Project Title:

Smart Attendance System Using Face Recognition

Team: 578

Smart Bridge Externship Project

Name	Collage Register Number
Virupakshi Akhileswar Reddy	20BCE2353
Polu Akshith Sai Kumar Reddy	20BCE2391
Virupakshi Dharaneswar Reddy	20BCE2351
Abhinav Raikwar	20BAI10375

1 Introduction:

1.1 Overview:

A reliable and consistent attendance tracking system is a must for any school. There are two ways in which an organization might keep track of employee attendance: either manually or automatically.

The vast majority of students still use traditional methods of taking attendance, such as shouting out names from attendance registers, during class. It's a time-consuming process to hand call out students' names. There is a risk that unauthorized individuals will make up fake attendance by misusing the RFID cards given to each student because of their unique identifiers.

1.2 Purpose:

A smart attendance system using face recognition offers a convenient and efficient solution for tracking attendance. By automatically recognizing individuals' faces as they enter or leave a premises, the system eliminates the need for manual sign-in sheets or swipe cards. This not only saves time and reduces administrative work but also minimizes the chances of fraudulent attendance. With enhanced security features, organizations can ensure that only authorized personnel gain access to specific areas or facilities. Moreover, the system allows for real-time monitoring, alerts, and seamless integration with other systems, leading to improved resource management and operational efficiency. Overall, a smart attendance system using face recognition simplifies attendance tracking, enhances security, and optimizes organizational processes.

2 Literature Survey:

2.1 Existing Problems:

Racial bias due to testing inaccuracies

Racial bias in facial recognition systems is a concern, with algorithms achieving over 90% classification accuracy but not universality. Recent developments challenge the ethics of facial recognition.

Lack of informed consent and transparency

Privacy concerns arise in data mining, particularly online, where anonymized information is often collected. Facial recognition algorithms excel on large datasets, capturing multiple images under different lighting conditions and angles.

2.2 Proposed Solutions:

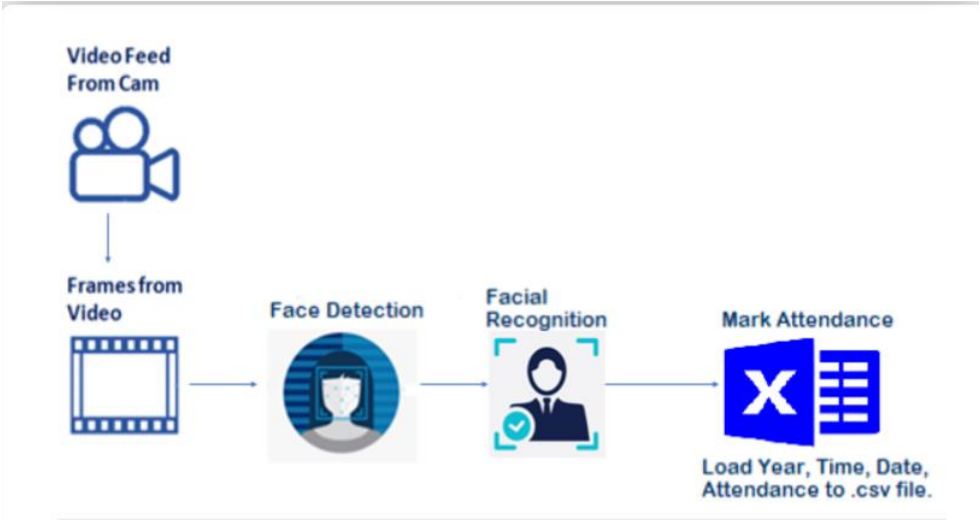
1.Increasing Efficiency and Capability of model Employee attendance tracking is crucial for organizations, but manual methods can be time-consuming and susceptible to human errors. Automated systems like facial recognition reduce mistakes, improve productivity, and free up time for other critical activities by increasing the efficiency of the trained model.

