WIFI ENABLED IN CAMPUS SURVEILLANCE SYSTEM USING HAAR CASCADE ALGORITHM

A PROJECT REPORT

Submitted by

KISHORE S (210919104019)
MATHI SHANKAR S (210919104024)

in partial fulfillment for the award of the degree
of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING
LOYOLA INSTITUTE OF TECHNOLOGY, CHENNAI

ANNA UNIVERSITY::CHENNAI 600 025

MAY 2023

ANNA UNIVERSITY::CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **“WIFI ENABLED IN CAMPUS SURVEILLANCE SYSTEM USING HAAR CASCADE ALGORITHM”** is the bonafide work of **“KISHORE S (210919104019), MATHI SHANKAR S (210919104024)”**, who carried out the project work under my supervision.

SIGNATURE

Dr. G. BHUVANESWARI, Ph,D.,

HEAD OF THE DEPARTMENT,

PROFESSOR,
Computer Science and Engineering,
Loyola Institute of Technology,
Palanchur,
Chennai – 600 123.

SIGNATURE

Mrs. J. JONCY, M.E.,

SUPERVISOR,

ASSISTANT PROFESSOR,
Computer Science and Engineering,
Loyola Institute of Technology,
Palanchur,
Chennai – 600 123.

ACKNOWLEDGEMENT

A project of this magnitude and nature requires kind cooperation and support from many for successful completion. We wish to express sincere thanks to all of those who were involved in the completion of the project. It is our immense pleasure to express our deep sense of gratitude to our honourable Chairman **Rev. Fr. Dr. J.E. ARULRAJ**, for providing such an excellent lab facility our Correspondent **Rev. Sr. TERESA** and our Administrator **Rev. Sr. AROKIYA JANCI RANI** for continuous motivation to complete the project.

We are extremely thankful to our beloved and dynamic Principal **Dr. SUJATHA JAMUNA ANAND, Ph.D.,** for having given us an opportunity to demonstrate our innovative abilities.

We would like to thank our Head, Department of Computer Science and Engineering, **Dr. G. BHUVANESWARI, Ph.D.,** for the necessary help to complete the project successfully. We take this moment as a great opportunity to thank our project coordinator **Ms. J. SAVIJA, M.E.,** Assistant Professor, Department of Computer Science and Engineering and guide**. Mrs**. **J. JONCY, M.E.,** Assistant Professor of Computer Science and Engineering for giving us their constant encouragement, support and valuable guidance to us during the project. Last but not least, we are very grateful to our parents, friends and well-wishers for their constant encouragement during the project.

KISHORE S

MATHI SHANKAR S

ABSTRACT

This project focuses on the development of Wifi enabled in campus surveillance system using haar cascade algorithm that utilizes face recognition technology. The system is designed to enhance security and improve efficiency in workflow. The system captures and stores images of individuals accessing a building and matches them against a pre-existing database of authorized personnel. If a match is found, the system logs the entry time. Similarly, during exit, the system matches the individual's face and logs the exit time. This technology is designed to overcome the limitations of traditional access control systems like barcode scanner, fingerprint-based system. The images are processed using facial recognition algorithms that are capable of identifying a person's unique facial features and matching them against a pre-existing database of authorized personnel. Unlike access cards or passwords, which can be easily stolen or shared, a person's face cannot be replicated or forged, making it a more reliable means of identification and

identify any security breaches. This system is more time efficient way of entry process. With this system, authorized personnel can simply walk into the building, and their entry is automatically recorded. Its ability to authenticate individuals based on their facial features is making it an attractive option for organizations looking to improve their security.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

WORDS	ABBREVIATIONS
CV	Computer Vision
CSV	Comma-Separated Values
KNN	K-Nearest neighbour

CHAPTER 1

INTRODUCTION

Wi-Fi Enabled in Campus Surveillance System Using Haar Cascade Algorithm is a software-based system that utilizes face recognition to identify individuals entering and exiting a restricted area. The system works by capturing a digital image of a person's face and determine if there is a Image match with the image stores in database, it will make note of time in csv file which is created in system local storage as well as name, id, entry/exit time will be rendered in web page for easy access. If any facial feature changes in the database will invalidate the matching process.

The system uses a combination of techniques in two topics; face detection and face recognition. Haar cascade algorithm compares the information with a database of known faces to find a match. Facial recognition can help verify personal identity, but it also raises privacy issues is used to recognize the face by using positive and negative images.

Face recognition is a method of identifying or verifying the identity of an individual using their face. Face recognition systems can be used to identify people in photos, video, or in real-time. So this two steps that is face detection and face recognition steps are used in this Surveillance. It makes the work very easy and quick.

This technology is designed to overcome the limitations of traditional access control systems like barcode scanner, fingerprint-based system.

Unlike access cards or passwords, which can be easily stolen or shared, a person's face cannot be replicated or forged, making it a more reliable means of identification and identify any security breaches.

This system is more time efficient way of entry process. With this system, authorized personnel can simply walk into the building, and their entry is automatically recorded. Its ability to authenticate individuals based on their facial features is making it an attractive option for organizations looking to improve their security.

Integration with Wi-Fi security camera: To record time for entry as well as exit the face recognition system would need to be integrated with two different Wi-Fi security systems.

Once implemented, the system will provide improved security, increased efficiency, secure way to manage the flow of people entering and exiting a building. The system was also able to log the entry and exit times of individuals, along with their identity without any wait time and accurate record-keeping for the organization.

1.1. OBJECTIVES

The objectives are to reduce wait times for individuals entering and exiting, and provide accurate records for auditing and compliance purposes.

CHAPTER 2

LITERATURE SURVEY

[2.1] Mrs. B Rajeswari, SK. Hasan Ahammad, A. Nikhil Kumar, G. Praveen Kumar, P. Murali Mahesh, Development of an Automatic & Manual Class Attendance System using Haar Cascade-based Facial Recognition, International Research Journal of Engineering and Technology (IRJET) Volume: 10, e-ISSN: 2395-0056, Jan 2023.

Firstly, a camera is fixed at the entrance of the classroom or working hall. whenever a person is moving into the classroom face detection is done by the camera that is fixed at the entrance. Using LBPH Algorithm Face recognition is done and the identified face is matched with the details present in the dataset. when the faces are matched the attendace of the student are marked in the database. Manual attendace which a student came mark his/her attendace with the help of enrolment id and name.

[2.2] Mitesh Chauhan, Mandar Dhakate, Jaiveek Baria, Prof. Nileema Pathak, Attendance System using Face Recognition, International Research Journal of Engineering and Technology (IRJET) Volume: 10, e-ISSN: 2395-0056, March 2022.

This paper is to mark the attendance using the Face recognition technique. It contains mainly two parts, first is Enrolment of Student or Registration of Students in the database, and another part is Marking Attendance, as the name says, will be used for Marking down the attendance of students. In both of these processes, the two most important part is common, which is Face Detection and Face Recognition.

[2.3] Mrunal Aware, Prasad Labade, Manish Tambe, Aniket Jagtap, Chinmay Beldar, International Journal of Scientific Research in Computer Science, Engineering and Information Technology ISSN: 2456-3307, Attendance Management System Using Facial-Recognition, April 2021.

A facial recognition system is a computerized software which is suited for determining or validating a person by performing comparisons on patterns based on their facial appearances. In this system OpenCV & if image that were present in the frame is tilted then Face Landmark Estimation algorithm will be carried out and face will be transformed to be as close as possible to perfectly cantered. After that system will encode all the images that were present in the database as well as the face which were detected in the frame. It will get carried out & for each face 128 measurements were generated then it will be detected in frame it get compared with the measurements of the faces that were present in the image which is earlier stored in the database.

[2.4] Raktim Ranjan, NathKaberi Kakoty,Dibya Jyoti Bora, Face Detection and Recognition Using Machine Learning Algorithm, UGC Care Journal, ISSN: 2249-6661,Vol-43, No.-03 (III), Jan 2021.

