case diagram

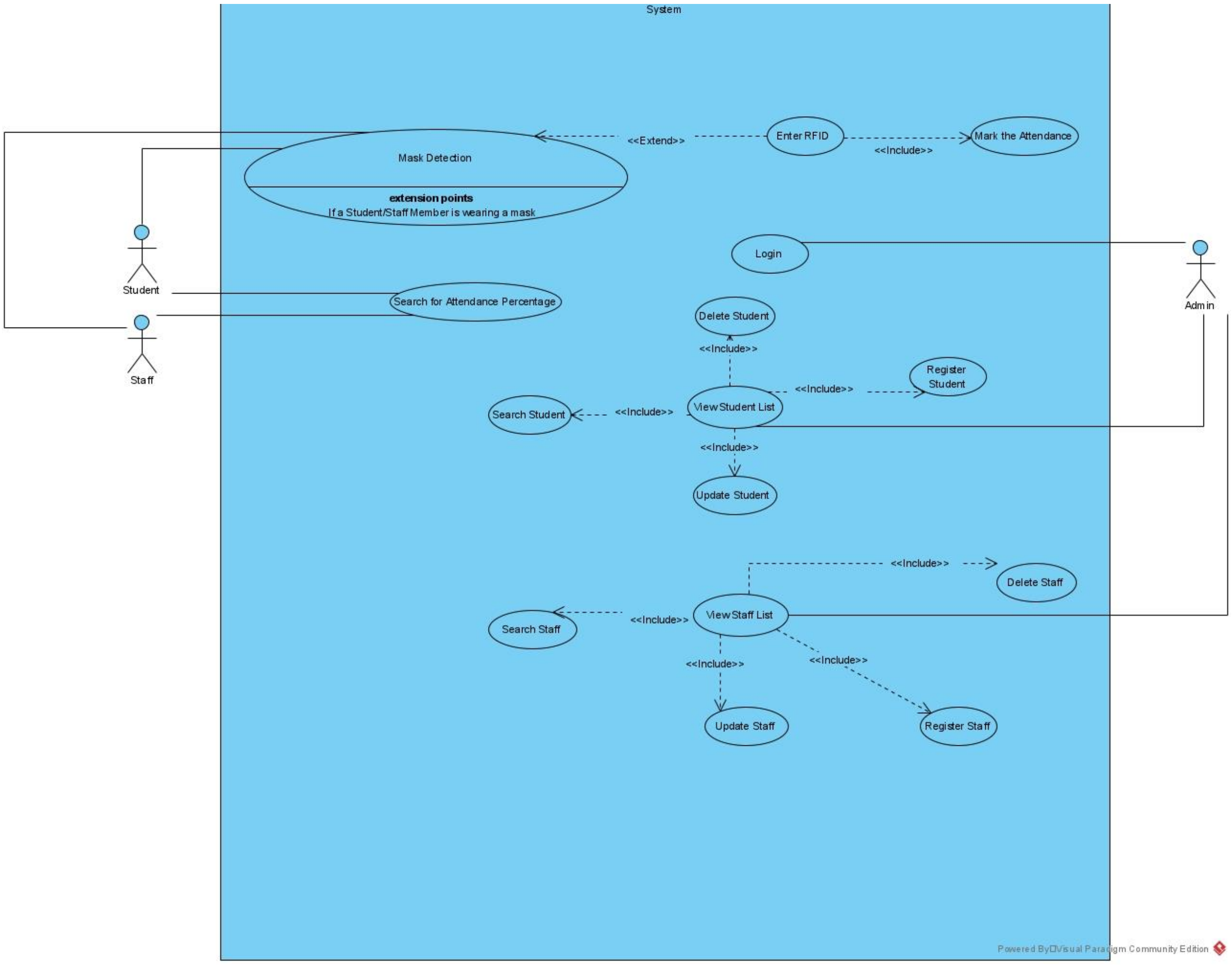


Figure 8: Use case Diagram

6.2 Use Case Scenarios

6.2.1 Mask Detection

Use Case Name	Mask Detection
Description	Masks should be detected from the Students and Staff
Primary Actor	Student/Staff (User)
Pre-conditions	Anyone can check if they are wearing the mask properly
Main Scenario	1. Use case begins when the user clicks “Attendance System Button” 2. System displays the live camera feed 3. User shows his face to the camera 4. System checks whether the user is wearing a mask 5. Prompt a message Saying Enter RFID card 6. User Enters the RFID card 7. System Validates the data 8. System marks the user as present 9. System saves the Details in the database 10. Prompt a message saying “Attendance Marked” 11. Use case ends when the user is Redirected to the Main Scenario 1
Alternatives	4a. System shows that the user is not wearing a mask 1. Not allowing the user to mark himself present without a mask 7a. Not a Valid Input 1. Redirect the User to Main Scenario 4
Post Condition	Attendance Should be marked for the relevant user

6.2.2 Request for Attendance

Use Case Name	Request for Attendance
Description	Students and Staff can request for their Attendance Percentage
Primary Actor	Student/Staff (User), Admin
Pre-conditions	Students should be registered into the system by admin
Main Scenario	1. Use case begins when the user enters the RFID 2. System displays the “Request Attendance” button 3. User clicks “Request Attendance” button 4. Front end passes the request details to the back end 5. System saves the request details 6. Admins checks for attendance request 7. System displays the Attendance request list 8. Use case ends when the admin sends the Attendance Percentage via email
Alternatives	
Post Condition	An email should be sent to the relevant user

6.2.3 View Student List

Use Case Name	View Student List
Description	Admin should be able to View the Students List and manage the students
Primary Actor	Admin
Pre-conditions	Admin should be logged into the system
Main Scenario	1. Use case starts when the admin clicks the Student List button 2. System views the Student List with Attendance Summary 3. Use case ends when the admin exits the Student List
Alternatives	
Post Condition	System should display the student list with Attendance

6.2.4 View Staff List

Use Case Name	View Staff List	
Description	Admin should be able to View the Staff List and manage the staff	
Primary Actor	Admin	
Pre-conditions	Admin should be logged into the system	
Main Scenario	1. Use case starts when the admin clicks the Staff List button 2. System views the Staff List with Attendance Summary 3. Use case ends when the admin exits the Staff List	
Alternatives		
Post Condition	System should display the staff list with Attendance	

6.2.5 Login

Use Case Name	Login
Description	Admin should be able to login to the system
Primary Actor	Admin
Pre-conditions	Admins should be included into the system
Main Scenario	1. Use case begins when the admin displays the login screen 2. Admin enters the Username and Password 3. Admin clicks on “Login” button 4. System validates entered fields 5. Admin sends login credentials to server. 6. Admin validates the entered credentials. 7. Use case ends when admin is directed to the admin dashboard
Alternatives	4a. Admin has left fields blank. 1. System prompts error message. 2. Redirect the admin to Main scenario 2. 6a. Admin credentials are invalid. 1. Application prompts error message. 2. Redirect the admin to Main scenario 2.
Post Condition	Admin should be able to manage the system

6.2.6 Register Student

Use Case Name	Register Student
Description	Admin should be able to register a student
Primary Actor	Admin
Pre-conditions	Admin should be logged into the system
Main Scenario	1. Use case begins when the admin clicks View Student List Button 2. System displays the Student List 3. Admin clicks the “Register Student” button 4. System displays the fields that the admin should enter details 5. Admin enters the details 6. Admin clicks “Register” button 7. System validates the details 8. Saves the details in the database 9. Use case ends when the system prompts “Successfully Registered” message
Alternatives	7a. Admin has left the fields blank 1. System prompts an error message 2. Redirect the user to Main Scenario 4
Post Condition	Student should be saved into the database

6.2.7 Update Student

Use Case Name	Update Student	
Description	Admin should be able to update a student	
Primary Actor	Admin	
Pre-conditions	Admin should be logged into the system	
Main Scenario	1. Use case begins when the admin clicks View Student List Button 2. System displays the Student List 3. Admin clicks the “Update Student” button 4. System displays the fields that the admin should enter details 5. Admin enters the details 6. Admin clicks “Update” button 7. System validates the details 8. Saves the new details in the database 9. Use case ends when the system prompts “Successfully Updated” message	
Alternatives	7a. Admin has left the fields blank 3. System prompts an error message 4. Redirect the user to Main Scenario 4	
Post Condition	New details should be saved into the system	

6.2.8 Delete Student

Use Case Name	Delete Student	
Description	Admin should be able to delete a student	
Primary Actor	Admin	
Pre-conditions	Admin should be logged into the system	
Main Scenario	1. Use case begins when the admin clicks View Student List Button 2. System displays the Student List 3. Admin clicks the “Delete Student” button 4. System displays the message “Are you sure or not” 5. Admin clicks “Yes” button 6. System checks for that student’s profile 7. System Delete that student’s record 8. Use case ends when the system prompts “Successfully Deleted” message	
Alternatives	5a. Admin clicks no 1. Redirect the admin to Main Scenario 2	
Post Condition	Relevant Student details should be deleted from the database	

6.2.9 Register Staff

Use Case Name	Register Staff	
Description	Admin should be able to register a staff	
Primary Actor	Admin	
Pre-conditions	Admin should be logged into the system	
Main Scenario	1. Use case begins when the admin clicks View Staff List Button 2. System displays the Staff List 3. Admin clicks the “Register Staff” button 4. System displays the fields that the admin should enter details 5. Admin enters the details 6. Admin clicks “Register” button 7. Front End validates the inputs 8. Passes the details to the backend 9. Validates the details 10. Saves the details in the database 11. Use case ends when the system prompts “Successfully Registered” message	
Alternatives	7a. Admin has left the fields blank 1. System prompts an error message 2. Redirect the user to Main Scenario 4 9a. Invalid Details 1. System prompts an error message 2. Redirect the user to Main Scenario 4	
Post Condition	Staff should be saved into the database	

6.2.10 Update Staff

Use Case Name	Update Staff	
Description	Admin should be able to update a staff member	
Primary Actor	Admin	
Pre-conditions	Admin should be logged into the system	
Main Scenario	1. Use case begins when the admin clicks View Staff List Button 2. System displays the Staff List 3. Admin clicks the “Update Staff” button 4. System displays the fields that the admin should enter details 5. Admin enters the details 6. Admin clicks “Update” button 7. System validates the details 8. Saves the new details in the database 9. Use case ends when the system prompts “Successfully Updated” message	
Alternatives	7a. Admin has left the fields blank 3. System prompts an error message 4. Redirect the user to Main Scenario 4	
Post Condition	New details should be saved into the system	