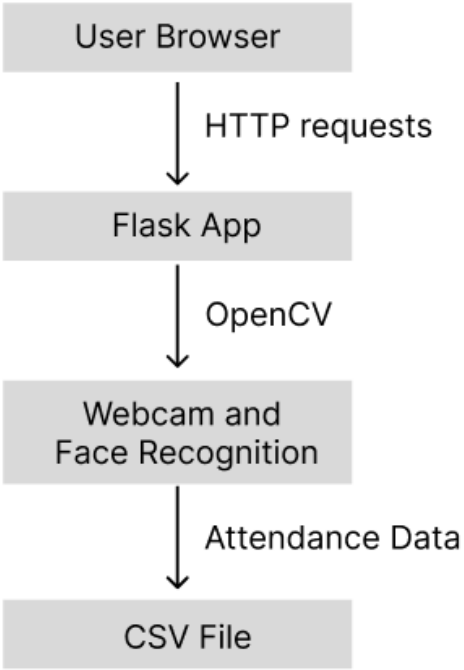
2.Integration with Other Systems

Facial recognition attendance systems can be integrated with payroll systems to track employee work hours and calculate pay. These flexible systems can be easily modified and geolocated, allowing companies to monitor their field force at all work locations worldwide.

3 Theoretical Analysis:

3.1 Block Diagram:





3.2 Hardware / Software designing:

Hardware Requirements:

Window OS: Min.OS 10, gpu required.
Ram: min. 8gb required
Rom: min.256gb required.

Software Requirements:

Application: PyCharm /Anaconda
Programming Language: Python
Libraries needed:
OpenCV
Face-recognition
Dlib

4 Experimental Investigations:

Careful study and inquiry are required when creating a smart attendance system that makes use of facial recognition technology. To determine if using facial recognition software to track attendance is practical, a feasibility study is carried out. Existing face recognition algorithms, their quality, the hardware needed to implement them, any obstacles that might arise, and the system's cost-effectiveness are all taken into account in this analysis.

To guarantee trustworthy outcomes, face recognition algorithms are rigorously tested and assessed for their accuracy and performance. The accuracy of the algorithm in identifying individuals in a given database is evaluated through extensive testing. Understanding the efficacy of the algorithm requires looking at metrics like false acceptance rate (FAR) and false rejection rate (FRR). To get the best possible performance in real-world settings, it is necessary to evaluate and fine-tune a variety of algorithms.

The face recognition algorithm requires a large and varied dataset of face photos from which to learn. In order to make the system more robust, this dataset includes a wide range of lighting, positions, and facial emotions. An effective and trustworthy face recognition model relies heavily on the quantity and quality of the dataset used for training.

