Table of Contents

Chapter 1 4

INTRODUCTION 4

1.1 Overview 4

1.2 Problem Statement 6

1.3 Objectives 6

1.4 Proposed System 7

Chapter 2 8

LITERATURE REVIEW 8

2.1 Automated smart attendance system using face recognition 8

2.2 Real Time Smart Attendance Management System Using Face Recognition Techniques 9

2.3 Face Recognition Based Smart Attendance System 9

2.4 Student Attendance System using Face Recognition 10

2.5 Face Augmentation GAN for Deformation-Invariant Face Recognition 11

Chapter 3 12

METHODOLOGY 12

3.1 Python Libraries 13

3.2 Algorithm 14

3.3 System Architecture 15

3.4 Data Flow Diagram 16

3.5 System Requirements 17

Chapter 4 21

IMPLEMENTATION 21

4.1 Installing procedure 21

4.2 Code Snippets 22

Chapter 5 31

RESULTS 31

5.1 Initialization of login page 31

5.2 Output for successful user registration 32
5.3 Facial recognition of user 32
5.4 Logout by the User 33
5.5 Unknown user identified 34
5.6 Fake User 34
CHAPTER 6 36
CONCLUSION AND FUTURE ENHANCEMENTS 36
6.1 Conclusion 36
6.2 Future Enhancement 36
REFERENCES 38

Figures

Fig 1. 1 Face Recognition Attendance System 2
Fig 3. 1 Methodology 13
Fig 3. 2 flowchart of program 15
Fig 3. 3 system architecture 16
Fig 3. 4 Data flow diagram 17

Fig 5. 1 Initialization of login page 31
Fig 5. 2 Output for Succesful User Registration 32
Fig 5. 3 Facial Recognition of User 33
Fig 5. 4 Logout by the User 33
Fig 5. 5 Unknown user identified 34
Fig 5. 6 Fake User 35

Chapter 1

INTRODUCTION

A Face Recognition Attendance System uses cameras and software to automatically identify and record people's attendance by recognizing their faces. This system captures images of individuals as they enter a room or building and compares these images with stored facial data to confirm their identity. It offers high accuracy, minimizing errors compared to manual sign-ins or card swipes. The process is quick, saving time for both users and administrators, and enhances security by reducing the chances of fraudulent attendance marking. This modern approach streamlines attendance management, making it efficient and reliable for schools, offices, and events.

1.1 Overview

A face recognition attendance system is an advanced biometric solution designed to streamline and secure the process of recording attendance in various settings, such as schools, offices, and other organizations. Unlike traditional methods of attendance tracking, which often rely on manual or electronic check-ins, face recognition systems use sophisticated algorithms to automatically identify and verify individuals based on their facial features.

The system operates by capturing images or video footage of individuals as they enter or interact with the system. Using advanced image processing techniques and machine learning algorithms, the system analyzes key facial features, such as the distance between eyes, nose shape, and jawline contours. This data is then compared against a pre-stored database of facial profiles to confirm the individual's identity.

One of the primary advantages of face recognition attendance systems is their high level of accuracy and efficiency. Unlike manual methods that are prone to errors and can be time-consuming, face recognition systems provide real-time processing, significantly reducing the time spent on attendance tracking. This technology also enhances security by minimizing the risk of fraudulent activities, such as buddy punching, where one person clocks in for another.

Moreover, face recognition attendance systems are highly scalable and adaptable to various environments. They can be integrated with existing attendance management software and databases, offering a seamless transition from traditional systems. Additionally, these systems can be configured to accommodate various levels of security, from simple time and attendance tracking to more complex access control applications.

Despite their numerous benefits, face recognition attendance systems also raise certain concerns, particularly related to privacy and data security. The collection and storage of biometric data necessitate stringent measures to protect this sensitive information from unauthorized access and misuse. As such, organizations implementing these systems must ensure compliance with relevant data protection regulations and best practices to safeguard individual privacy.

Face recognition attendance systems represent a modern and efficient approach to managing attendance and access control. By leveraging advanced biometric technology, these systems offer improved accuracy, security, and convenience, though they also require careful consideration of privacy and data security issues.