6.2.11 Delete Staff

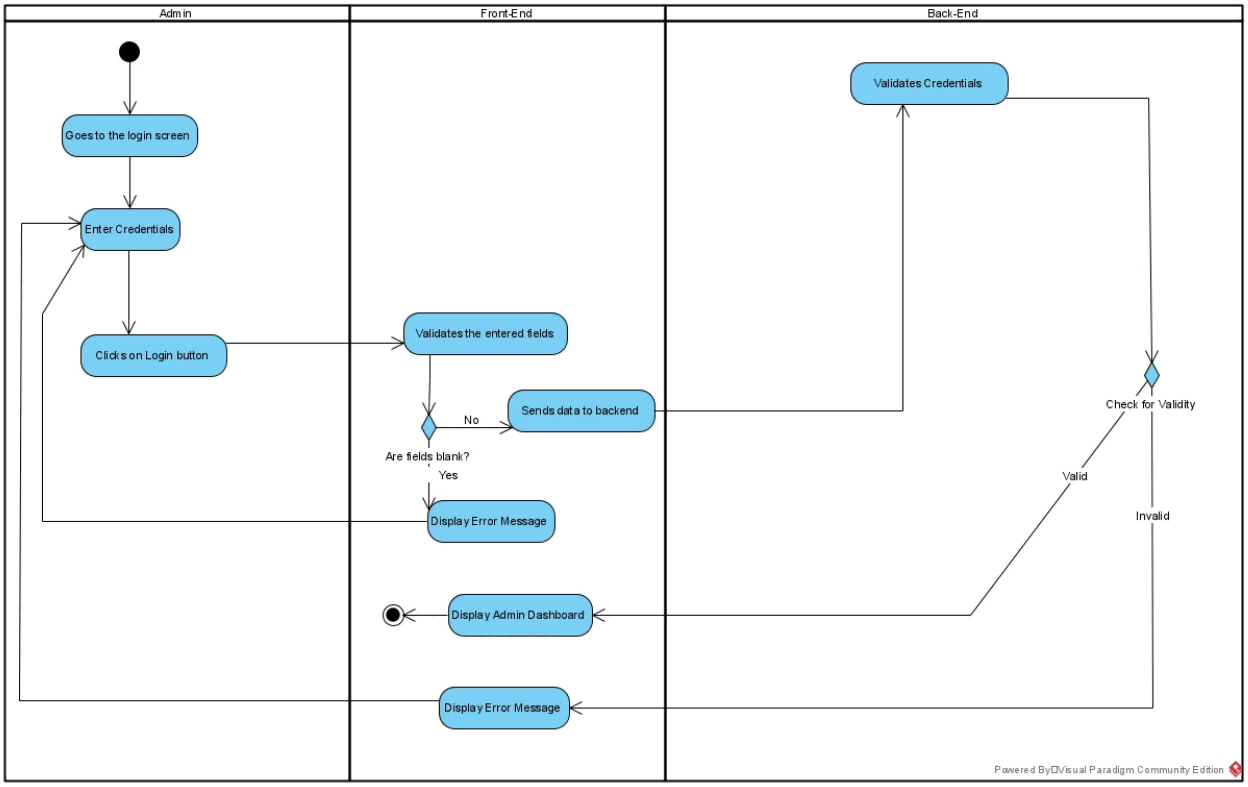
Use Case Name	Delete Staff	
Description	Admin should be able to delete a staff member	
Primary Actor	Admin	
Pre-conditions	Admin should be logged into the system	
Main Scenario	1. Use case begins when the admin clicks View Staff List Button 2. System displays the Staff List 3. Admin clicks the “Delete Staff” button 4. System displays the message “Are you sure or not” 5. Admin clicks “Yes” button 6. System Delete that staff member’s record 7. Delete those details from the database 8. Use case ends when the system prompts “Successfully Deleted” message	
Alternatives	5a. Admin clicks no 2. Redirect the admin to Main Scenario 2	
Post Condition	Relevant Staff details should be deleted from the database	