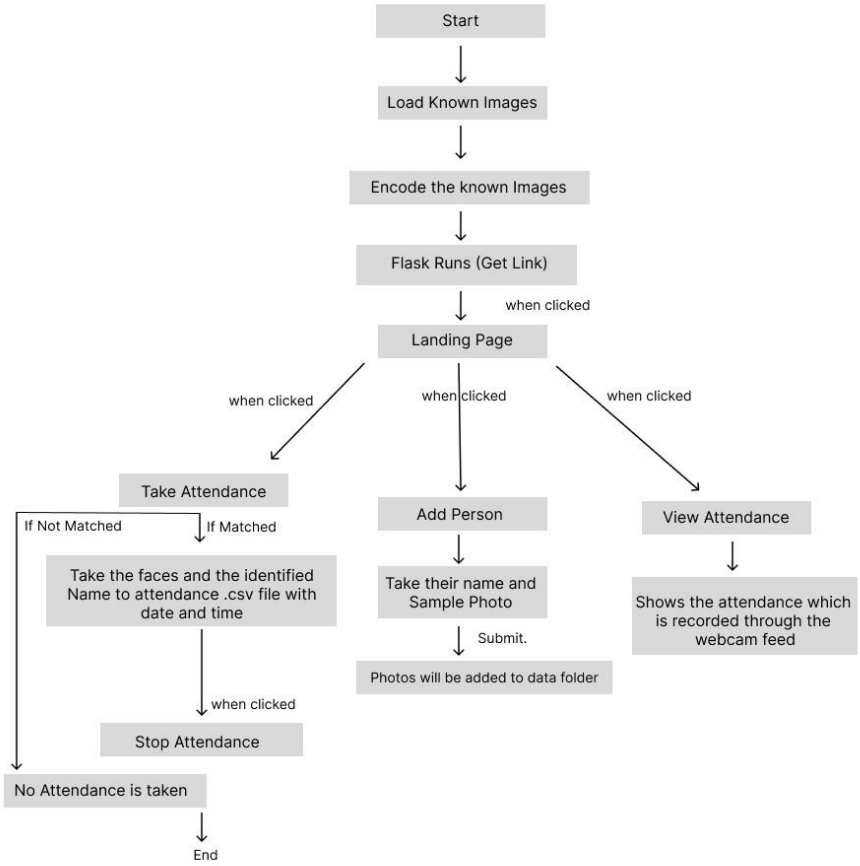
Other possible analyses include the implementation of privacy and security measures to protect personal data, the evaluation of the system's scalability to handle a large number of users, and the evaluation of computational resources required for real-time face recognition.

Developers may rest assured that the smart attendance system based on face recognition technology will be well-designed, accurate, and capable of reaching the necessary

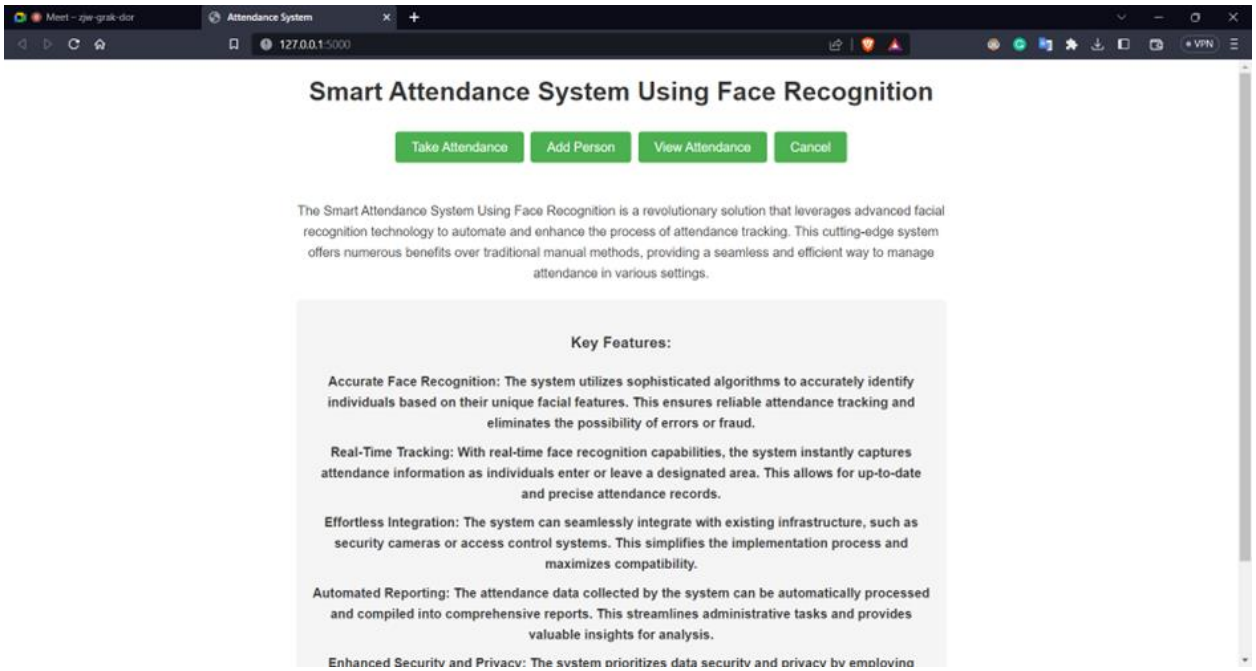
objectives in an efficient and dependable manner if they conduct the aforementioned analyses and investigations.