In this paper, CLAHE, HOG features and SVM classifier-based face recognition algorithm is introduced. This proposed algorithm is compared with HOG features and SVM classifier-based face recognition algorithm. Results show that the proposed algorithm is having an improved face recognition performance. It is a time-consuming algorithm but give more accuracy and productiveness rather than other machine learning algorithms.

[2.5] Dr. M. NavaneethaKrishnan M.E., Ph.D., Mr. M. Gowtham, Mr. U. Maria Livin, Automatic Attendance Scheme using Face Recognition System, International Journal of Advanced Research in Computer Science Engineering and Information Technology Volume:6, March2021.

An automatic attendance management system that aims at solving the issues faced by manual methods of existing systems. We use the concept of face recognition to implement a system that marks the attendance of a particular person by detecting and recognizing the face. This system eliminates the cost for extra equipment, minimizes attendance-taking time, and allows users to access the data efficiently.

[2.6] Abdul Azeem, Ankit Verma, Akansha Bhatnagar, Harsh Choudhary, Ms. Kanchan Singh, Assistant Professor, Attendance system using cascade classifier, International Journal of creative Research thoughts Volume: 6 ISSN: 2320-2882, May 2020.

This system consists of a camera that captures the images of the classroom and sends it to the image enhancement module. After enhancement the image comes in the Face Detection and Recognition modules and then the attendance is marked on the database server. If any face is recognized the attendance is marked on the database from where teachers can access and use it for different purposes. Camera takes the images to detect and recognize all the students in the classroom. Two databases are displayed in the experimental setup. Face Database is the collection of face images and extracted features at the time of enrolment process and the second attendance database contains the information about the teachers and students and also uses to mark attendance

[2.7] Smitha, Pavithra S Hegde, Afshin, Face Recognition based Attendance Management System, International Journal of Engineering Research & Technology (IJERT)ISSN: 2278-0181 Vol. 9, May 2020.

All the students of the class must register themselves by entering the required details and then their images will be captured and stored in the dataset. During each session, faces will be detected from live streaming video of classroom. The faces detected will be compared with images present in the dataset. If match found, attendance will be marked for the respective student. At the end of each session, list of absentees will be mailed to the respective faculty handling the session.

[2.8] Rajath S Bharadwaj, Tejus S Rao, Vinay T R, Attendance Management Using Facial Recognition, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8, April 2019.

This system will encode all the images that were present in the database as well as the face which were detected in the frame. For performing encoding Deep Conversional Neural Network algorithm will get carried out & for each face 128 measurements were generated and were detected in frame it get compared with the database so at last by using simple liner SVM algorithm system will find the person in database of know peoples who has closest measurements to the image that were detected by camera. After finding perfect match system will generate the name and date & time & present mark and store the entry in CSV file.

[2.9] Shivam Singh, Prof. S. Graceline Jasmine, Face Recognition System, International Journal of Engineering Research & Technology (IJERT)ISSN: 2278-0181, vol-8, May 2019.

In this paper, after experimenting several techniques all technique is working well face recognition. Face Recognition Systems is based on face recognition. This system can be used to identify unknown person. In realtime scenarios, PCA outperforms other algorithms. The future work is for the recognition of the algorithm. In the system developed only by recognizing the 30-degree angle variations that should be improved. Gait recognition can be fused with face recognition systems. Poor lighting conditions.

[2.10] Ibrahim Mohamed Ahmed Ali, Abdel-Badeeh M. Salem, Face Recognition for Student Attendance using Haar Cascades Algorithm, Egyptian Computer Science Journal Vol. 43 ISSN-1110-2586, May2019.

The system connects all gathered information and performs inferences through its knowledge process to output a verification of the student. In the detection face we had enrollment 50 image of student in the database, any image saved in 20 intra class variation, the total image in the database is 1000.After training the data set the system gave 100% of recognition images and a live face.

CHAPTER 3

SYSTEM DESIGN

3.1 EXISTING SYSTEM

Automated attendance management system using face recognition. It Use Eigen face for Recognition. The main drawback is Multiple faces were not recognized. Face recognition attendance system. Stores the faces that are detected and automatically marks attendance. Used for security purposes in organization.

Previous system of Automatic & manual attendance using haar cascade based facial recognition having different modules such as Take images, Train images, Automatic Attendance, Manual attendance, check sheets. Initially Every student has to register in the system using their name, enrollment id and images of different angles to register student details. About 70 images of students are taken as a input image to train data set.

LIMITATIONS

Previous system is used for attendance marking of an individual and it is a complex system has many modules. It uses large amount of training database, it many lack storage when it is implemented in large organization

3.2 PROPOSED SYSTEM

Our proposed system focuses on the development of Wi-Fi Enabled in Campus Surveillance System Using Haar Cascade Algorithm that utilizes face recognition technology. The system is designed to enhance security and improve efficiency in workflow. The process is broken into three steps:(1) face detection and (2) face recognition to identify particular person (3) making the entry of a particular person. The system captures and stores images of individuals accessing a building and matches them against a pre-existing database of authorized personnel. If a match is found, the system logs the entry time. Similarly, during exit, the system matches the individual's face and logs the exit time. This technology is designed to overcome the limitations of traditional access control systems. The images are processed using facial recognition algorithms that are capable of identifying a person's facial features and matching them against a pre-existing database of authorized personnel.

PROPOSED ALGORITHM

```
1. Capture the person face using Face recognition.

2. Apply face detection algorithm to detect face from the registered database.

3. Convert to grey scale and proceed for data processing

4. if person face in database
       then store in Csv file as well as render in webpage
   else
       return
   endif
```

ADVANTAGES

- Unlike access cards or passwords, which can be easily stolen or shared, a person's face cannot be replicated, making it a more reliable means of identification.
- This system is more time efficient way of entry process. With this system, authorized personnel can simply walk into the building, and their entry is automatically recorded.

3.3PROBLEM STATEMENT

The system should be able to recognize faces accurately, capture the images, and store them in a database for future reference.

SOLUTION

Face detection accuracy: The system should be able to detect faces accurately in various lighting conditions and angles.

Face recognition accuracy: The system should be able to recognize faces accurately, even with changes in facial features.

Speed and efficiency: The system should be fast and efficient, with minimal lag time between face detection and recognition.

Security: The system should be secure, with authorized personnel only having access to the database and management features.

To address these challenges, the system should use Haar cascades for face detection and K-Nearest neighbor algorithms for face recognition. The system should also have a robust database to store and manage user information and logs. Finally, the system should have a user-friendly interface that allows for easy management and monitoring of the system.

CHAPTER 4

SYSTEM SPECIFICATIONS

4.1 HARDWARE REQUIREMENTS

CAMERA: We need a 2 Wi-Fi camera to capture images of the person's face. A high-resolution camera with 1080p resolution is recommended for better accuracy.

COMPUTER: We need a computer to process the images captured by the camera and perform face recognition. The computer should have a multi-core processor and a dedicated graphics card for faster processing and to access database

MEMORY: We need sufficient memory on your computer to store the face recognition models and data. We recommend 8GB of RAM to maintain efficiency of software.

STORAGE: We need sufficient storage space to store the captured images, face recognition models, and other data. A 1Tb solid-state drive (SSD) is recommended for faster read and write speeds.

NETWORK CONNECTIVITY: We need a stable internet connection if we plan to connect the entry and exit to a remote server for data storage and management.

4.2 SOFTWARE REQUIREMENTS

PROGRAMMING SOFTWARE: Python (V3.8.16) is a popular programming language. Python can be used on a server to create web applications.

OPEN CV LIBRARY: OpenCV (OpenSource Computer Vision) is a popular open-source computer vision library used for image and video processing. It includes various functions for face detection and recognition, including Haar cascades.