Chapter 7 – System Design

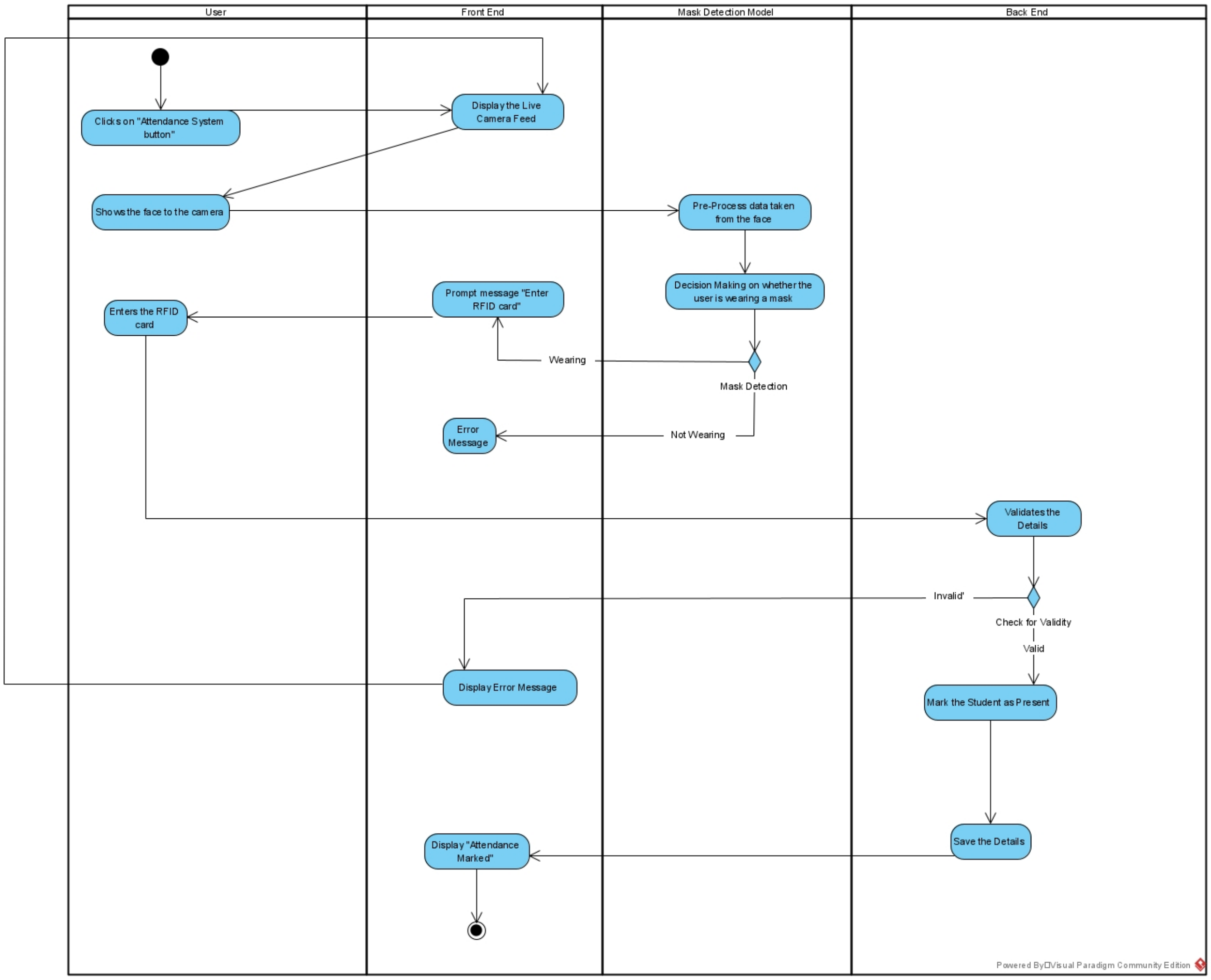
7.1 Class Diagrams

7.2 Activity Diagrams

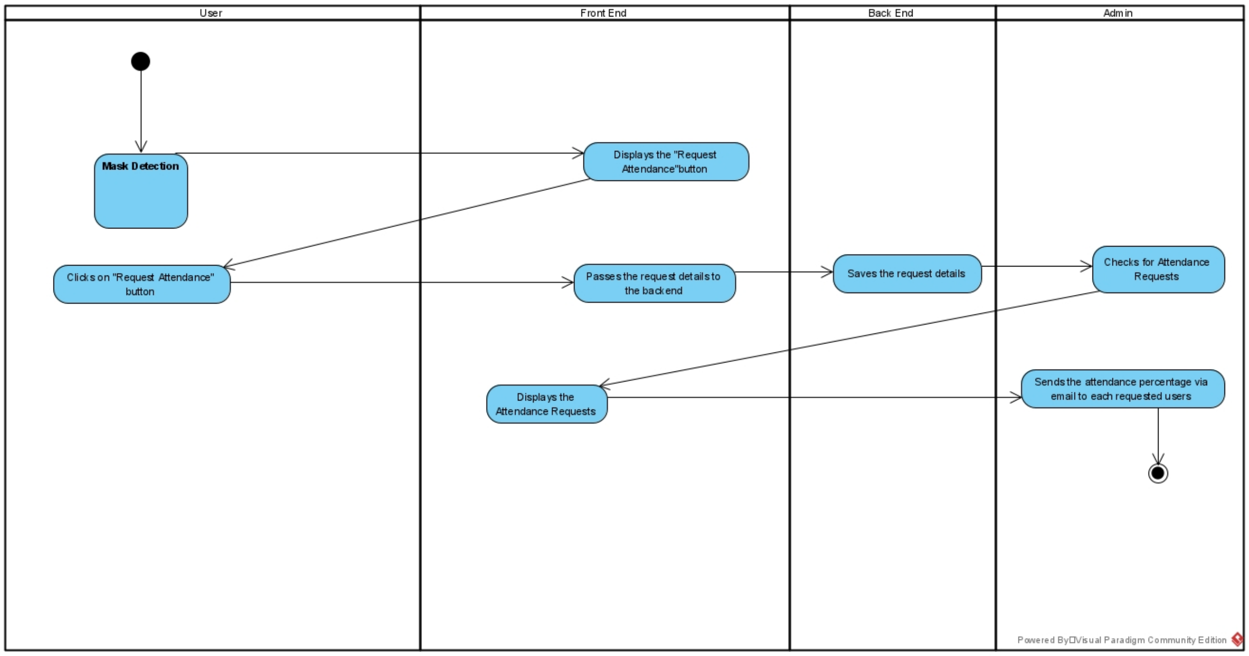
7.2.1 Login



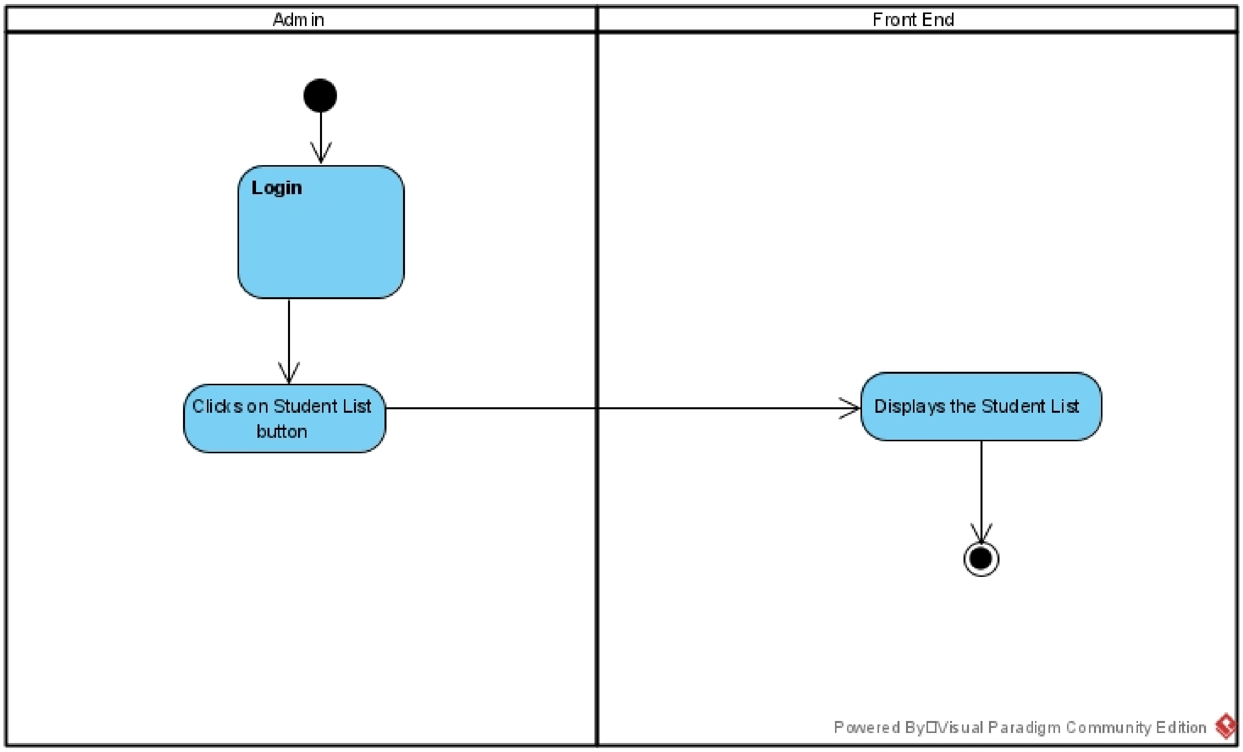
7.2.2 Mask Detection


User
Front End
Mask Detection Model
Back End
Clicks on "Attendance System button"
Display the Live Camera Feed
Shows the face to the camera
Pre-Process data taken from the face
Decision Making on whether the user is wearing a mask
Prompt message "Enter RFID card"
Enters the RFID card
Wearing
Mask Detection
Error Message
Not Wearing
Validates the Details
Invalid'
Check for Validity
Valid
Display Error Message
Mark the Student as Present
Save the Details
Display "Attendance Marked"
Powered By Visual Paradigm Community Edition

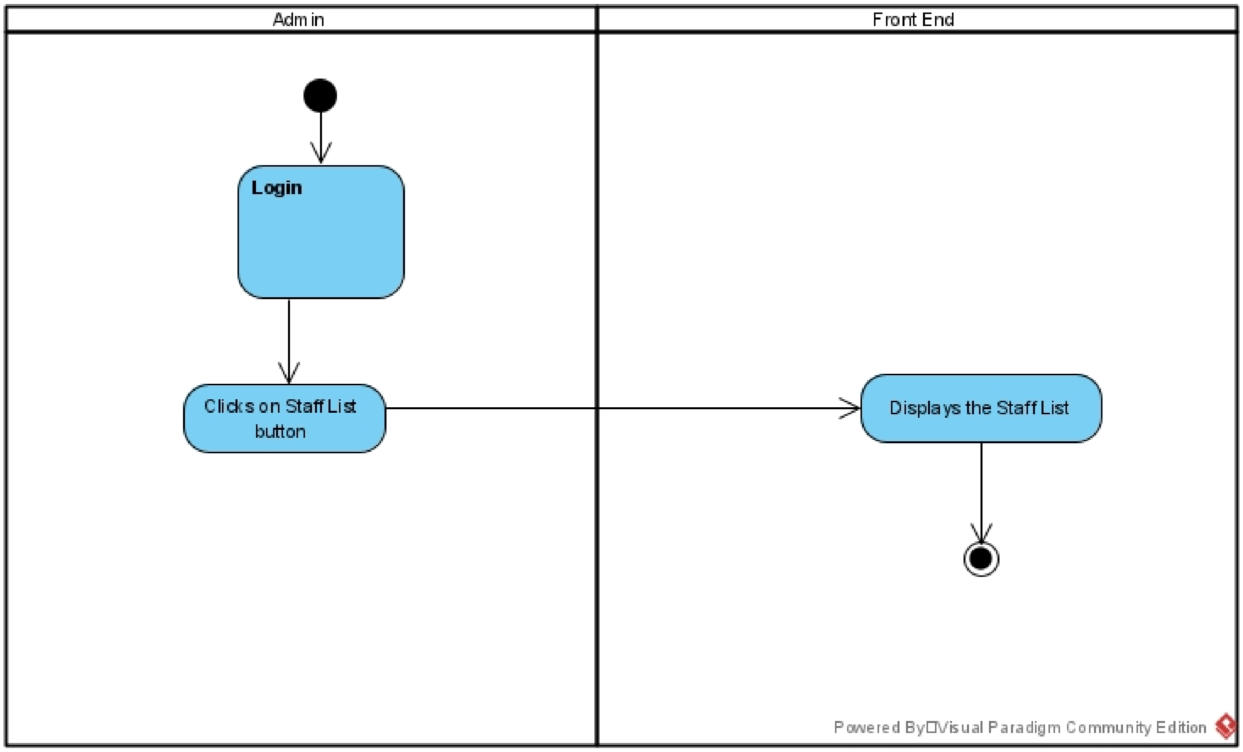
7.2.3 Request for Attendance


User
Front End
Back End
Admin
Mask Detection
Displays the "Request Attendance"button
Clicks on "Request Attendance" button
Passes the request details to the backend
Saves the request details
Checks for Attendance Requests
Displays the Attendance Requests
Sends the attendance percentage via email to each requested users
Powered By Visual Paradigm Community Edition

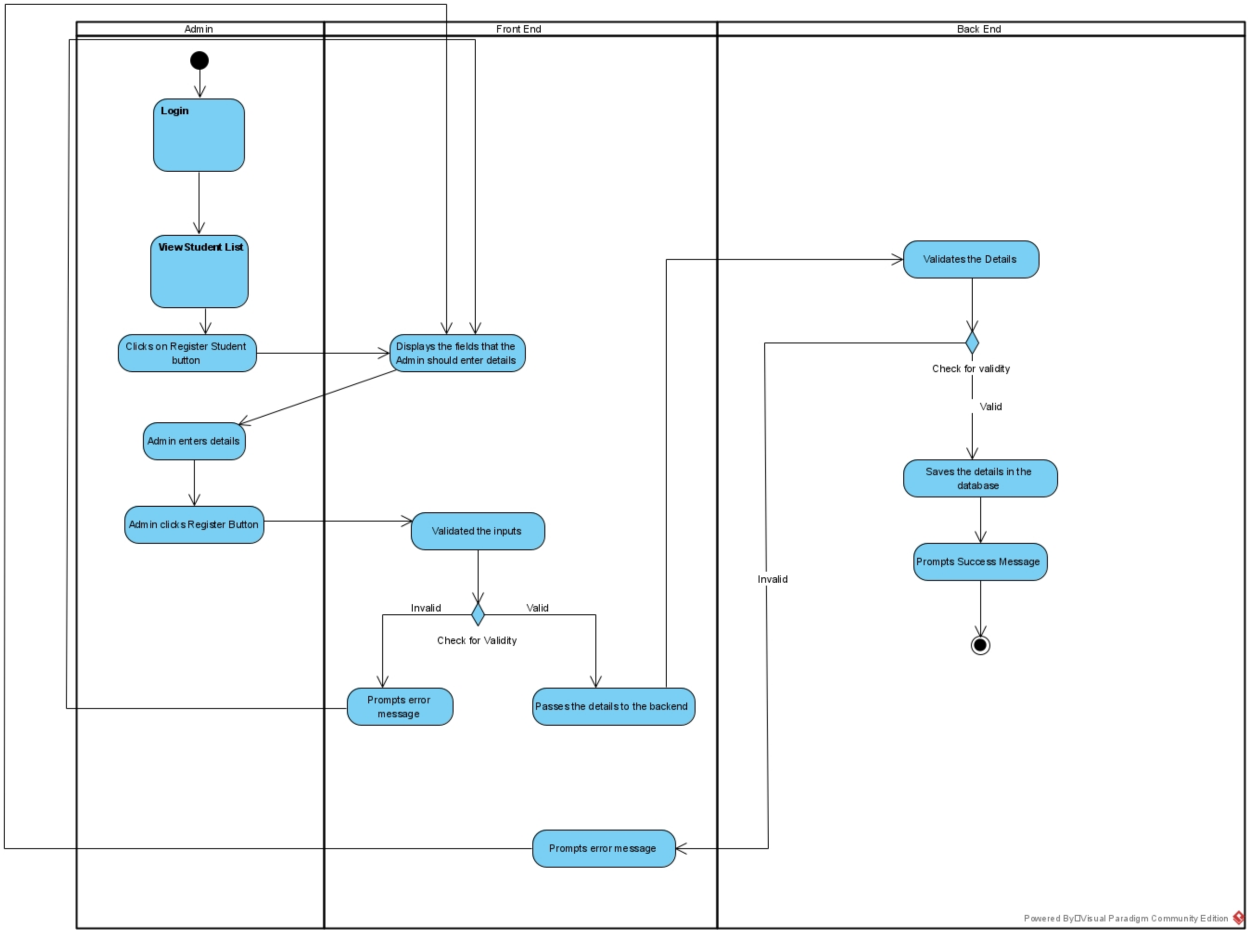
7.2.4 View Student List


Adm in
Front End
Login
Clicks on Student List button
Displays the Student List
Powered By Visual Paradigm Community Edition

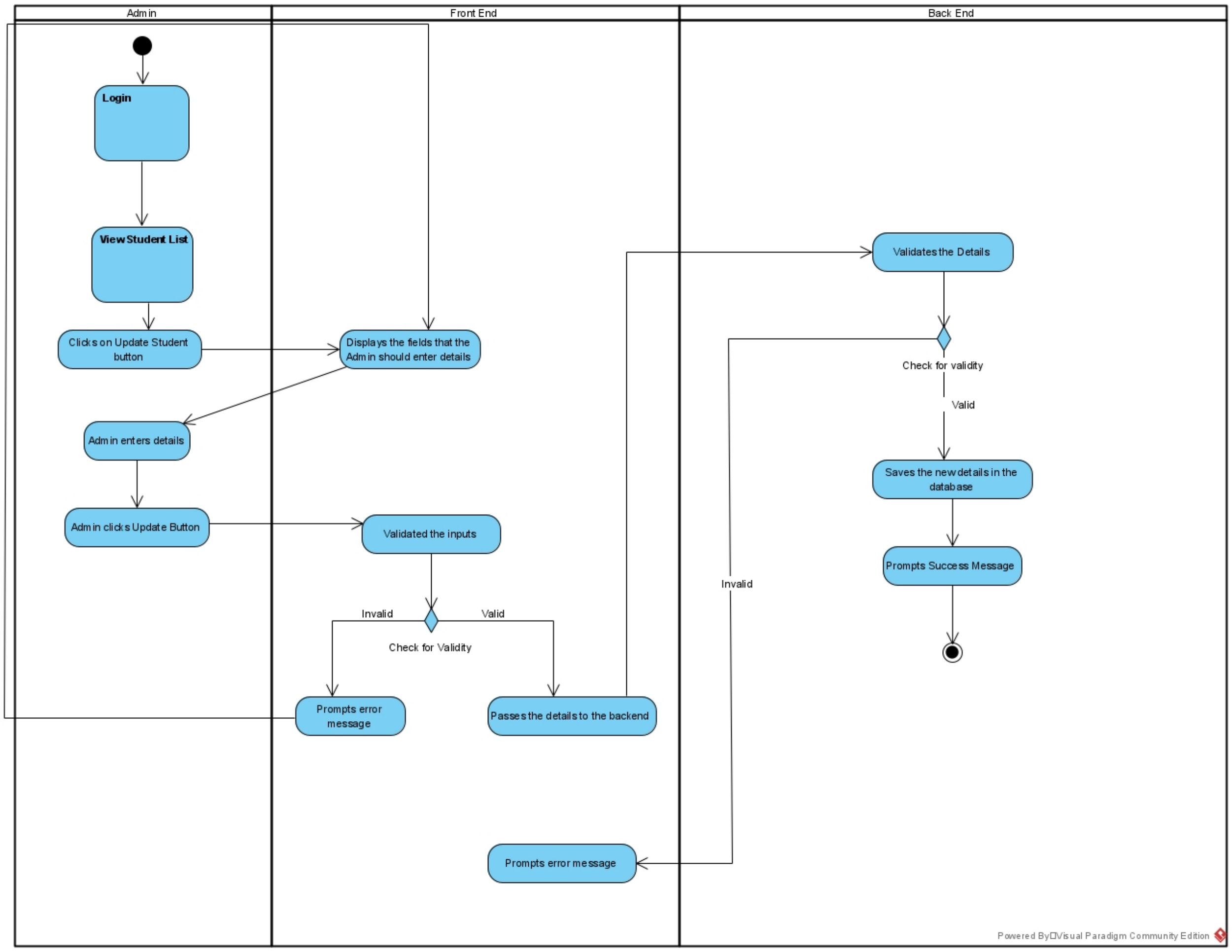
7.2.5 View Staff List


Admin
Front End
Login
Clicks on Staff List button
Displays the Staff List
Powered By Visual Paradigm Community Edition