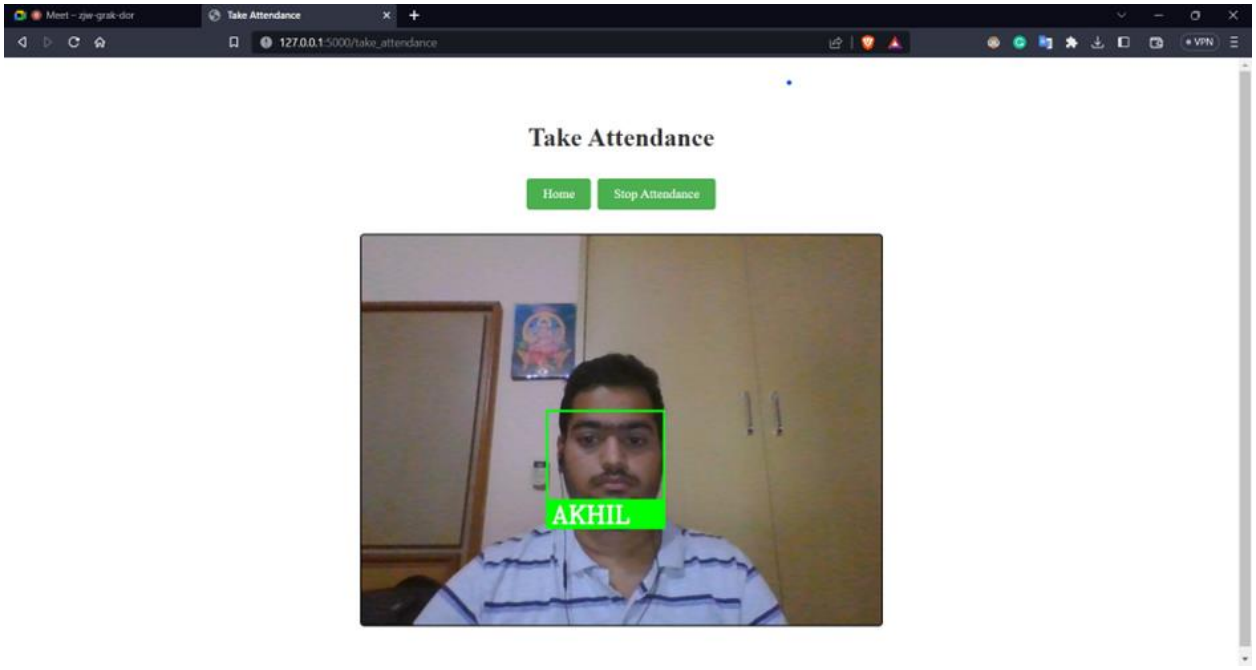
An integral part of developing a smart attendance system based on facial recognition technology is researching any moral or legal problems that may arise. Implementing measures to ensure the security of private information involves analyzing privacy risks, ensuring compliance with data protection legislation, and taking appropriate action. Developers can instill trust in the system's usage and encourage responsible deployment of face recognition technology by conducting a thorough assessment of these factors, which will allow them to design a system that respects user privacy, maintains data integrity, and complies with relevant laws and regulations.