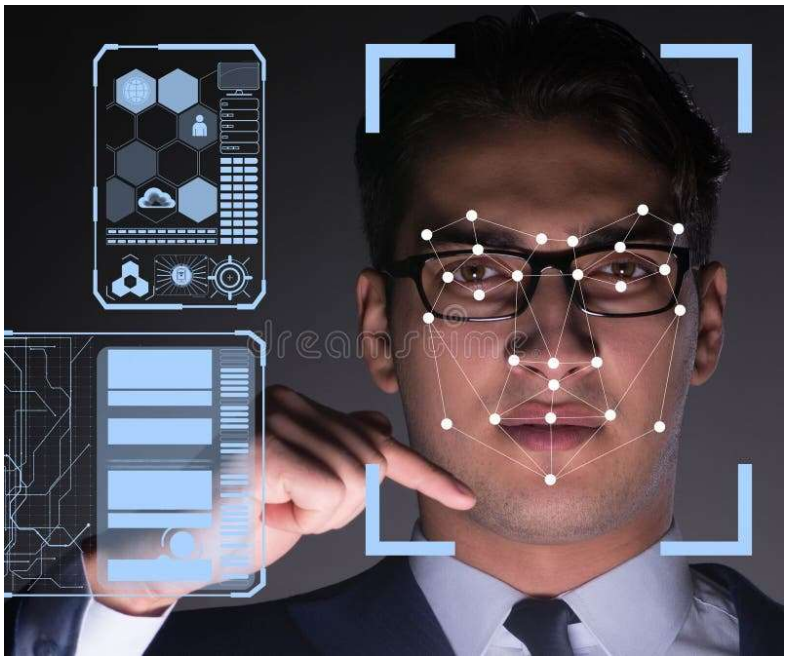

Fig 1. 1 Face Recognition Attendance System

1.2 Problem Statement

The increasing complexity and demands of modern organizational environments have exposed limitations in traditional attendance tracking methods, such as manual sign-ins and card-based systems. These conventional approaches often suffer from inefficiencies, inaccuracies, and security vulnerabilities, leading to issues such as fraudulent attendance records, time theft, and administrative overhead. Furthermore, manual methods are time-consuming and prone to human error, which can compromise the integrity of attendance data. In response to these challenges, there is a growing need for a more reliable, secure, and efficient solution. A face recognition attendance system addresses these problems by leveraging advanced biometric technology to automate and streamline the attendance process.

1.3 Objectives

- Enhance Accuracy: Develop and deploy a system that accurately identifies individuals based on their facial features, minimizing errors associated with manual attendance tracking and reducing instances of buddy punching or other fraudulent activities.
- Increase Efficiency: Streamline the attendance recording process by automating check-ins and check-outs, thereby reducing the time and administrative effort required for manual attendance management and improving overall operational efficiency.
- Improve Security: Ensure a high level of security by using biometric verification to prevent unauthorized access and safeguard against tampering or manipulation of attendance records.
- Facilitate Scalability: Design a system that can be easily scaled and integrated with existing attendance management software and databases, allowing for flexible deployment across different environments, such as schools, offices, and industrial settings.
- Ensure Privacy and Compliance: Implement robust data protection measures to secure biometric data and ensure compliance with relevant privacy regulations and standards, addressing concerns related to the collection, storage, and use of sensitive personal information.

- Provide Real-Time Monitoring and Reporting: Offer features for real-time monitoring of attendance and generate comprehensive reports to support data-driven decision-making and management oversight.
- Enhance User Experience: Develop an intuitive and user-friendly interface that simplifies interaction with the system for both administrators and users, ensuring a smooth and positive experience for all parties involved.

1.4 Proposed System

The proposed face recognition attendance system aims to revolutionize traditional attendance tracking by integrating advanced biometric technology to enhance accuracy, efficiency, and security. The system features high-resolution cameras positioned at key entry and exit points to capture real-time facial images. These images are processed by sophisticated facial recognition algorithms, which analyse unique facial features and create digital profiles for each individual. The system includes a secure, encrypted database for storing facial profiles and attendance records, ensuring data protection and compliance with privacy regulations. An intuitive user interface allows administrators to monitor attendance, generate reports, and manage profiles with ease. Key functionalities include automated check-ins and check-outs, real-time monitoring, and alerts for any security anomalies. The system is designed for easy integration with existing attendance management software and HR systems, facilitating a smooth transition and effective data management. Implementation involves a pilot test to refine the system, followed by full deployment, user training, and ongoing support. Regular performance monitoring and user feedback will drive continuous optimization, ensuring the system remains accurate, efficient, and responsive to organizational needs.

Chapter 2

LITERATURE REVIEW

A literature review on AI healthcare chatbots examines the advancements in AI-driven conversational agents designed for medical assistance, patient engagement, and health monitoring. It highlights the effectiveness, challenges, and ethical considerations of deploying chatbots in clinical settings, emphasizing their potential to enhance patient care and streamline healthcare services. Section 2.1 Automated smart attendance system using face recognition, 2.2 Artificial Intelligence Healthcare Chatbot System, 2.3 AI-based Healthcare Chatbot, 2.4 Personal Healthcare Chatbot for Medical Suggestions Using Artificial Intelligence and Machine Learning, 2.5 The Development and Use of Chatbots in Public Health, 2.6 A Survey of Health Care Chatbot for Patient Support, 2.7 Disease Prediction Based On Image Processing Using Chatbot – MEDIKNOW, 2.8 A Healthcare Chatbot System Using Python and NLP.