HAAR CASCADES: Haar cascades are a type of classifier used for object detection in images. You will need to download the pre-trained Haar cascades for face detection from the OpenCV website.

NUMPY: Numpy is a Python library for numerical computing. It provides various functions for handling arrays and matrices, which are useful for manipulating images in computer vision applications.

FACE RECOGNITION LIBRARY: You will need a face recognition library like face recognition to perform face recognition on the detected faces.

CHAPTER 5

SYSTEM DESCRIPTION

5.1 DATA FLOW DIAGRAM

The whole system is shown as a single process in a level Dataflow diagram. Each step in the system's assembly process, including all intermediate steps, are recorded here.

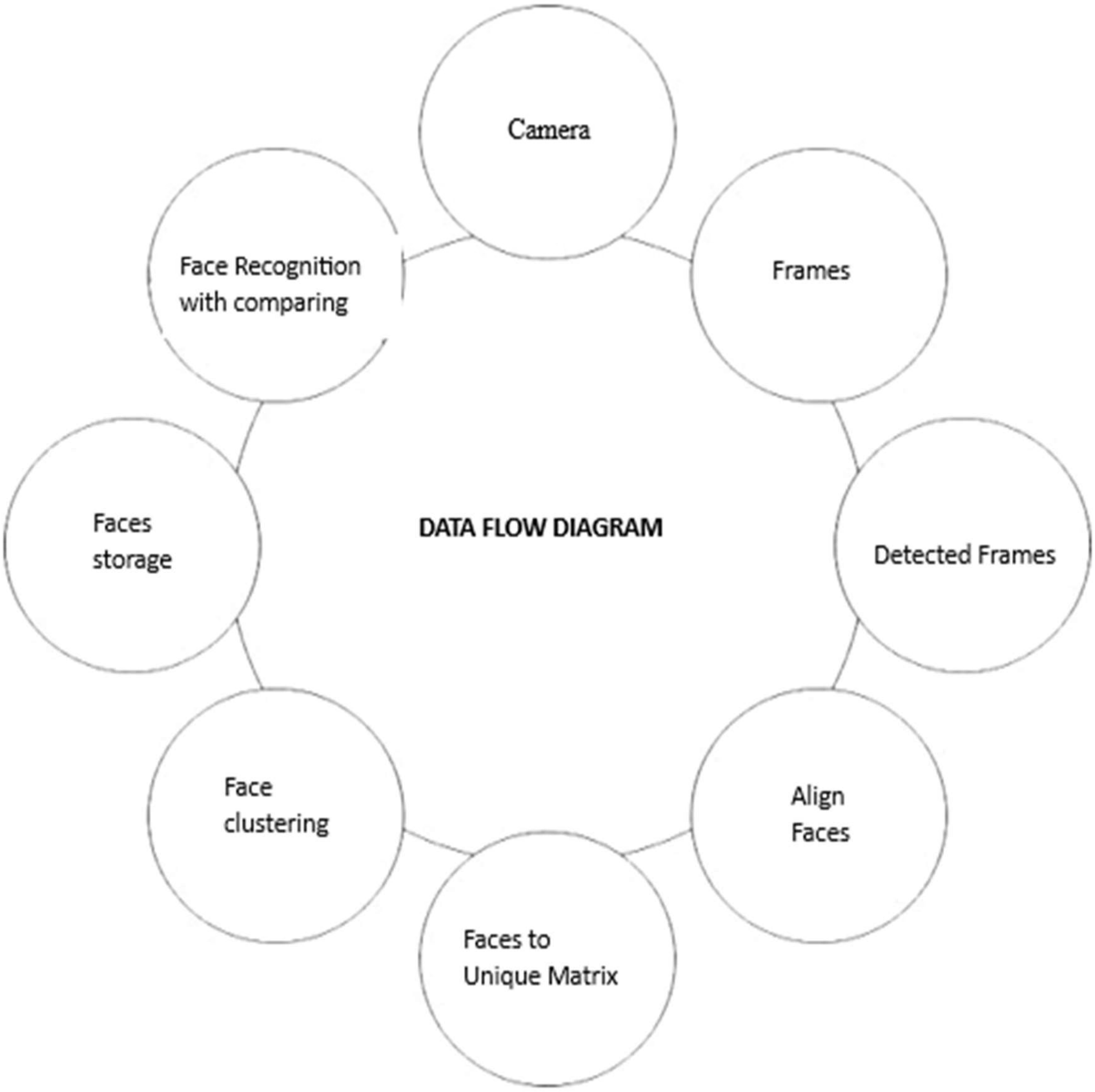


Figure 5.1 Data Flow Diagram

5.2 ARCHITECTURE DIAGRAM

This design provides a concise and understandable description of all the entities currently integrated into the system. The diagram shows how the many actions and choices are linked together. You might say that the whole process and how it was carried out is a picture. The figure below shows the functional connections between various entities.

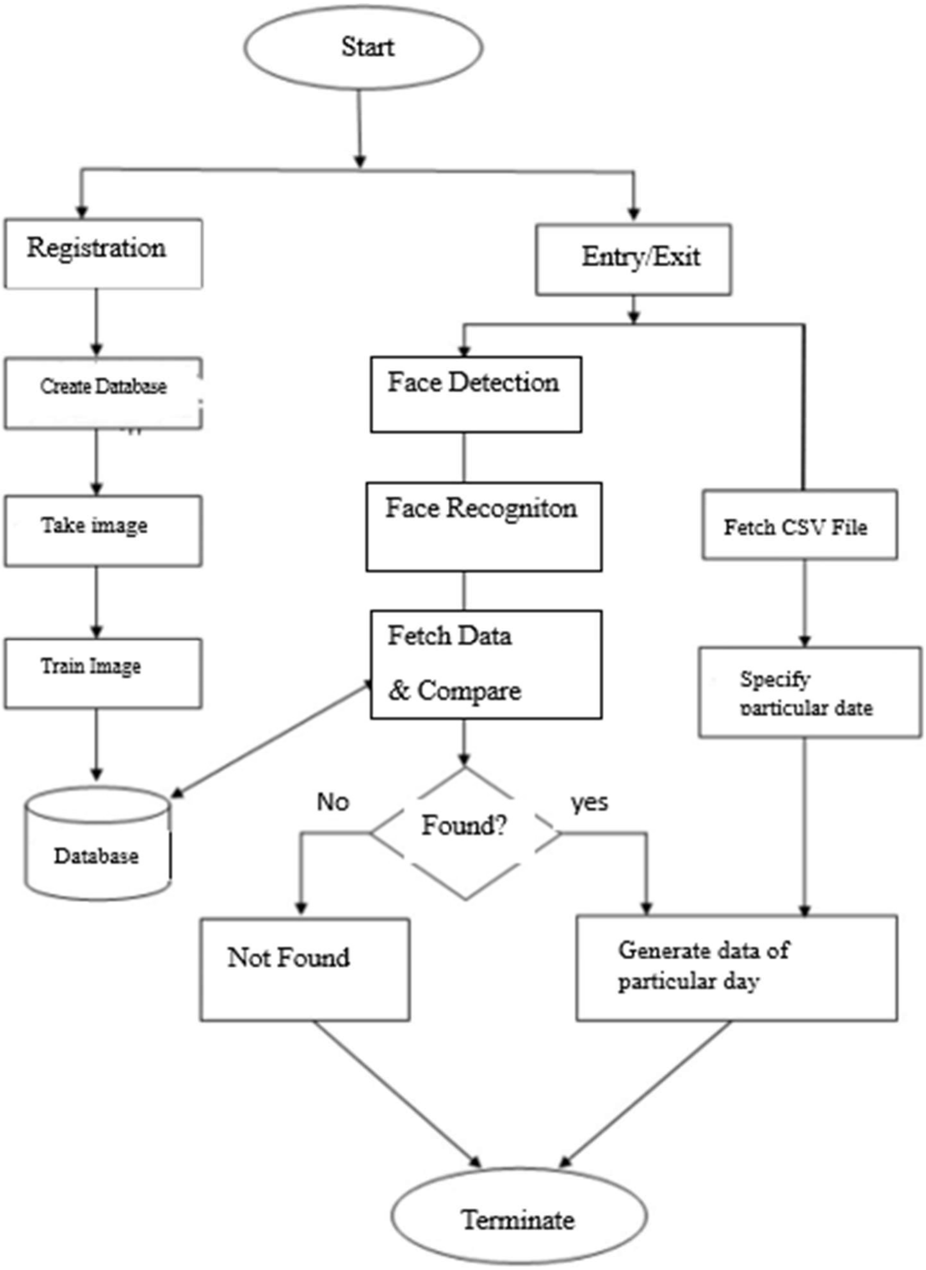


Figure 5.2 Architecture Diagram

CHAPTER 6

MODULE DESCRIPTION

The modules in our project are separated in four categories they are as follows

Pre-processing:

Initially preprocessing stage involve in the process of registering users information such as username and unique id will be gathered and specific location will be allocated for the specified user. To recognize the registered user their face will be captured in a quantity of 40 images per person and those images are stored in their specific location in the local system.

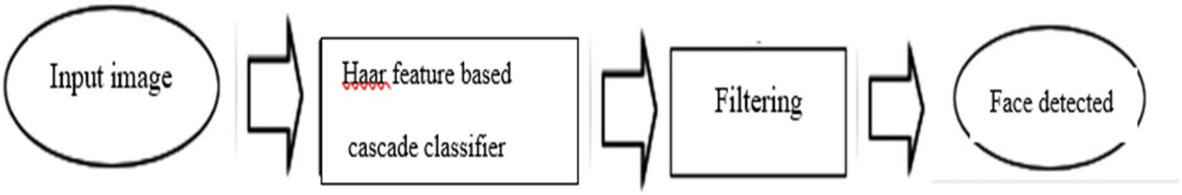


Figure 6.1 Preprocessing

Face Detection:

For face detection we have used Haar cascade algorithm. It is based on a set of simple rectangular features called Haar-like features. The training process of Haar Cascade involves providing the algorithm with a set of positive and negative images. The positive image contains example of the object being detected, while the negative images contain the images which is not to be detected. Haar Cascade algorithm is useful in real time application as it can process images and videos quickly and efficiently. This method is the oldest algorithm but still, it is a powerful algorithm for face detection. As per the study the accuracy for Haar Cascade has been obtained as 96.24%.

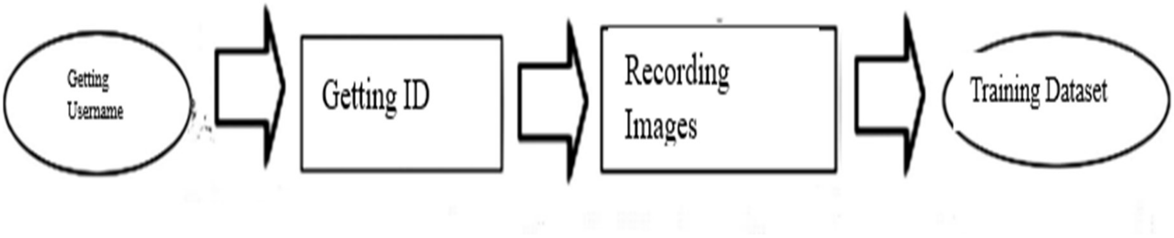


Figure 6.2 Face Detection

Face Recognition:

For face recognition we have used K-Nearest Neighbors Algorithm. It is powerful machine learning algorithm used for both classification and regression task. KNN algorithm is used to check the similarity between new and available data and put the new one into the category that is similar. When new instance is given to the algorithm, it simply finds the K closest instance in training data and predicts the output based on the average of the K nearest neighbors. The accuracy of KNN comes around 93-98%, it varies depends upon the training data set.

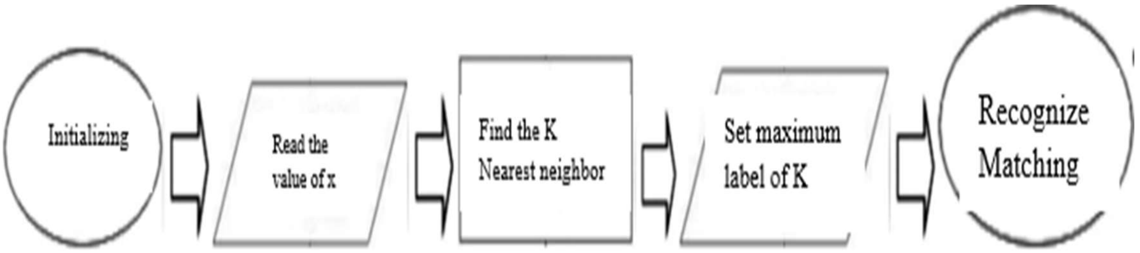


Figure 6.3 Face Recognition

Data processing:

Once the face is recognized from the trained data set, it will make note of time in csv file which is created in system local storage as well as name, id, entry/exit time will be rendered in web page for easy access.

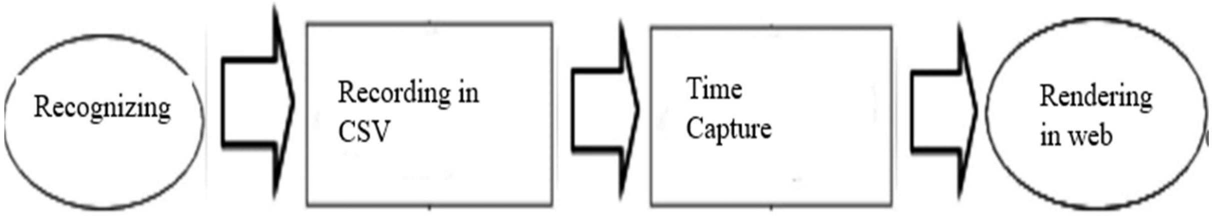


Figure 6.4 Data Processing

CHAPTER 7

RESULT AND DISCUSSION

The Wi-Fi Enabled in Campus Surveillance System Using Haar Cascade Algorithm using face recognition technology was successfully designed and developed. The system was tested using a database of stored faces, and the results showed that the system was accurate in identifying individuals. The system provides an efficient and secure way to manage the flow of people entering and exiting a building. The system was also able to log the entry and exit times of individuals, along with their identity.

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION

The Wi-Fi Enabled in Campus Surveillance System Using Haar Cascade Algorithm using face recognition provides an efficient and secure way to manage the flow of people entering and exiting a building. The system can accurately identify individuals. The system can also log the entry and exit times of individuals, providing an accurate record of who entered and exited the building and when. This system can save time and costs associated with manual processes, and it can improve the security of the building.