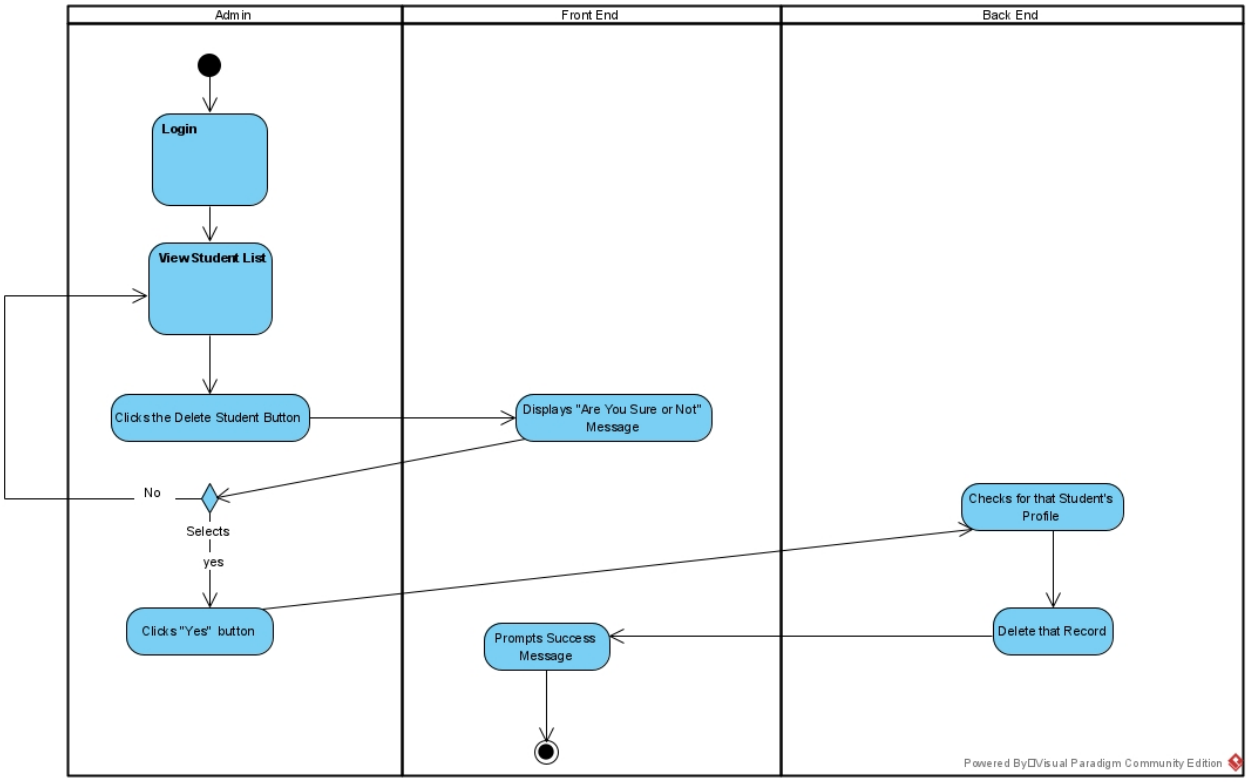
7.2.6 Register Student



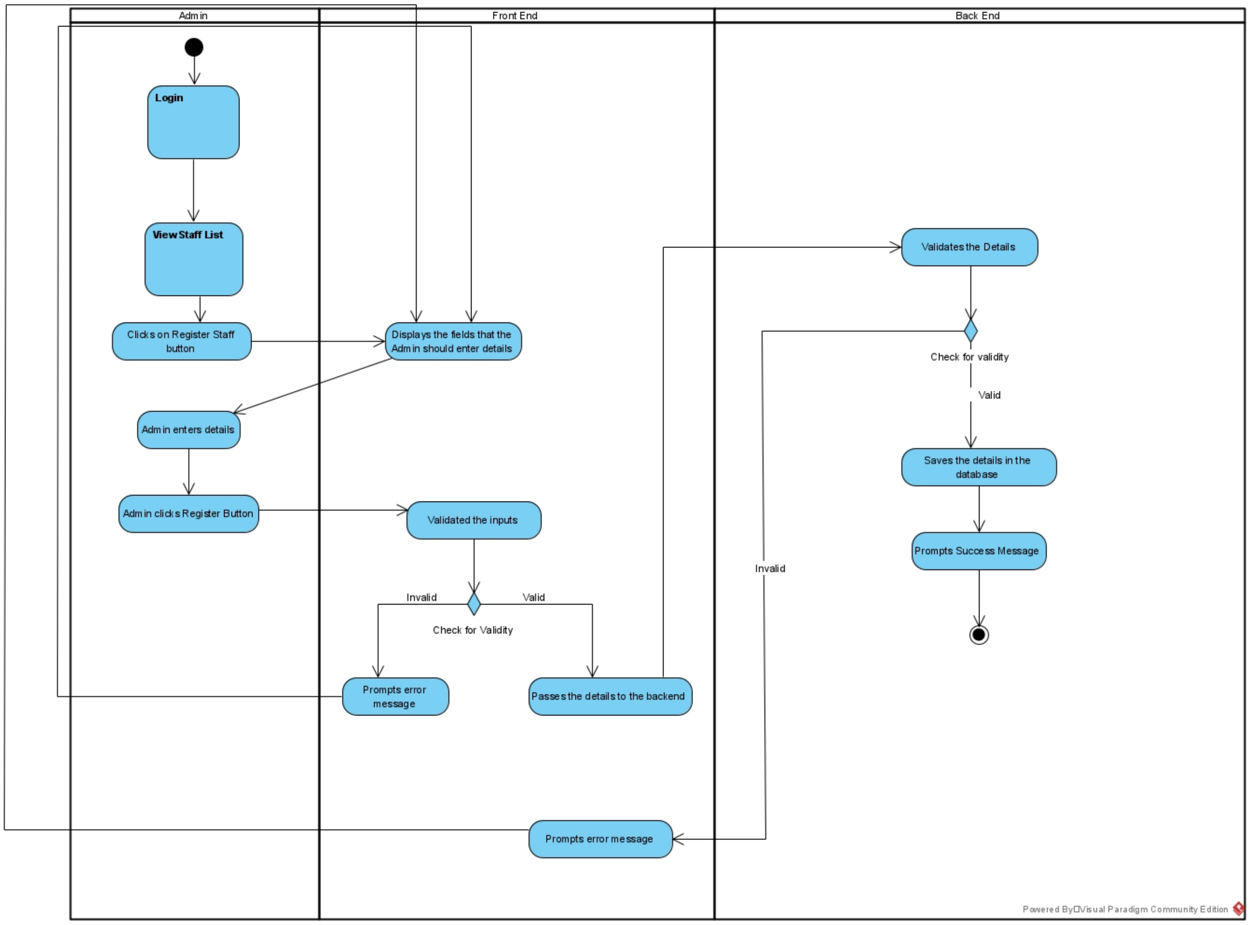
7.2.7 Update Student



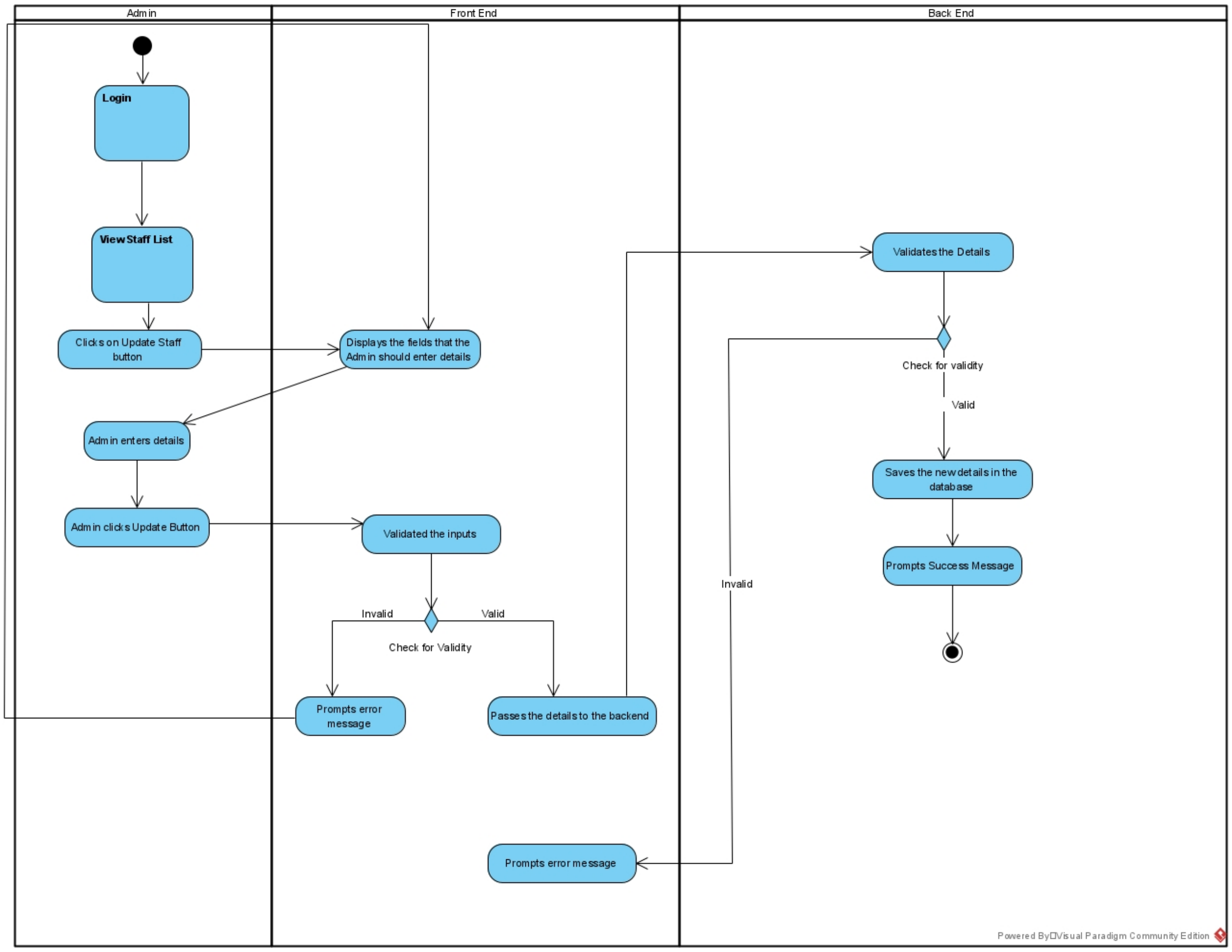
7.2.8 Delete Student


Admin
Front End
Back End
Login
View Student List
Clicks the Delete Student Button
Displays "Are You Sure or Not" Message
No
Selects
yes
Checks for that Student's Profile
Clicks "Yes" button
Delete that Record
Prompts Success Message
Powered By Visual Paradigm Community Edition