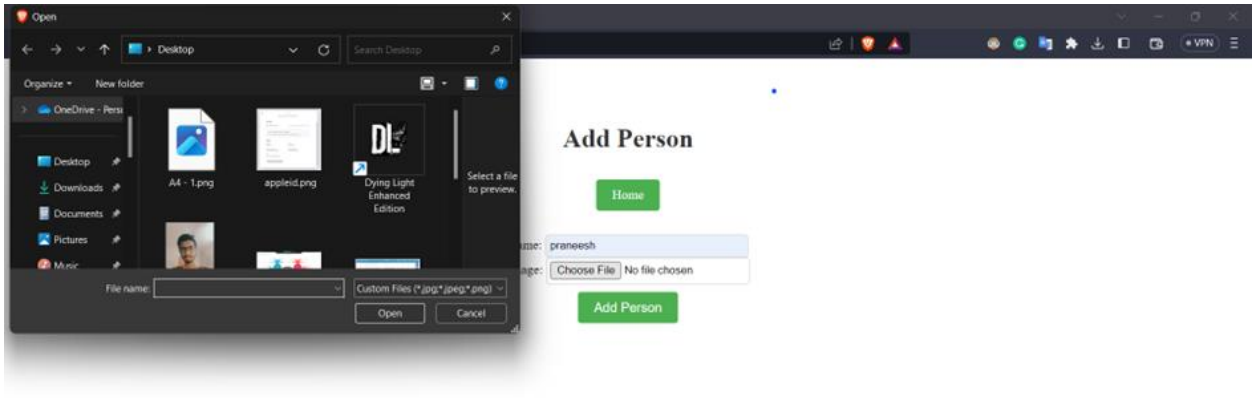
5 Flowchart:

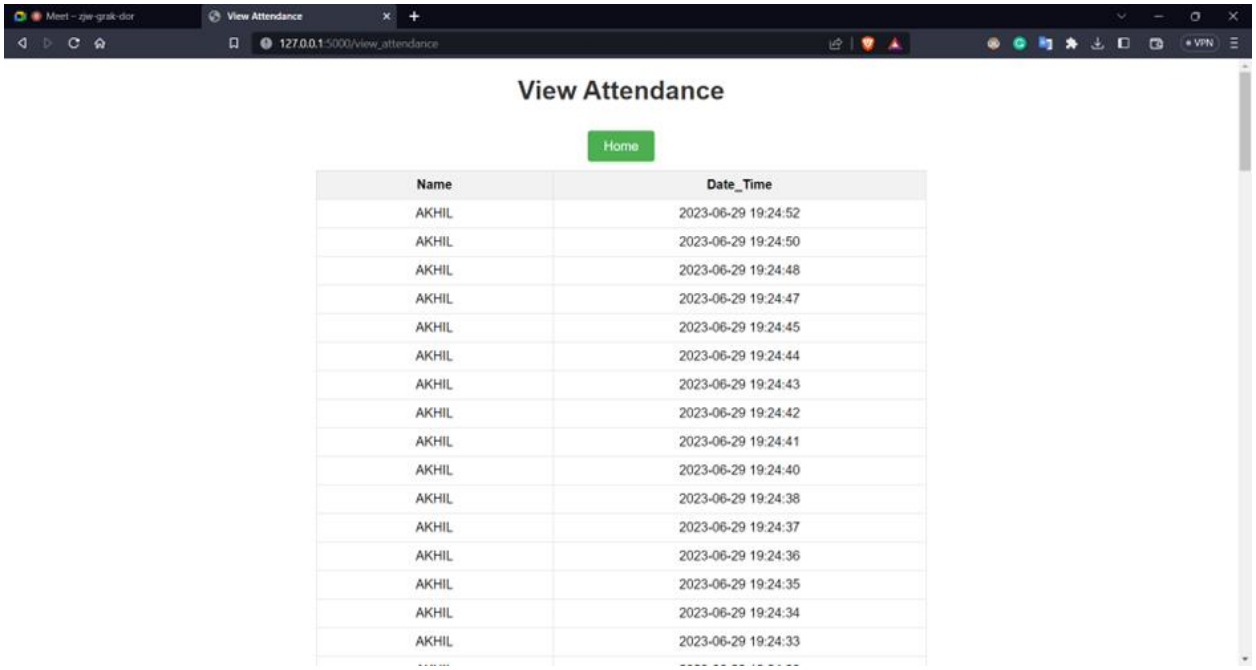


6 Results/findings of our project along with the screenshot:

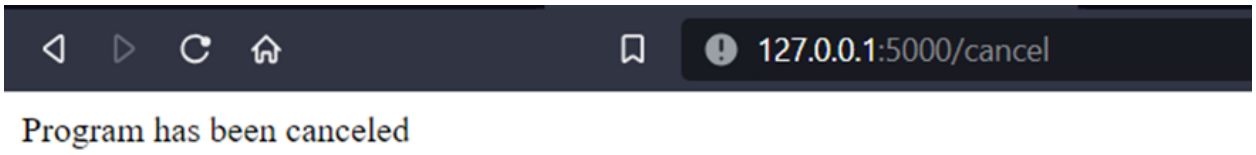








Name	Date_Time
AKHIL	2023-06-29 19:24:52
AKHIL	2023-06-29 19:24:50
AKHIL	2023-06-29 19:24:48
AKHIL	2023-06-29 19:24:47
AKHIL	2023-06-29 19:24:45
AKHIL	2023-06-29 19:24:44
AKHIL	2023-06-29 19:24:43
AKHIL	2023-06-29 19:24:42
AKHIL	2023-06-29 19:24:41
AKHIL	2023-06-29 19:24:40
AKHIL	2023-06-29 19:24:38
AKHIL	2023-06-29 19:24:37
AKHIL	2023-06-29 19:24:36
AKHIL	2023-06-29 19:24:35
AKHIL	2023-06-29 19:24:34
AKHIL	2023-06-29 19:24:33



7 Advantages and Disadvantages of the Proposed Solution: -

7.1 Advantages:

It enhances security system of any organization:

This system can recognize employees and confirm or refuse access, especially for sensitive data. This ensures that sensitive information, such as work schedules and pay rates, remains confidential and only accessible to the employee.

It has automated time tracking:

Facial recognition attendance systems automate entry and exit monitoring, enabling analytics to locate and recognize faces without human intervention. This system tracks employee work time, attendance, overtime, and leave, eliminating deception through proxy attendance. Reliable logs show on-site presence and time.

7.2 Disadvantages:

Pose variation in face detection can cause problems due to changes in observer angle and head position rotation. While some systems can tolerate small variations, large rotational angles are difficult. Pose correction is essential for sensitive facial recognition systems (FRSs) to align the face with the image's axis.

Variations in illumination can reduce FRS efficiency, making face detection and recognition difficult for moderate background lighting. Higher light levels can lead to over-exposure and undetectable facial patterns. Algorithms like equalization techniques and multiple algorithms can help address this issue, but relying solely on these techniques is not desirable.

Variations Face images can vary due to individual expressions influenced by their emotional state. Recognizing different facial expressions is crucial for evaluating emotional states.

Human expressions include macro-expressions like disgust, anger, happiness, fear, sadness, and surprise. Cosmetics and hair styles can also affect facial expression.

8 Applications: -

8.1 In Healthcare sector: -

Smart attendance facial recognition improves hospital environment by enabling contactless authentication of medical staff, especially during pathogen spread. It enables staff to be identified even without masks and gloves, enabling remote monitoring and consultations. Multinational corporations have developed technology for remote monitoring and self-verification.

8.2 In Manufacturing sector: -

Smart attendance helps in monitoring and tracking in an industrial environment are challenging due to long queues and inability to carry credentials or access cards. Face recognition technology addresses these issues by allowing employees to mark their attendance at the factory gate without needing to visit a section office. This technology accurately tracks work time, in-out, and detects intruders, allowing employees to focus on their work without interruptions.

8.3 In Education sector: -

Educational institutions can improve attendance management by using facial recognition cameras at class entrances. These cameras can automatically register student and teacher attendance, reducing time spent on marking and canteen management.