FUTURE ENHANCEMENT

- Our future enhancement includes to provide Secured access control for security propose on certain unauthorized cabins using Wi-Fi camera.
- The system can be further improved by incorporating additional features, such as real-time notifications to security personnel or administrators in case of any unauthorized access attempts

CHAPTER 9

APPENDIX

APPENDICES 1- CODE

ENTRY.HTML

```
<!doctype html>
<html lang="en">

<style type='text/css'>
  * {
    padding: 0;
    margin: 0;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  }


  body {
        background-image: url('https://www.techscrolling.com/wp-content/uploads/2017/10/Cyber-Security-Approach.jpg');
    background-size: cover;
    font-family: sans-serif;
    margin-top: 40px;
    height: 100vh;
    padding: 0;
    margin: 0;
  }

  table {
        border: 1px;
        font-family: arial, sans-serif;
        border-collapse: collapse;
        width: 86%;
        margin: auto;
    }

    td,
    th {
        border: 1px solid black !important;
        padding: 5px;
    }

    tr:nth-child(even) {
        background-color: #dddddd;
    }
</style>
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
     <meta name="viewport" content="width=device-width, initial-
scale=1">
                                             <link rel="stylesheet"
href="https://fonts.googleapis.com/icon?family=Material+Icons">

    <!-- Bootstrap CSS -->
              <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/css/bootstrap.min.css" rel="stylesheet"
```

```
integrity="sha384-eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6" crossorigin="anonymous">

    <title>Survillance-Entry</title>
</head>
<body>

    <div class='mt-3 text-center'>
            <h1 style="width: auto;margin: auto;color: white;padding: 11px;font-size: 44px; text-decoration:underline wheat;">SURVELLIANCE</h1>
        <h2 style="width: auto;margin: auto;color: white;padding: 5px;font-size: 14px;text-decoration:underline wheat;">(ENTRY)</h2>
    </div>

    <div class='mt-3 text-center'>
        <h3 style="font-size: 22px;color:beige;">{{ datetoday2 }} | <span id="clock"></span></h3>
    </div>

    {% if mess%}
     <p class="text-center" style="color: red;font-size: 20px;">{{ mess }}</p>
    {% endif %}
    <div class="row text-center" style="padding: 20px;margin: 20px;">
        <div class="col"
```

```
                    style="border-radius:   20px;padding:   0px;background-
color:rgb(211,211,211,0.5);box-shadow:5px              15px              5px
#8888;margin:0px 500px 0px 200px;min-height: 400px;">
          <h2 style="border-radius: 20px 20px 0px 0px;background-color:
#070604;color: white;padding: 10px;">Today's
             Entry <i class="material-icons">assignment</i></h2>

          <table style="background-color: white;">
             <tr>
                <th>S No</th>
                <th>Name</th>
                <th>UniqueId</th>
                <th>EntryTime</th>
             </tr>
             {% if l %}

             {% for i in range(l) %}
             <tr>
                <td>{{ i+1 }}</td>
                <td>{{ names[i] }}</td>
                <td>{{ rolls[i] }}</td>
                <td>{{ times[i] }}</td>
             </tr>
             {% endfor %}
             {% endif %}
          </table>
         <a style="text-decoration: none;max-width: 300px;" href="/start">
          <button
```

```
                        style="font-size: 24px;font-weight: bold;border-radius:
10px;background-color: #050401;width:490px;padding: 10px;margin-
top: 30px;margin-bottom: 30px;"
            type='submit' class='btn btn-primary'>Take Entry <i
                class="material-icons">beenhere</i></button>
            </a>
        </div>


        <div class="col"
             style="border-radius: 20px;padding: 0px;box-shadow:5px 15px
5px #8888;background-color:rgb(211,211,211,0.5);margin:100px 100px
10px 200px;height: 400px;">
             <form action='/add' method="POST" enctype="multipart/form-
data">
                   <h2 style="border-radius: 20px 20px 0px 0px;background-
color: #0f0c01;color: white;padding: 10px;">Add
                                                New User <i class="material-
icons">control_point_duplicate</i></h2>
                         <label style="font-size: 20px;"><b>Enter New User
Name*</b></label>
                <br>
                <input type="text" id="newusername" name='newusername'
                                    style="font-size: 20px;margin-top:10px;margin-
bottom:10px;" required>
                <br>
                         <label style="font-size: 20px;"><b>Enter New User
Id*</b></label>
                <br>
                <input type="number" id="newusereid" name='newuserid'
```

```
                              style="font-size:   20px;margin-top:10px;margin-
bottom:10px;" required>
            <br>
           <button style="width: 232px;margin-top: 20px;font-size: 20px;"
type='submit' class='btn btn-dark'>Add
               New User
            </button>
            <br>
                <h5 style="padding: 25px;"><i>Total Users in Database:
{{totalreg}}</i></h5>
         </form>
      </div>
   </div>
   <script type="text/javascript">
      var clockElement = document.getElementById('clock');

      function clock() {
         clockElement.textContent = new Date().toString().slice(15, 24);
      }

      setInterval(clock, 1000);
      setTimeout(function() {
         location.reload();
      }, 30000); // refresh the page every 10 seconds
   </script>
</body>
</html>
```

ENTRY.PY

```
import cv2
import os
from flask import Flask,request,render_template
from datetime import date
from datetime import datetime
import numpy as np
from sklearn.neighbors import KNeighborsClassifier
import pandas as pd
import joblib
#### Defining Flask App
app = Flask(__name__)
port=4001
#### Saving Date today in 2 different formats
datetoday = date.today().strftime("%d_%m_%y")
datetoday2 = date.today().strftime("%d-%B-%Y")
#### Initializing VideoCapture object to access WebCam
face_detector_default                                                    =
cv2.CascadeClassifier('static/haarcascade_frontalface_default.xml')
face_detector_alt                                                        =
cv2.CascadeClassifier('static/haarcascade_frontalface_alt.xml')
cap1 = cv2.VideoCapture(0)
if not cap1.isOpened():
   print("Error opening cameras")
   exit()
#### If these directories don't exist, create them
if not os.path.isdir('Entry'):
   os.makedirs('Entry')
```

```
if not os.path.isdir('static/faces'):
    os.makedirs('static/faces')
if f'Entry-{datetoday}.csv' not in os.listdir('Entry'):
    with open(f'Entry/Entry-{datetoday}.csv','w') as f:
        f.write('Name,UniqueId,EntryTime')
#### get a number of total registered users
def totalreg():
    return len(os.listdir('static/faces'))
#### extract the face from an image
def extract_faces(img):
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    face_points = face_detector_default.detectMultiScale(gray, 1.3, 5)
    face_points = face_detector_default.detectMultiScale(gray, 1.3, 5)
    return face_points
#### Identify face using ML model
def identify_face(facearray):
    model = joblib.load('static/face_recognition_model.pkl')
    return model.predict(facearray)
#### A function which trains the model on all the faces available in faces
folder
def train_model():
    faces = []
    labels = []
    userlist = os.listdir('static/faces')
    for user in userlist:
        for imgname in os.listdir(f'static/faces/{user}'):
            img = cv2.imread(f'static/faces/{user}/{imgname}')
            resized_face = cv2.resize(img, (20, 20))
            faces.append(resized_face.ravel())
        labels.append(user)
    faces = np.array(faces)
    knn = KNeighborsClassifier(n_neighbors=5)
    knn.fit(faces,labels)
    joblib.dump(knn,'static/face_recognition_model.pkl')
#### Extract info from today's Entry file in Entry folder
def extract_Entry():
    df = pd.read_csv(f'Entry/Entry-{datetoday}.csv')
    names = df['Name']
    rolls = df['UniqueId']
    times = df['EntryTime']
    l = len(df)
    return names,rolls,times,l
#### Add Entry of a specific user
def add_Entry(name):
    username = name.split('_')[0]
    userid = name.split('_')[1]
    current_time = datetime.now().strftime("%H:%M:%S")

    df = pd.read_csv(f'Entry/Entry-{datetoday}.csv')
    if int(userid) not in list(df['EmpId']):
        with open(f'Entry/Entry-{datetoday}.csv','a') as f:
            f.write(f'\n{username},{userid},{current_time}')
###################          ROUTING          FUNCTIONS
##########################
#### Our main page
@app.route('/')
def entry():
    names,rolls,times,l = extract_Entry()
```

```
return render_template('entry.html',names=names,rolls=rolls,times=times,l=l,totalreg=totalreg(),datetoday2=datetoday2)


#### This function will run when we click on Take Entry Button
@app.route('/start',methods=['GET'])
def start():
    if 'face_recognition_model.pkl' not in os.listdir('static'):
return render_template('entry.html',totalreg=totalreg(),datetoday2=datetoday2, mess='There is no trained model in the static folder. Please add a new face to continue.')
    # Initialize Pygame and buzzer sound
    pygame.init()
    buzzer_sound = pygame.mixer.Sound("buzzer.wav")
    # Open video capture device
    cap = cv2.VideoCapture(0)
    # Loop through frames
    while True:
        # Read a frame
        ret, frame = cap.read()

        # Extract faces from frame
        faces = extract_faces(frame)

        # Loop through detected faces
        for face in faces:
            # Get coordinates of face
            (x, y, w, h) = face

            # Extract face image and resize it
            face_img = cv2.resize(frame[y:y+h,x:x+w], (20, 20))

            # Identify the person in the face image
            identified_person = identify_face(face_img.reshape(1,-1))[0]

            # If the person is not in the database, play buzzer sound
            if identified_person is None:
                cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255), 2)
                buzzer_sound.play().sleep(5)
            # If the person is in the database, mark their exit and display their
name
            else:
                add_Entry(identified_person)
                cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 20), 2)
                              cv2.putText(frame,   f'{identified_person}',   (30,30),
cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 20), 2, cv2.LINE_AA)

        # Display the frame
        cv2.imshow('Entry', frame)

        # Break out of loop if 'q' is pressed
        if cv2.waitKey(1) == 27:
            break

    # Release video capture device and destroy windows
    cap.release()
    cv2.destroyAllWindows()

    names,rolls,times,l = extract_Entry()
    return render_template('entry.html',names=names,rolls=rolls,times=times,l=l,totalreg=totalreg(),datetoday2=datetoday2)


#### This function will run when we add a new user
@app.route('/add',methods=['GET','POST'])
def add():
    newusername = request.form['newusername']
    newuserid = request.form['newuserid']
    userimagefolder = 'static/faces/'+newusername+'_'+str(newuserid)
    if not os.path.isdir(userimagefolder):
        os.makedirs(userimagefolder)
    cap1 = cv2.VideoCapture(0)
    i,j = 0,0
    while 1:
        _,frame1 = cap1.read()
        faces = extract_faces(frame1)
        for (x,y,w,h) in faces:
            cv2.rectangle(frame1,(x, y), (x+w, y+h), (255, 0, 20), 2)
            cv2.putText(frame1,f'Images Captured: {i}/40',(30,30),cv2.FONT_HERSHEY_SIMPLEX,1,(255, 0, 20),2,cv2.LINE_AA)
            if j%10==0:
                name = newusername+'_'+str(i)+'.jpg'
                cv2.imwrite(userimagefolder+'/'+name,frame1[y:y+h,x:x+w])
            i+=1
          j+=1
        if j==400:
          break
        cv2.imshow('Adding new User',frame1)
        if cv2.waitKey(1)==27:
          break
    cap1.release()
    cv2.destroyAllWindows()
    print('Training Model')
    train_model()
    names,rolls,times,l = extract_Entry()
                                                                  return
render_template('entry.html',names=names,rolls=rolls,times=times,l=l,to
talreg=totalreg(),datetoday2=datetoday2)


#### Our main function which runs the Flask App
if __name__ == '__main__':
    os.environ.setdefault('FLASK_DEBUG', 'development')
    app.run(port=port,debug=True)
```