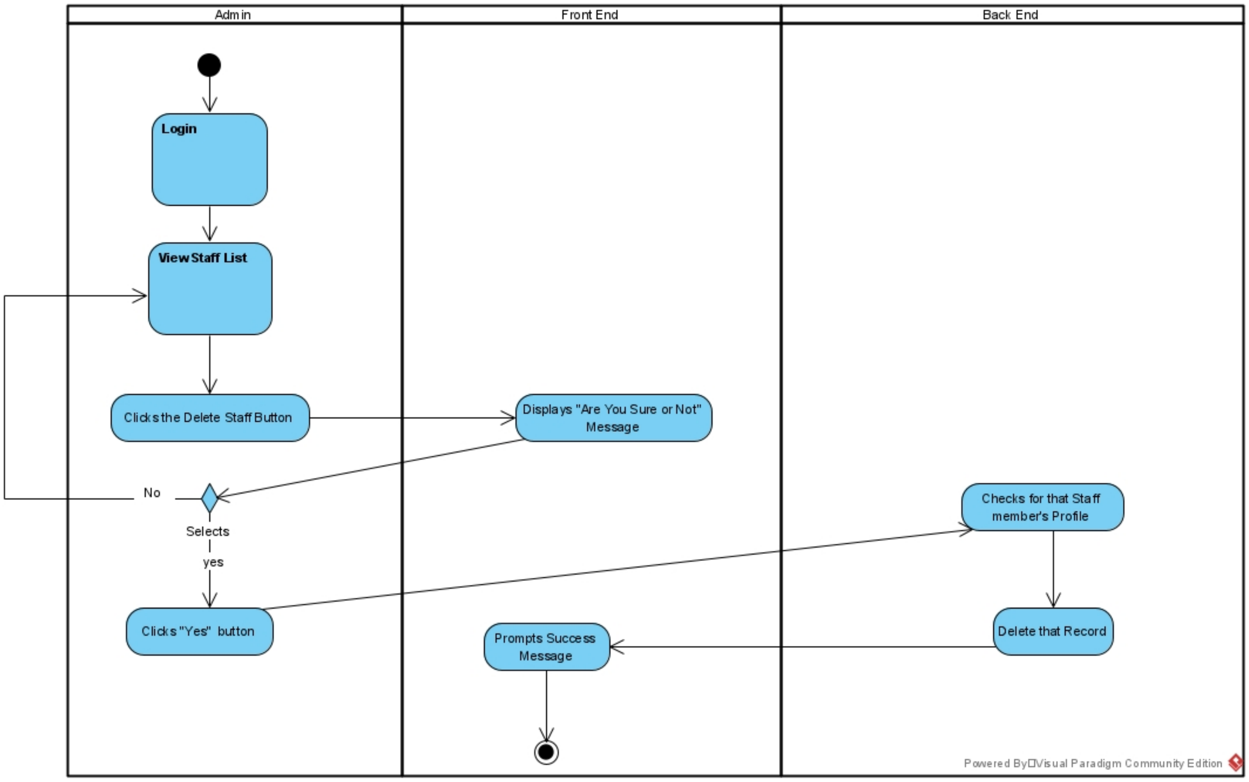
7.2.9 Register Staff



7.2.10 Update Staff



7.2.11 Delete Staff


Admin
Front End
Back End
Login
View Staff List
Clicks the Delete Staff Button
Displays "Are You Sure or Not" Message
No
Selects
yes
Clicks "Yes" button
Checks for that Staff member's Profile
Delete that Record
Prompts Success Message
Powered By Visual Paradigm Community Edition

Chapter 8 – Implementation

8.1 Hardware and Software Requirements

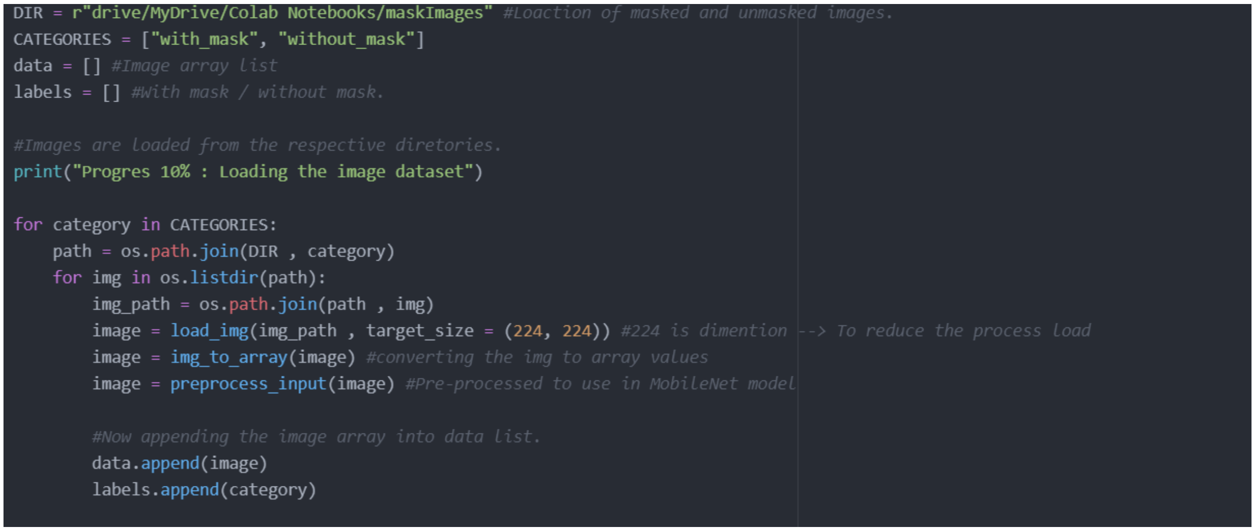
Hardware	**Minimum Requirement**
CPU	Intel Core i7-6200U up to 2.30GHZ
Memory (RAM)	8 GB.
GPU	4GB
Hard disk space	2GB

Software	**Requirement**
Operating System	Windows 10
Python	Version 3 or higher
PIP	Version 20.2 or higher
React.js	latest
Firebase	Latest
VS code	Latest version

8.2 Mask Detector

8.2.1 Data Pre- Processing

In here are we are going to convert all the images in our folders “with masks” dataset and “without mask” dataset into arrays. With those arrays we will be creating the deep learning model.

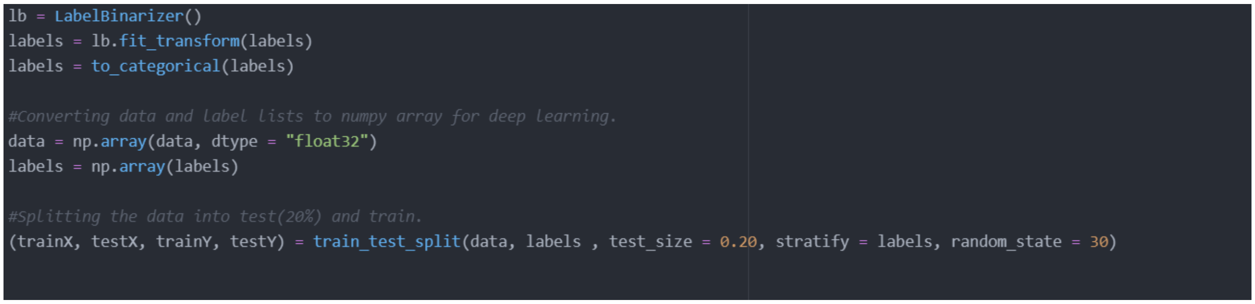


```python
DIR = r"drive/MyDrive/Colab Notebooks/maskImages" #Loaction of masked and unmasked images.
CATEGORIES = ["with_mask", "without_mask"]
data = [] #Image array list
labels = [] #With mask / without mask.

#Images are loaded from the respective diretories.
print("Progres 10% : Loading the image dataset")

for category in CATEGORIES:
    path = os.path.join(DIR , category)
    for img in os.listdir(path):
        img_path = os.path.join(path , img)
        image = load_img(img_path , target_size = (224, 224)) #224 is dimention --> To reduce the process load
        image = img_to_array(image) #converting the img to array values
        image = preprocess_input(image) #Pre-processed to use in MobileNet model

        #Now appending the image array into data list.
        data.append(image)
        labels.append(category)
```

Figure 9: Data Pre-Processing 1

DIR has the location of the folder of the dataset. And there are two categories mentioned because we have two datasets in those two folders. And we are going to loop through them to build our model.

And you can see there are two lists instantiated above. Data list contains all the image arrays. In the labels list we have the corresponding labels for the with masks and without masks images. In the loop method we will be looping through the two folders and covert those images into arrays and pre-process those arrays to use in mobilenet model and then finally we will append those arrays into lists.

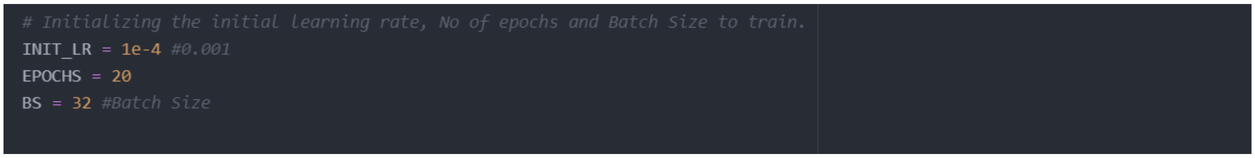


```python
lb = LabelBinarizer()
labels = lb.fit_transform(labels)
labels = to_categorical(labels)

#Converting data and label lists to numpy array for deep learning.
data = np.array(data, dtype = "float32")
labels = np.array(labels)

#Splitting the data into test(20%) and train.
(trainX, testX, trainY, testY) = train_test_split(data, labels , test_size = 0.20, stratify = labels, random_state = 30)
```

Figure 10 : Data Pre Processing 2

But those text characters in the labels are not very useful to us so will converting that to a readable array. So in order for doing that we have call the LabelBinarizer() method. With that we are going to change the “with mask” and without mask” labels into categorical variables. Basically, that means turning it into 0 and 1s. So after successfully converting those values into 0s and 1s we need to convert those into numpy arrays. To do that we are using np.array() method. Both the lists are now turned into numpy arrays. And then using train_test_split method we will be splitting the data into test and for train. In this case we have used 20% of the dataset for testing and 80% for training.