9 Conclusion:

Smart Attendance face recognition systems face numerous challenges, including aging, occlusions, illumination, resolution, expression, and poses. These challenges can be controlled using appropriate algorithms, but large-scale problems can hinder face recognition. Despite advancements in algorithms, Smart attendance face recognition faces numerous challenges and opportunities for innovation.

10 Future Scope:

The security of the system can be improved by implementing multi-factor authentication, in which many methods of authentication are used. This kind of authentication uses multiple factors to increase confidence in the attendance records.

The addition of real-time monitoring features can greatly enhance the system's usefulness. It can help administrators keep tabs on attendance patterns, spot outliers, and intervene as

soon as necessary. Proactive methods to enhance attendance management can be made possible through real-time monitoring.

Accuracy Improvements Constant study and development can boost the precision of face recognition programmes. Higher recognition rates can be achieved with fewer false acceptances and rejections by improving the algorithms and training models using more broad and diverse datasets.

Implementing strong anti-spoofing mechanisms can protect the facial recognition system from being tricked fraudulently. Improve the system's safety and stop spoofing attacks by using techniques like liveness detection, which confirms the presence of a live individual.

Administrative tasks can be simplified with the smart attendance system's seamless interaction with pre existing infrastructure, such as access control systems or HR management software. This integration has the potential to streamline attendance tracking and data administration, cutting down on administrative time and improving productivity.

The simplicity and accessibility of taking attendance can be greatly improved with the use of mobile applications and cloud connectivity. Accessing and analyzing attendance records in real time from anywhere is made possible by cloud computing, which also allows for their centralization and synchronization.
Using facial analytics for purposes other than tracking attendance can yield interesting results. Facial analytics may help businesses make data-driven decisions to boost employee and student engagement and well-being by gathering demographic information, doing emotion analysis, and identifying behavioral trends.

11 Bibliography:

11.1 References:

[1] Kar, N., Debbarma, M. K., Saha, A., & Pal, D. R. (2012). Study of implementing automated attendance system using face recognition technique. *International Journal of computer and communication engineering*, *1*(2), 100-103.

[2] RoshanTharanga, J. G., Samarakoon, S. M. S. C., Karunarathne, T. A. P., Liyanage, K. L. P. M., Gamage, M. P. A. W., & Perera, D. (2013). Smart attendance using real time face recognition (smart-fr). *Department of Electronic and Computer Engineering, Sri Lanka Institute of Information Technology (SLIIT), Malabe, Sri Lanka*.

[3] A. V. a. R. Tokas, "Fast Face Recognition Using Eigen Faces," IJRITCC, vol. 2, no. 11, pp. 3615-3618, November 2014.

[4] Vinay Hermath, Ashwini Mayakar, "Face Recognition Using Eigen Faces and," IACSIT, vol. 2, no. 4, pp. 1793-8201, August 2010

[5] Patil, Ajinkya, and Mrudang Shukla. "Implementation of classroom attendance system based on face recognition in class." International Journal of Advances in Engineering & Technology 7.3 (2014): 974.

[6] Kanti, Jyotshana, and Shubha Sharm. "Automated Attendance using Face Recognition based on PCA with Artificial Neural Network." International journal ofscience and research IJSR(2012).

[7] MuthuKalyani, K., and A. VeeraMuthu. "Smart application for AMS using face recognition." Computer Science &Engineering 3.5 (2013): 13.

[8] Deshmukh, Badal J and Sudhir M. Kharad. "Efficient Attendance Management: A Face Recognition Approach."(2014).

[9] Wagh, Priyanka, et al. "Attendance system based on face recognition using eigen face and PCA algorithms." 2015International Conference on Green Computing and Internet of Things (ICGCIoT). IEEE, 2015.

[10] N. M. Ara, N. S. Simul and M. S. Islam, "Convolutional Neural Network Approach for Vision Based Student Recognition System," in 2017 20th International Conference of Computer and Information Technology (ICCIT), 22-24 December, 2017, Sylhet, 2017.