2.1 Automated smart attendance system using face recognition

In the human body, the face is the most crucial factor in identifying each person as it contains many vital details. There are different prevailing methods to capture person's presence like biometrics to take attendance which is a time-consuming process. This paper develops a model to classify each character's face from a captured image using a collection of rules i.e., LBP algorithm to record the student attendance. LBP (Local Binary Pattern) is one among the methods and is popular as well as effective technique used for the image representation and classification and it was chosen for its robustness to pose and illumination shifts. The proposed ASAS (Automated Smart Attendance System) will capture the image and will be compared to the image stored in the database. The database is updated upon the enrolment of the student using an automation process that also includes name and rolls number. ASAS marks individual attendance, if the captured image matches the image in the database i.e., if both images are identical. The proposed algorithm reduces effort and captures day-to-day actions of managing each student and also makes it simple to mark the presence [1].

2.2 Real Time Smart Attendance Management System Using Face Recognition Techniques

The management of the attendance can be a great burden on the teachers if it is done by hand. To resolve this problem, smart and auto attendance management system is being utilized. But authentication is an important issue in this system. The smart attendance system is generally executed with the help of biometrics. Face recognition is one of the biometric methods to improve this system. Being a prime feature of biometric verification, facial recognition is being used enormously in several such applications, like video monitoring and CCTV footage system, an interaction between computer & humans and access systems presents indoors and network security. By utilizing this framework, the problem of proxies and students being marked present even though they are not physically present can easily be solved. The main implementation steps used in this type of system are face detection and recognizing the detected face. This paper proposes a model for implementing an automated attendance management system for students of a class by making use of face recognition technique, by using Eigenface values, Principal Component Analysis (PCA) and Convolutional Neural Network (CNN). After these, the connection of recognized faces ought to be conceivable by comparing with the database containing student's faces. This model will be a successful technique to manage the attendance and records of students [2].

2.3 Face Recognition Based Smart Attendance System

Education institutes today are concerned about the consistency of students ' performance. One cause of this decrease in student performance is the inadequate attendance. There are several ways to mark your attendance, the most common ways to sign or call the students. It took longer and was problematic. From now on, a computer-based student attendance checking system is required that supports the faculty to keep records of attendance. We have used an intelligent attendance system based on face recognition in this project. We have proposed to implement a "Smart Attendance System for Face Recognition" through these large applications are incorporated. The present implementation includes facial identification that is time saving and eradicates the possibilities of proxy attendance due to the facial authorization. This system can now be used in an area in which participation plays an important role. Raspberry Pi, Open CV and Dlib using

python are the basic requirements for this system. The system implemented uses LBPH face recognizer to identify the face of the person in real time. Eigen faces and Fisher faces are affected both by light and we cannot ensure perfect light conditions in real life. An improvement in the LBPH faces recognizer to overcome this problem. This system compares the image of the test and the training image and determines who is and is not present. The attendance data is stored in an excel sheet that is automatically updated in the system. If a student is absent a message will be automatically sent to their parent's phone number using GSM. Students can check their attendance using an Android application that we have developed using MIT app Inventor [3].

2.4 Student Attendance System using Face Recognition

Face recognition is among the most productive image processing applications and has a pivotal role in the technical field. Recognition of the human face is an active issue for authentication purposes specifically in the context of attendance of students. Attendance system using face recognition is a procedure of recognizing students by using face biostatistics based on the high-definition monitoring and other computer technologies. The development of this system is aimed to accomplish digitization of the traditional system of taking attendance by calling names and maintaining pen-paper records. Present strategies for taking attendance are tedious and time-consuming. Attendance records can be easily manipulated by manual recording. The traditional process of making attendance and present biometric systems is vulnerable to proxies. This paper is therefore proposed to tackle all these problems. The proposed system makes the use of Haar classifiers, KNN, CNN, SVM, Generative adversarial networks, and Gabor filters. After face recognition attendance reports will be generated and stored in excel format. The system is tested under various conditions like illumination, head movements, the variation of distance between the student and cameras. After vigorous testing overall complexity and accuracy are calculated. The Proposed system proved to be an efficient and robust device for taking attendance in a classroom without any time consumption and manual work. The system developed is cost-efficient and need less installation [4].

2.5 Face Augmentation GAN for Deformation-Invariant Face Recognition

Substantial improvements have been achieved in the field of face recognition due to the successful application of deep neural networks. However, existing methods are sensitive to both the quality and quantity of the training data. Despite the availability of large-scale datasets, the long tail data distribution induces strong biases in model learning. In this paper, we present a Face Augmentation Generative Adversarial Network (FA-GAN) to reduce the influence of imbalanced deformation attribute distributions. We propose to decouple these attributes from the identity representation with a novel hierarchical disentanglement module. Moreover, Graph Convolutional Networks (GCNs) are applied to recover geometric information by exploring the interrelations among local regions to guarantee the preservation of identities in face data augmentation. Extensive experiments on face reconstruction, face manipulation, and face recognition demonstrate the effectiveness and generalization ability of the proposed method [5].