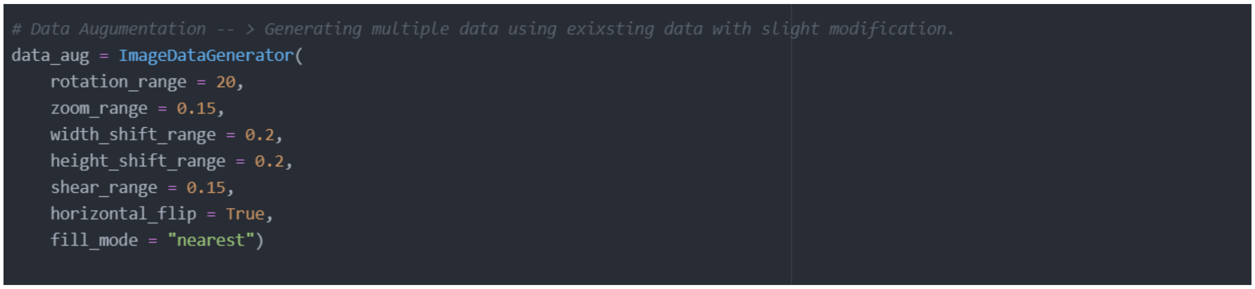
8.2.2 Training



```
# Initializing the initial learning rate, No of epochs and Batch Size to train.
INIT_LR = 1e-4 #0.001
EPOCHS = 20
BS = 32 #Batch Size
```

Figure 11 : Training 1

In here we have initialized the learning rate. And we have assigned our learning rate to a lesser value. Because when your learning rate is less our loss will be calculated properly. That means we can get the better accuracy very soon.

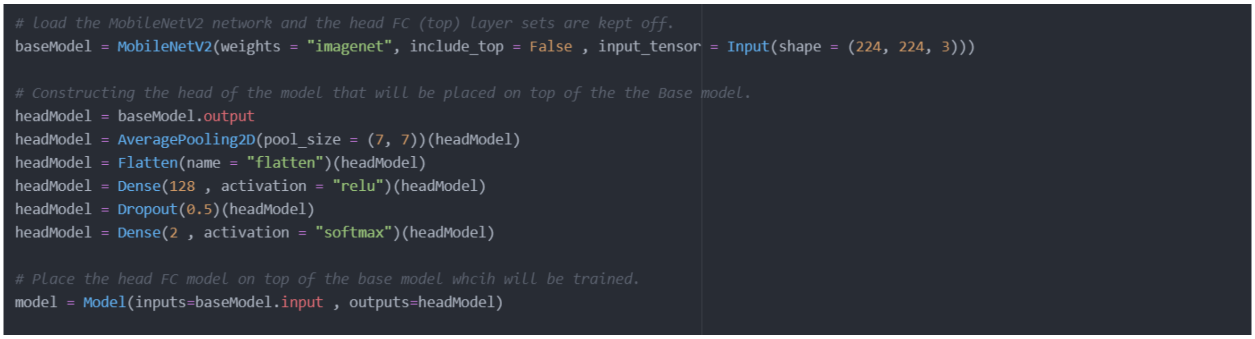


```
# Data Augumentation -- > Generating multiple data using exixsting data with slight modification.
data_aug = ImageDataGenerator(
    rotation_range = 20,
    zoom_range = 0.15,
    width_shift_range = 0.2,
    height_shift_range = 0.2,
    shear_range = 0.15,
    horizontal_flip = True,
    fill_mode = "nearest")
```

Figure 12: Training 2

Then another important function is the Image Data Generator . What ImageDataGenerator does is create data augmentation. It creates many images from a single image by adding various properties like flipping, rotating the image and many other properties.

We are using mobilenetv2 model for this solution and actually by using mobilnet we are creating two models. The first model will be the mobilnet model whose output will be passing into the normal module we will create. We call those two models as head model and base model.



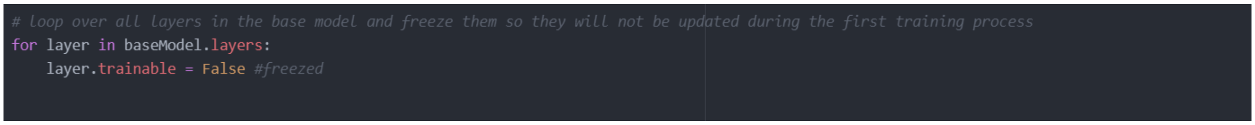
```
# load the MobileNetV2 network and the head FC (top) layer sets are kept off.
baseModel = MobileNetV2(weights = "imagenet", include_top = False , input_tensor = Input(shape = (224, 224, 3)))

# Constructing the head of the model that will be placed on top of the the Base model.
headModel = baseModel.output
headModel = AveragePooling2D(pool_size = (7, 7))(headModel)
headModel = Flatten(name = "flatten")(headModel)
headModel = Dense(128 , activation = "relu")(headModel)
headModel = Dropout(0.5)(headModel)
headModel = Dense(2 , activation = "softmax")(headModel)

# Place the head FC model on top of the base model whcih will be trained.
model = Model(inputs=baseModel.input , outputs=headModel)
```

Figure 13: Training 3

As stated above we are creating the base model using the MobileNetV2. There is an parameter called “weights=”imagenet”. Basically that means, weights which has been used for a pretrained model that is specially designed for images. So when we use that parameter it will give us better results. Then include_top = false is actually a Boolean value which is to say whether to include the fully connected layer at the top of our network. We have set that value to false because we are going to connect the fully connected layer by ourselves. Next we have use the input_tensor. Input_tensor is nothing but the shape of the image going through. We have set the image size as 224*224 (Height and Width) and 3 is the three channel values in that image. We are inputting colored images here. And colored images have three channels which is RGB(Red, Green, Blue). So that tells why the value 3 is used.

Once the base model is done we need to create the fully connected layer . For that we will be creating a headmodel object and we will be passing the base model output as the first parameter. And the pool size is 7*7 . Then we will flatten this layer using flatten function. Then we are adding a dense layer with 128 neurons. And the activation layer is “relu” . Relu is the go to activation function for non-linear use cases. Then I use dropout for avoid overfitting of our model. Then the final output model which has 2 layers. One is for with masks and the other for without masks. In here we are giving the activation function as Softmax. Generally in the output layer we are giving the “Softmax” activation layer because they are based on 0s and 1s. Since in this solution we are using binary classification we have used softmax as the activation function.

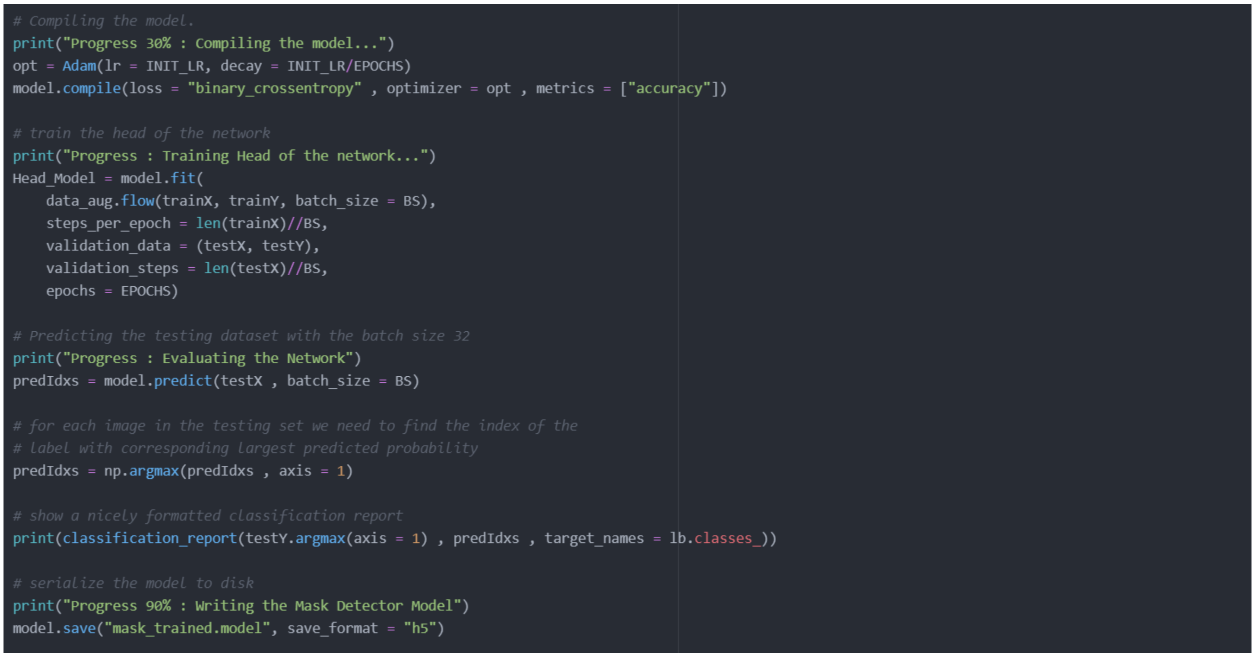
And once that is done we have call the model function. In here model function accepts two parameters one is input and the other one is output. And the inputs will be the basemodel and the output will be the headmodel.



```
# loop over all layers in the base model and freeze them so they will not be updated during the first training process
for layer in baseModel.layers:
    layer.trainable = False #freezed
```

Figure 14: Training 4

After that we have freeze the layers in the basemodel so that they will not be updated during the first training.



```
# Compiling the model.
print("Progress 30% : Compiling the model...")
opt = Adam(lr = INIT_LR, decay = INIT_LR/EPOCHS)
model.compile(loss = "binary_crossentropy" , optimizer = opt , metrics = ["accuracy"])

# train the head of the network
print("Progress : Training Head of the network...")
Head_Model = model.fit(
    data_aug.flow(trainX, trainY, batch_size = BS),
    steps_per_epoch = len(trainX)//BS,
    validation_data = (testX, testY),
    validation_steps = len(testX)//BS,
    epochs = EPOCHS)

# Predicting the testing dataset with the batch size 32
print("Progress : Evaluating the Network")
predIdxs = model.predict(testX , batch_size = BS)

# for each image in the testing set we need to find the index of the
# label with corresponding largest predicted probability
predIdxs = np.argmax(predIdxs , axis = 1)

# show a nicely formatted classification report
print(classification_report(testY.argmax(axis = 1) , predIdxs , target_names = lb.classes_))

# serialize the model to disk
print("Progress 90% : Writing the Mask Detector Model")
model.save("mask_trained.model", save_format = "h5")
```

Figure 15: Training 5

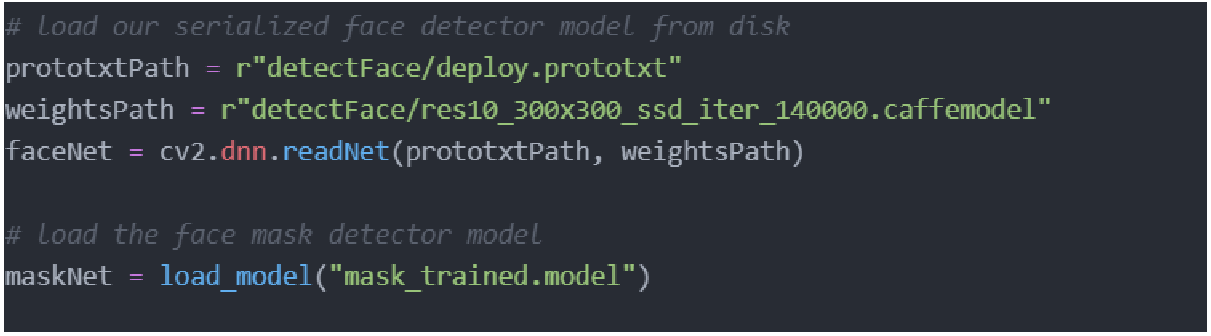
Then we are going to compile the model for that we are giving an learning rate and decay which is the “INIT_LR” we initialized at the top of the code. Then we have given the value “binary_crossentropy” for loss parameter. And the optimizer we are using is the “Adam” optimizer which is the go-to optimizer for any image predictor method. Then we have tracked the “accuracy” metric which is the only metric we are going to track in this solution.