Appendix:

Source code

```
import cv2
import numpy as np
import face_recognition
import os
from datetime import datetime, timedelta
import csv
from flask import Flask, render_template, Response, request
import time
app = Flask(__name__)
path = 'data'
images = []
```

```
classNames = []
myList = os.listdir(path)
print(myList)
for cl in myList:
    curImg = cv2.imread(f'{path}/{cl}')
    images.append(curImg)
    classNames.append(os.path.splitext(cl)[0])
print(classNames)
def findEncodings(images):
    encodeList = []
    for img in images:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
        encodeList.append(encode)
    return encodeList
def markAttendance(name):
    with open('Attendance.csv', 'a') as f:
        now = datetime.now()
        dtString = now.strftime('%Y-%m-%d %H:%M:%S')
        f.writelines(f'{name},{dtString}\n')
encodeListKnown = findEncodings(images)
print('Encoding Complete')
def generate_frames():
    cap = cv2.VideoCapture(0)
    while True:
        success, img = cap.read()
        imgS = cv2.resize(img, (0, 0), None, 0.25, 0.25)
        imgS = cv2.cvtColor(imgS, cv2.COLOR_BGR2RGB)

        facesCurFrame = face_recognition.face_locations(imgS)
        encodesCurFrame = face_recognition.face_encodings(imgS, facesCurFrame)

        for encodeFace, faceLoc in zip(encodesCurFrame, facesCurFrame):
            matches = face_recognition.compare_faces(encodeListKnown, encodeFace)
            faceDis = face_recognition.face_distance(encodeListKnown, encodeFace)
            matchIndex = np.argmin(faceDis)

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

```
            y1, x2, y2, x1 = faceLoc
            y1, x2, y2, x1 = y1 * 4, x2 * 4, y2 * 4, x1 * 4
            cv2.rectangle(img, (x1, y1), (x2, y2), (0, 255, 0), 2)
            cv2.rectangle(img, (x1, y2 - 35), (x2, y2), (0, 255, 0), cv2.FILLED)
            cv2.putText(img, name, (x1 + 6, y2 - 6), cv2.FONT_HERSHEY_COMPLEX, 1,
(255, 255, 255), 2)
            markAttendance(name)

        ret, buffer = cv2.imencode('.jpg', img)
        frame = buffer.tobytes()
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
    cap.release()
@app.route('/')
def home():
    return render_template('index.html')

@app.route('/take_attendance')
def take_attendance():
    return render_template('take_attendance.html')

@app.route('/start_attendance')
def start_attendance():
    time.sleep(1)  # Add a delay before starting the attendance process
    return Response(generate_frames(), mimetype='multipart/x-mixed-replace;
boundary=frame')

@app.route('/stop_attendance')
def stop_attendance():
    time.sleep(1)  # Add a delay before closing the webcam window
    # Release the webcam resources
    cv2.destroyAllWindows()
    return render_template('index.html')

@app.route('/add_person', methods=['GET', 'POST'])
def add_person():
    if request.method == 'POST':
        name = request.form['name']
        image = request.files['image']
        image_path = os.path.join(path, f'{name}.jpg')
```

```
            image.save(image_path)
            img = cv2.imread(image_path)
            images.append(img)
            classNames.append(name)
            encodeListKnown.append(findEncodings([img])[0])
            return 'Person added successfully'

    return render_template('add_person.html')


@app.route('/view_attendance')
def view_attendance():
    attendance_data = []
    with open('Attendance.csv', 'r') as file:
        csv_reader = csv.reader(file)
        for row in csv_reader:
            attendance_data.append(row)

    # Reverse the attendance_data list to show the recorded attendance first
    attendance_data.reverse()

    return render_template('view_attendance.html', attendance_data=attendance_data)

@app.route('/cancel')
def cancel():
    return 'Program has been canceled'

if __name__ == '__main__':
    app.run()
```