EXIT.HTML

```
<!doctype html>
<html lang="en">

<style type='text/css'>
    * {
        padding: 0;
```

```
   margin: 0;
   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}
body {
           background-image:  url('https://www.techscrolling.com/wp-
content/uploads/2017/10/Cyber-Security-Approach.jpg');
   background-size: cover;
   font-family: sans-serif;
   margin-top: 40px;
   height: 100vh;
   padding: 0;
   margin: 0;
}

table {
   border: 1px;
   font-family: arial, sans-serif;
   border-collapse: collapse;
   width: 86%;
   margin: auto;
}

td,
th {
   border: 1px solid black !important;
   padding: 5px;
}

tr:nth-child(even) {
        background-color: #dddddd;
    }
</style>
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link rel="stylesheet"
href="https://fonts.googleapis.com/icon?family=Material+Icons">
    <!-- Bootstrap CSS -->
        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-
eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlp
bzKgwra6" crossorigin="anonymous">
    <title>Survillance-Exit</title>
</head>

<body>

    <div class='mt-3 text-center'>
        <h1 style="width: auto;margin: auto;color: white;padding:
11px;font-size: 44px; text-decoration:underline
wheat;">SURVELLIANCE</h1>
        <h2 style="width: auto;margin: auto;color: white;padding: 5px;font-
size: 14px;text-decoration:underline wheat;">(EXIT)</h2>
    </div>
```

```
<div class='mt-3 text-center'>
    <h3 style="font-size: 22px;color:beige;">{{ datetoday2 }} | <span
id="clock"></span></h3>
</div>

{% if mess%}
 <p class="text-center" style="color: red;font-size: 20px;">{{ mess
}}</p>
{% endif %}

<div class="row text-center" style="padding: 20px;margin: 20px;">

    <div class="col"
            style="border-radius: 20px;padding: 10px 10px;background-
color:rgb(211,211,211,0.5);box-shadow:5px 15px 5px
#8888;margin:0px 350px 10px 350px;min-height: 400px;">
        <h2 style="border-radius: 20px 20px 0px 0px;background-color:
#070604;color: white;padding: 10px;">Today's
            Entry <i class="material-icons">assignment</i></h2>

        <table style="background-color: white;">
            <tr>
                <th>S No</th>
                <th>Name</th>
                <th>UniqueId</th>
                <th>ExitTime</th>
            </tr>
            {% if l %}
        {% for i in range(l) %}
        <tr>
          <td>{{ i+1 }}</td>
          <td>{{ names[i] }}</td>
          <td>{{ rolls[i] }}</td>
          <td>{{ times[i] }}</td>
        </tr>
        {% endfor %}
        {% endif %}
      </table>
     <a style="text-decoration: none;max-width: 300px;" href="/start">
      <button
            style="font-size:   24px;font-weight:   bold;border-radius:
10px;background-color:   #050401;width:490px;padding:   10px;margin-
top: 30px;margin-bottom: 30px;"
      type='submit' class='btn btn-primary'>Take Entry <i
        class="material-icons">beenhere</i></button>
      </a>
    </div>
  </div>
  <script type="text/javascript">
    var clockElement = document.getElementById('clock');

    function clock() {
      clockElement.textContent = new Date().toString().slice(15, 24);
    }

    setInterval(clock, 1000);
    setTimeout(function() {
            location.reload();
        }, 10000); // refresh the page every 10 seconds
    </script>
</body>

</html>
```