Then we have fit the model and the “imagedatagenerator” which we explained right in the beginning so we get more data to train our model.

And finally we have serialized the model and saved to the disk in h5 format.

8.2.3 Face Detector Part

Now as explained above we have created the mask detector model. But one thing that lacks is the face detector model. For that we have downloaded two files that have will help to detect faces.



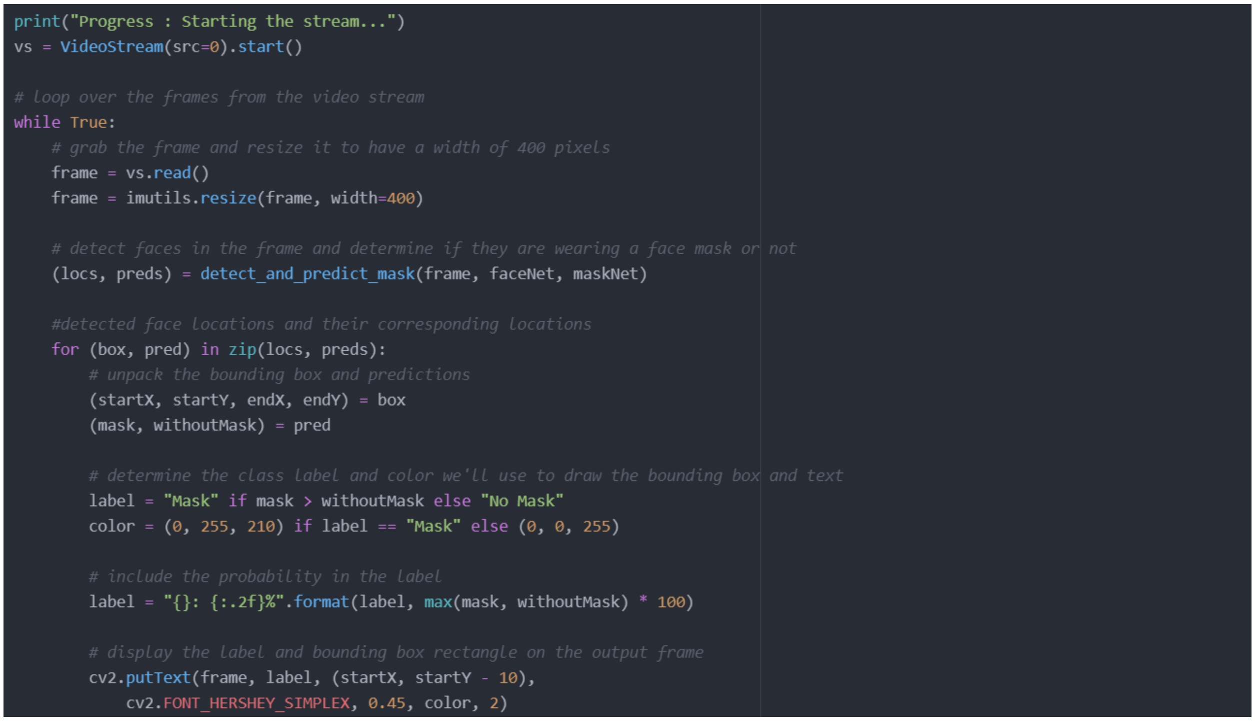
```
# load our serialized face detector model from disk
prototxtPath = r"detectFace/deploy.prototxt"
weightsPath = r"detectFace/res10_300x300_ssd_iter_140000.caffemodel"
faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)

# load the face mask detector model
maskNet = load_model("mask_trained.model")
```

Figure 16: Face Detector 1

Then we created a new .py file and loaded the facenet. Facenet is nothing but the two models that we downloaded for face detection.

And for reading those two models we have used the readNet function.

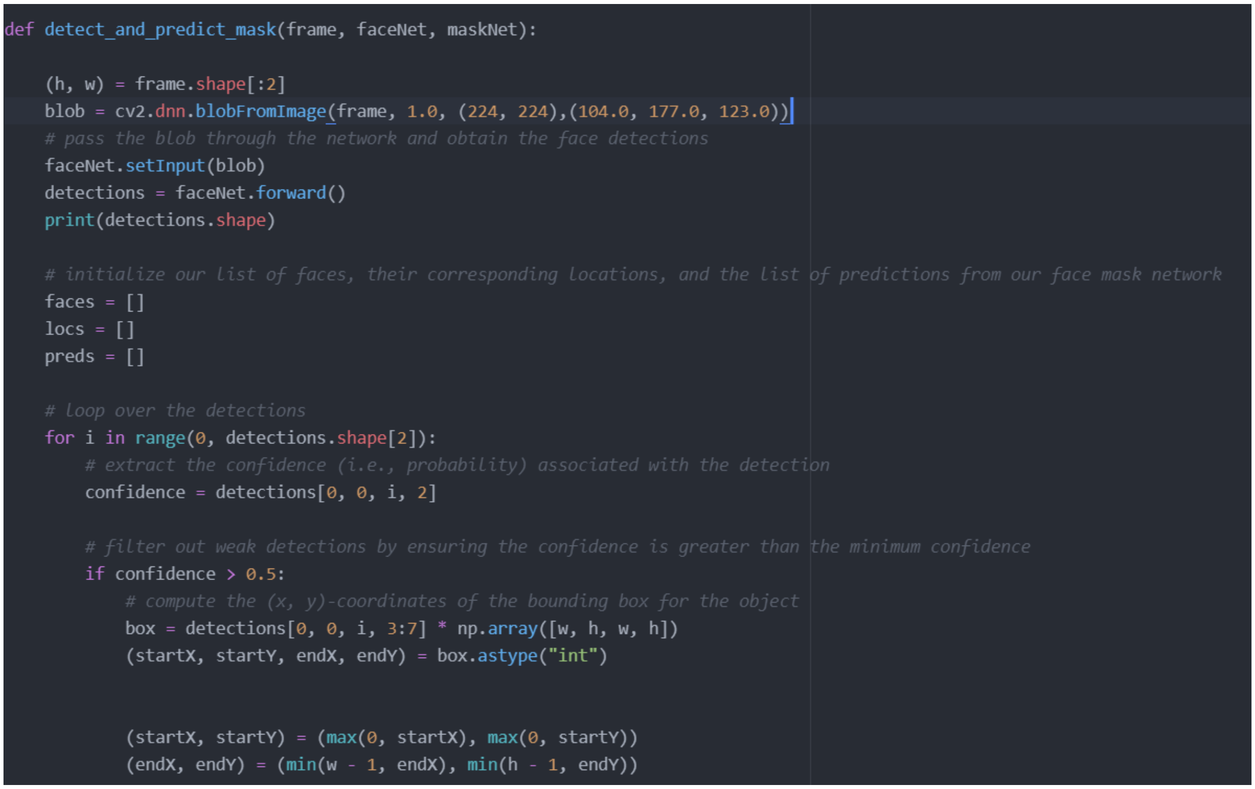
Then we have loaded the mask detection model also which we created earlier.



```
print("Progress : Starting the stream...")
vs = VideoStream(src=0).start()

# loop over the frames from the video stream
while True:
    # grab the frame and resize it to have a width of 400 pixels
    frame = vs.read()
    frame = imutils.resize(frame, width=400)

    # detect faces in the frame and determine if they are wearing a face mask or not
    (locs, preds) = detect_and_predict_mask(frame, faceNet, maskNet)

    #detected face locations and their corresponding locations
    for (box, pred) in zip(locs, preds):
        # unpack the bounding box and predictions
        (startX, startY, endX, endY) = box
        (mask, withoutMask) = pred

        # determine the class label and color we'll use to draw the bounding box and text
        label = "Mask" if mask > withoutMask else "No Mask"
        color = (0, 255, 210) if label == "Mask" else (0, 0, 255)

        # include the probability in the label
        label = "{}: {:.2f}%".format(label, max(mask, withoutMask) * 100)

        # display the label and bounding box rectangle on the output frame
        cv2.putText(frame, label, (startX, startY - 10),
            cv2.FONT_HERSHEY_SIMPLEX, 0.45, color, 2)
```

Figure 17: Face Detector 2

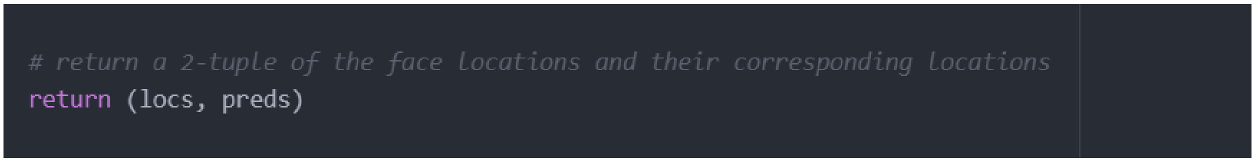
So, after that we need to load the camera. For that we used the "VideoStream" method. Then we have initialized a while method. We are reading the frame while True. Every frame is an image. So every image is flowing through sequentially in here. So those frames per second will be displayed as an video because of the illusion of the eye. Then I am opening frame with the width of 400. Now we have the facenet, masknet and the mask. Facenet for face detection, masknet for mask detection and frame for the video. Now we are going to define a method called as define_and_predict_mask with frame,facenet and masknet as arguments.



```
def detect_and_predict_mask(frame, faceNet, maskNet):

    (h, w) = frame.shape[:2]
    blob = cv2.dnn.blobFromImage(frame, 1.0, (224, 224),(104.0, 177.0, 123.0))
    # pass the blob through the network and obtain the face detections
    faceNet.setInput(blob)
    detections = faceNet.forward()
    print(detections.shape)

    # initialize our list of faces, their corresponding locations, and the list of predictions from our face mask network
    faces = []
    locs = []
    preds = []

    # loop over the detections
    for i in range(0, detections.shape[2]):
        # extract the confidence (i.e., probability) associated with the detection
        confidence = detections[0, 0, i, 2]

        # filter out weak detections by ensuring the confidence is greater than the minimum confidence
        if confidence > 0.5:
            # compute the (x, y)-coordinates of the bounding box for the object
            box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
            (startX, startY, endX, endY) = box.astype("int")


            (startX, startY) = (max(0, startX), max(0, startY))
            (endX, endY) = (min(w - 1, endX), min(h - 1, endY))
```

Figure 18: Face Detector 3

Then we have done the normal manipulations after that.



```
# return a 2-tuple of the face locations and their corresponding locations
return (locs, preds)
```

Figure 19: Face Detector 4

At the end of this function we have returned two parameters. Locs stands for location and preds stand for prediction. Location is the x and y coordinates of the rectangle surrounding the face. And prediction is the accuracy of the mask of the person .