EXIT.PY

```
import cv2
import os
from flask import Flask,render_template
from datetime import date
from datetime import datetime
import numpy as np
from sklearn.neighbors import KNeighborsClassifier
import pandas as pd
import joblib
import pygame

#### Defining Flask App
app = Flask(__name__)
port = 5001




#### Saving Date today in 2 different formats
datetoday = date.today().strftime("%d_%m_%y")
datetoday2 = date.today().strftime("%d-%B-%Y")
```

```
#### Initializing VideoCapture object to access WebCam
face_detector_default                                              =
cv2.CascadeClassifier('static/haarcascade_frontalface_default.xml')
face_detector_alt                                                  =
cv2.CascadeClassifier('static/haarcascade_frontalface_alt.xml')
cap2 = cv2.VideoCapture(1)

if not cap2.isOpened():
    print("Error opening cameras")
    exit()

#### If these directories don't exist, create them
if not os.path.isdir('Exit'):
    os.makedirs('Exit')


if not os.path.isdir('static/faces'):
    os.makedirs('static/faces')
if f'Exit-{datetoday}.csv' not in os.listdir('Exit'):
    with open(f'Exit/Exit-{datetoday}.csv','w') as f:
        f.write('Name,UniqueId,ExitTime')

#### get a number of total registered users
def totalreg():
    return len(os.listdir('static/faces'))
```

```
#### extract the face from an image
def extract_faces(img):
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    face_points = face_detector_default.detectMultiScale(gray, 1.3, 5)
    face_points = face_detector_default.detectMultiScale(gray, 1.3, 5)
    return face_points


#### Identify face using ML model
def identify_face(facearray):
    model = joblib.load('static/face_recognition_model.pkl')
    return model.predict(facearray)


#### A function which trains the model on all the faces available in faces folder
def train_model():
    faces = []
    labels = []
    userlist = os.listdir('static/faces')
    for user in userlist:
        for imgname in os.listdir(f'static/faces/{user}'):
            img = cv2.imread(f'static/faces/{user}/{imgname}')
            resized_face = cv2.resize(img, (20, 20))
            faces.append(resized_face.ravel())
            labels.append(user)
    faces = np.array(faces)
    knn = KNeighborsClassifier(n_neighbors=5)
    knn.fit(faces,labels)
    joblib.dump(knn,'static/face_recognition_model.pkl')


#### Extract info from today's Exit file in Exit folder
def extract_Exit():
    df = pd.read_csv(f'Exit/Exit-{datetoday}.csv')
    names = df['Name']
    rolls = df['UniqueId']
    times = df['ExitTime']
    l = len(df)
    return names,rolls,times,l


#### Add Exit of a specific user
def add_Exit(name):
    username = name.split('_')[0]
    userid = name.split('_')[1]
    current_time = datetime.now().strftime("%H:%M:%S")

    df = pd.read_csv(f'Exit/Exit-{datetoday}.csv')
    if int(userid) not in list(df['EmpId']):
        with open(f'Exit/Exit-{datetoday}.csv','a') as f:
            f.write(f'\n{username},{userid},{current_time}')


##################                    ROUTING                    FUNCTIONS
#########################


#### Our main page
```

```
@app.route('/')
def exit():
    names,rolls,times,l = extract_Exit()
                                                                    return
render_template('exit.html',names=names,rolls=rolls,times=times,l=l,tot
alreg=totalreg(),datetoday2=datetoday2)


#### This function will run when we click on Take Exit Button
@app.route('/start',methods=['GET'])
def start():
    # Check if face recognition model exists
    if 'face_recognition_model.pkl' not in os.listdir('static'):
                                                                    return
render_template('exit.html',totalreg=totalreg(),datetoday2=datetoday2,m
ess='There is no trained model in the static folder. Please add a new face
to continue.')

    # Initialize Pygame and buzzer sound
    pygame.init()
    buzzer_sound = pygame.mixer.Sound("buzzer.wav")

    # Open video capture device
    cap = cv2.VideoCapture(0)

    # Loop through frames
    while True:
        # Read a frame
        ret, frame = cap.read()
    # Extract faces from frame
    faces = extract_faces(frame)

    # Loop through detected faces
    for face in faces:
        # Get coordinates of face
        (x, y, w, h) = face

        # Extract face image and resize it
        face_img = cv2.resize(frame[y:y+h,x:x+w], (20, 20))

        # Identify the person in the face image
        identified_person = identify_face(face_img.reshape(1,-1))[0]

        # If the person is not in the database, play buzzer sound
        if identified_person is None:
            cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255), 2)
            buzzer_sound.play().sleep(5)
        # If the person is in the database, mark their exit and display their
name
        else:
            add_Exit(identified_person)
            cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 20), 2)
            cv2.putText(frame, f'{identified_person}', (30,30),
cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 20), 2, cv2.LINE_AA)

    # Display the frame
    cv2.imshow('Exit', frame)
```

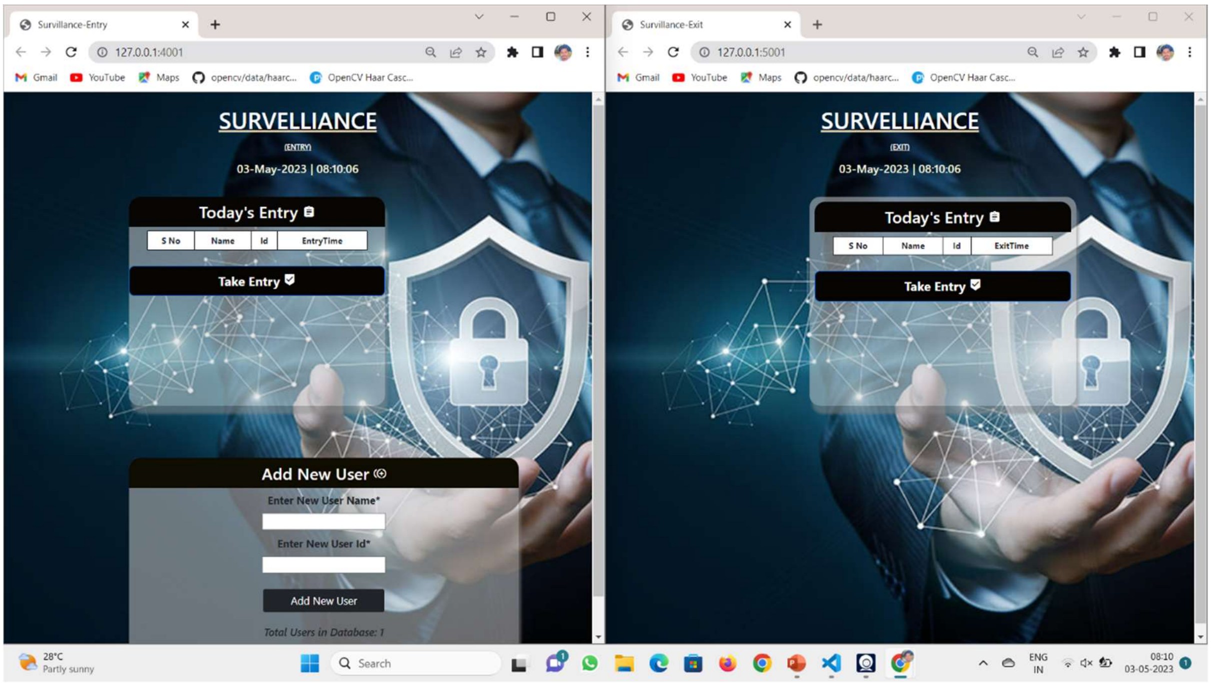
```
        # Break out of loop if 'q' is pressed
        if cv2.waitKey(1) == ord('q'):
            break

    # Release video capture device and destroy windows
    cap.release()
    cv2.destroyAllWindows()

    # Extract data from exit log and render exit page
    names, rolls, times, l = extract_Exit()
    return render_template('exit.html', names=names, rolls=rolls, times=times, l=l, totalreg=totalreg(), datetoday2=datetoday2)

#### Our main function which runs the Flask App
if __name__ == '__main__':
    os.environ.setdefault('FLASK_DEBUG', 'development')
    app.run(port=port,debug=True)
```

ENTRYDATA.PY

```
import pandas as pd

# get file names from user
Entryfile = input("provide dd_mm_yy ")
Exitfile = Entryfile

# set file paths
entry_path = "C:/Users/Mathi Shankar/OneDrive/Desktop/Survilence of HR Department to regulate In and Out-main/Survilence of HR Department-entry/Entry//"
```

```
exit_path = "C:/Users/Mathi Shankar/OneDrive/Desktop/Survilence of
HR Department to regulate In and Out-main/Survilence of HR
Department - exit/Exit//"

# extension
extension = ".csv"

#constant prefix for file name
constant1 = "Entry-"
constant2 = "Exit-"

   # read the files
f1 = pd.read_csv(entry_path + constant1 + Entryfile + extension )
f2 = pd.read_csv(exit_path + constant2  + Exitfile + extension)

   # merge the files
f3 = f1[["Name", "EmpId", "EntryTime"]].merge(f2[["Name", "EmpId",
"ExitTime"]], on=["Name", "EmpId"], how="left")

   # create a new file
f3.to_csv("entrydata.csv", index=False)
```

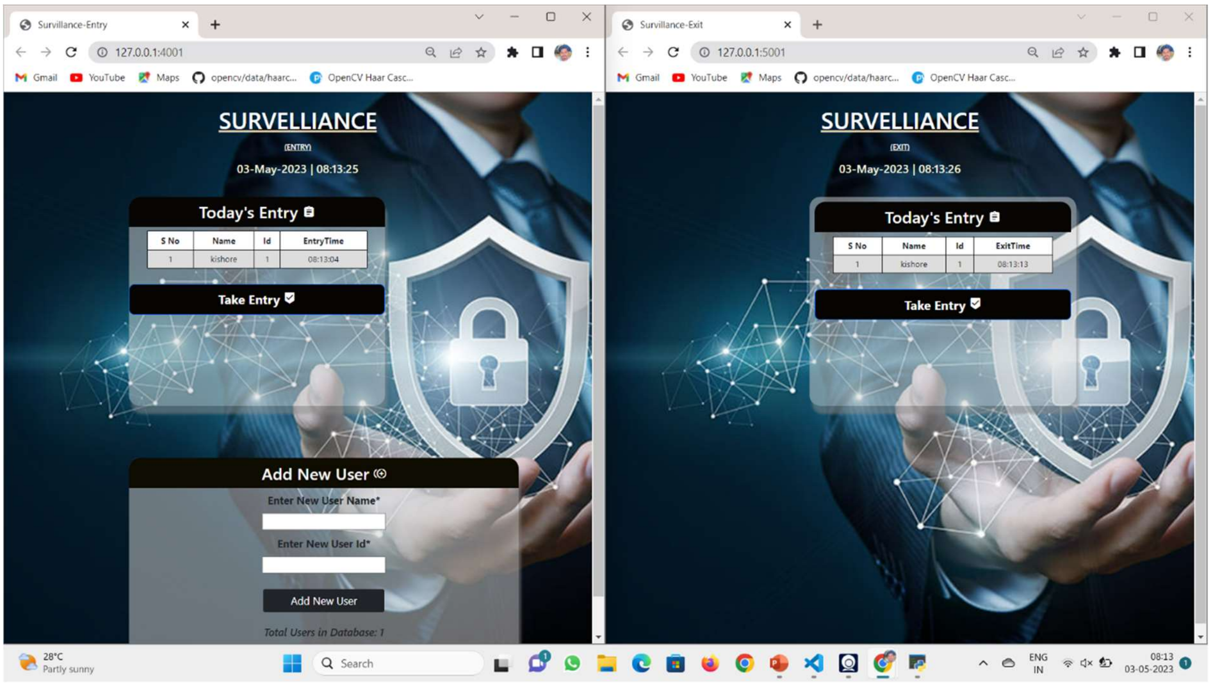
APPENDICES 2 - OUTPUT



S No	Name	Id	EntryTime

S No	Name	Id	ExitTime

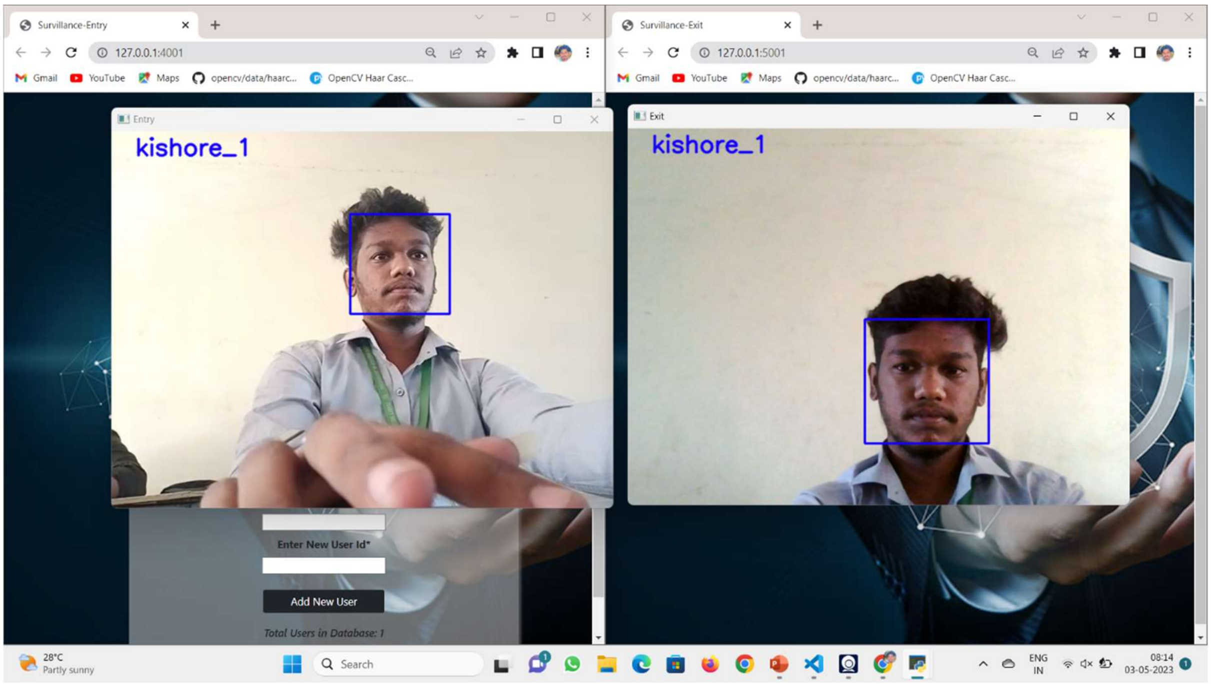
Figure 9.1 Template



S No	Name	Id	EntryTime
1	kishore	1	08:13:04

S No	Name	Id	ExitTime
1	kishore	1	08:13:13

Figure 9.2 Rendering



Total Users in Database: 1

Figure 9.3 Face capture

CHAPTER 10

REFERENCES

[1] Development of an Automatic & Manual Class Attendance System using Haar Cascade-based Facial Recognition, Mrs. B Rajeswari, SK. Hasan Ahammad, A. Nikhil Kumar, G. Praveen Kumar, P. Murali Mahesh, 2023.

[2] Attendance System using Face Recognition, Mitesh Chauhan, Mandar Dhakate, Jaiveek Baria, Prof. Nileema Pathak 2022.

[3] Automatic Attendance Scheme using Face Recognition System, Dr. M. Navaneetha Krishnan M.E., Ph.D., Mr. M. Gowtham, Mr. U. Maria Livin,2021.

[4] Attendance Management System Using Facial-Recognition, Mrunal Aware, Prasad Labade, Manish Tambe, Aniket Jagtap, Chinmay Beldar May/June 2021.

[5] Face Detection and Recognition Using Machine Learning Algorithm ,Raktim Ranjan ,NathKaberi Kakoty,Dibya Jyoti Bora, December 2020.

[6] Face Recognition based Attendance Management System, Smitha, Pavithra S Hegde, Afshin May 2020.

[7] Attendance system using cascade classifier, Abdul Azeem, Ankit Verma, Akansha Bhatnagar, Harsh Choudhary, Ms. Kanchan Singh, Assistant Professor 2020.

[8] Attendance Management Using Facial Recognition, Rajath S Bharadwaj, Tejus S Rao, Vinay T R 2019.

[9] Face Recognition for Student Attendance using Haar Cascades Algorithm Ibrahim Mohamed Ahmed Ali, Abdel-Badeeh M. Salem 2019.