Chapter 9- Testing and Evaluation

9.1 Functional Testing

Test ID	Test Case Name	Description	Expected Outcome	Actual Outcome	Result
001	Test Mask Detector with masks	Users should come to the camera with masks	Message with “Mask” will be displayed and forward the user to next process	Message with “Mask” is displayed and forward the user to next process	Pass
002	Test Mask Detector without masks	Users should come to the camera with masks	Message with “No Mask” will be displayed	Message with “No Mask” will be displayed	Pass
003	Test marking with RFID	Insert RFID and mark yourself present	Message with “Attendance Marked” will be displayed	Message with “Attendance Marked” will be displayed	Pass
004	Check for attendance marked students	Login as admin and check the student list	Attendance details of each student with days and time will be displayed	Attendance details of each student with days and time has displayed	Pass
005	Login	Login with admin credentials	Successfully logging into the system	Successfully logged into the system	Pass
006	Request for attendance	Users should be able to request their attendance percentage	An email will be sent to the user with attendance percentage	An email has been sent to the user with attendance percentage	Pass
007	View Student List	Admin should view the student list	Admin should be able to view the student list	Admin displayed the student list	Pass
008	Register a student	Admin should register a student into the system	Admin should be able to register a student	Admin registered a student	Pass

009	Update a student	Admin should update an already registered student	Displaying “Successfully updated” after updating user details	Displayed “Successfully Updated”	Pass
010	Delete a student	Admin should delete an already registered student	Displaying “Successfully deleted”	Displayed “Successfully Deleted”	Pass
011	Testing a newly registered student	Admin should check in the list for the newly registered student	Newly registered student should be in the list	Newly registered student is in the list	Pass
012	Testing the updated details of a student	Admin should check the list for the new details	Should display the updated details	Updated details are displayed	Pass
013	Testing the Deleted student	Admin should check the deleted student in the list	No record of the student	No record of the student	Pass
014	View Staff List	Admin should view the staff list	Admin should be able to view the staff list	Admin displayed the staff list	Pass
015	Register a staff member	Admin should register a staff member into the system	Admin should be able to register a staff member	Admin registered a staff member	Pass
016	Update a staff member	Admin should update an already registered staff member	Displaying “Successfully updated” after updating user details	Displayed “Successfully Updated”	Pass
017	Delete a staff member	Admin should delete an already	Displaying “Successfully deleted”	Displayed “Successfully Deleted”	Pass

		registered staff member			
018	Testing a newly registered student	Admin should check in the list for the newly registered staff member	Newly registered staff member should be in the list	Newly registered staff member is in the list	Pass
019	Testing the updated details of a student	Admin should check the list for the new details	Should display the updated details	Updated details are displayed	Pass
020	Testing the Deleted student	Admin should check the deleted staff member in the list	No record of the staff member	No record of the staff member	Pass
021	Testing RFID scanner	Entering an invalid card	Should Display “Invalid Card”	Displayed “invalid card”	Pass
022	Testing mask detector with different types of masks	Using different types of masks when checking	Should recognize the mask	Display “Mask” recognized mask	pass
023	Test Login	Login with invalid credentials	Should display “Enter valid credentials”	Displayed “Enter valid credentials”	Pass

9.2 Accuracy Testing

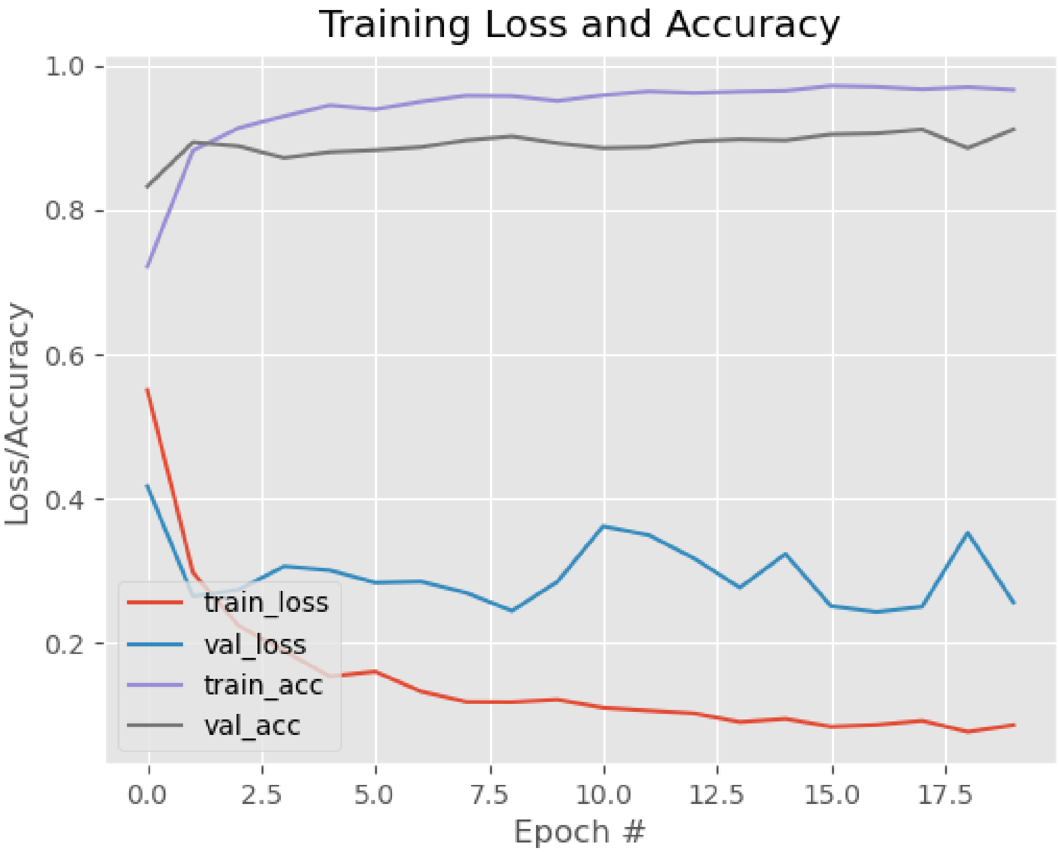
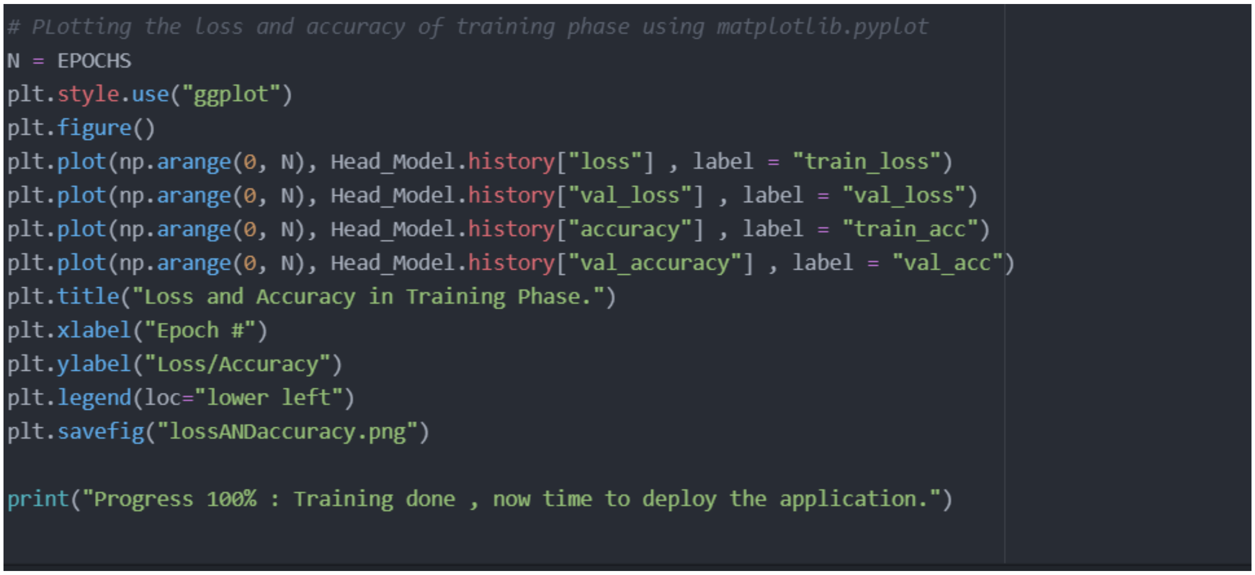


Figure 20: Accuracy Test graph

As you can see the Accuracy has been good and the loss has been reduced. So our model is very successful.



```
# PLotting the loss and accuracy of training phase using matplotlib.pyplot
N = EPOCHS
plt.style.use("ggplot")
plt.figure()
plt.plot(np.arange(0, N), Head_Model.history["loss"] , label = "train_loss")
plt.plot(np.arange(0, N), Head_Model.history["val_loss"] , label = "val_loss")
plt.plot(np.arange(0, N), Head_Model.history["accuracy"] , label = "train_acc")
plt.plot(np.arange(0, N), Head_Model.history["val_accuracy"] , label = "val_acc")
plt.title("Loss and Accuracy in Training Phase.")
plt.xlabel("Epoch #")
plt.ylabel("Loss/Accuracy")
plt.legend(loc="lower left")
plt.savefig("lossANDaccuracy.png")

print("Progress 100% : Training done , now time to deploy the application.")
```

Figure 21: Accuracy Test code

Chapter 10 - Conclusion

10.1 Limitations

This system does not have person recognition function only the mask with the face will be detected. Another limitation is students and staff cannot view their attendance percentage through this system , they will have to wait for admin's reply. Another limitation if the unique student card is lost there is no alternative to mark the attendance.

10.2 Future Enhancements

This solution can be developed further. Using an iris scanner for automatically detect the relevant student and staff member along with the mask detector can be stated as 1 further enhancement. Then having a method to view the attendance percentage for students via the system. Finally developing an alternative method of marking himself present if the RFID card is lost will be an excellent future enhancement. With those improvements this solution can be presented to the market.

Bibliography

Arjya Das, M. W. (2021, February 05). *Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV*. Retrieved from IEEE Xplore: https://ieeexplore.ieee.org/document/9342585/authors#authors

Data Flair. (2021). *Python Keras Advantages and Limitations*. Retrieved from Data Flair: https://data-flair.training/blogs/python-keras-advantages-and-limitations/

Techvidvan. (n.d.). *Advantages and Disadvantages of TensorFlow*. Retrieved from TechVidvan: https://techvidvan.com/tutorials/pros-and-cons-of-tensorflow/

The Economic Times. (2020, 11 18). *The Economic Times| Panache*. Retrieved from Masks, explained: What percentage of the virus can be blocked, and what it means for the infected and uninfected?: https://economictimes.indiatimes.com/magazines/panache/mask-explained-what-percentage-of-the-virus-can-be-blocked-and-what-it-means-for-the-infected-and-uninfected/articleshow/79282831.cms

Weidong Kuang, a. A. (2020). *A Real-Time Attendance System Using Deep Learning Face Recognition.* Texas: University of Texas Rio Grande Valley.