Chapter 3

METHODOLOGY

The methodology for developing a face recognition attendance system involves several key stages, each focusing on different aspects of the system's design, development, and implementation. Initially, a comprehensive requirements analysis is conducted to understand the specific needs of the organization and define the system's functional and non-functional requirements. Following this, the system architecture is designed, incorporating high-resolution cameras for capturing facial images, a robust server or cloud infrastructure for processing and storing data, and an intuitive user interface for administrators and users. The development phase employs advanced facial recognition algorithms, utilizing libraries such as OpenCV, Dlib, and face recognition for accurate and efficient image processing. Data security and privacy are paramount, so encryption techniques and secure access controls are implemented to protect biometric data. The system undergoes rigorous testing, starting with a pilot test to evaluate performance and gather feedback, followed by iterative refinements. Deployment is carefully managed to ensure smooth integration with existing systems and minimal disruption. User training and support are provided to facilitate adoption and address any issues. Continuous monitoring and user feedback guide ongoing optimization, ensuring the system remains accurate, efficient, and responsive to evolving organizational needs.

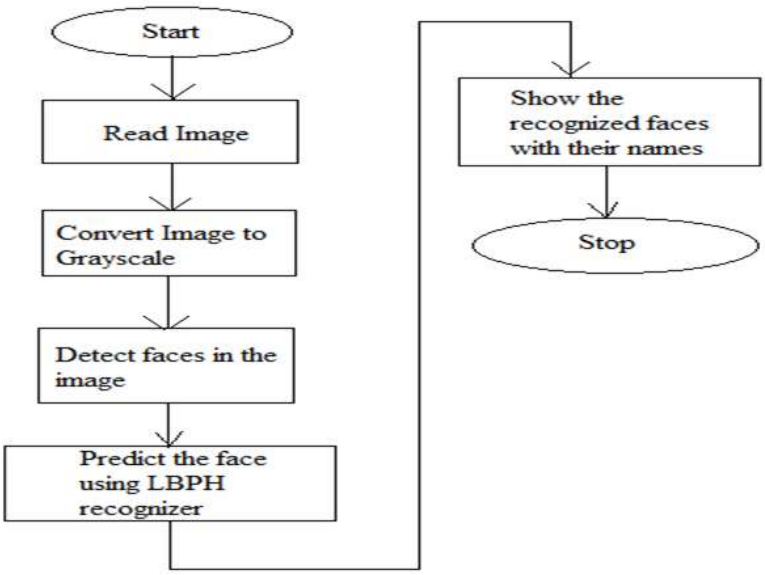

Fig 3.1 Methodology

3.1 Python Libraries

A python library is a collection of functions for specific operations. They are especially effective for accessing the pre-written frequently used codes, instead of writing them from scratch every single time. Below flowchart depicts about the libraries of Python.

OpenCV: A powerful library for real-time computer vision and image processing tasks.

Dlib: A toolkit containing machine learning algorithms and tools for creating complex software in C++ and Python, commonly used for face detection and face recognition.

face_recognition: A simple and powerful library built on top of Dlib for face recognition in Python.

os.path: A module in Python's standard library that provides functions for interacting with the file system, enabling operations such as path manipulation, file existence checking, and directory traversal.

tkinter: A standard GUI (Graphical User Interface) library in Python used to create desktop applications, offering a variety of widgets like buttons, labels, and text fields for building user interfaces.

pickle: A module in Python's standard library used for serializing and deserializing Python object structures, allowing for the saving and loading of Python objects to and from files.

messagebox: A module in tkinter that provides a set of standard dialog boxes, such as error, warning, and information boxes, for displaying messages to the user in a graphical user interface.

3.2 Algorithm

The algorithm for a facial recognition attendance system begins with image capture, where high-resolution cameras placed at entry and exit points continuously record facial images of individuals. These images are preprocessed using techniques such as scaling, normalization, and noise reduction to enhance quality. The preprocessed images are then fed into a facial detection module, typically leveraging libraries like OpenCV or Dlib, to locate faces within the image frames. Once detected, the facial regions are extracted and passed to a feature extraction module, which uses advanced algorithms to generate unique facial feature vectors. These vectors are compared against a database of stored facial profiles using a matching algorithm, such as a nearest neighbor search, to identify the individual. If a match is found, the system records the attendance timestamp in a secure database. The system also incorporates error handling and real-time alerts to address any discrepancies or unauthorized access attempts, ensuring robust and reliable attendance tracking.

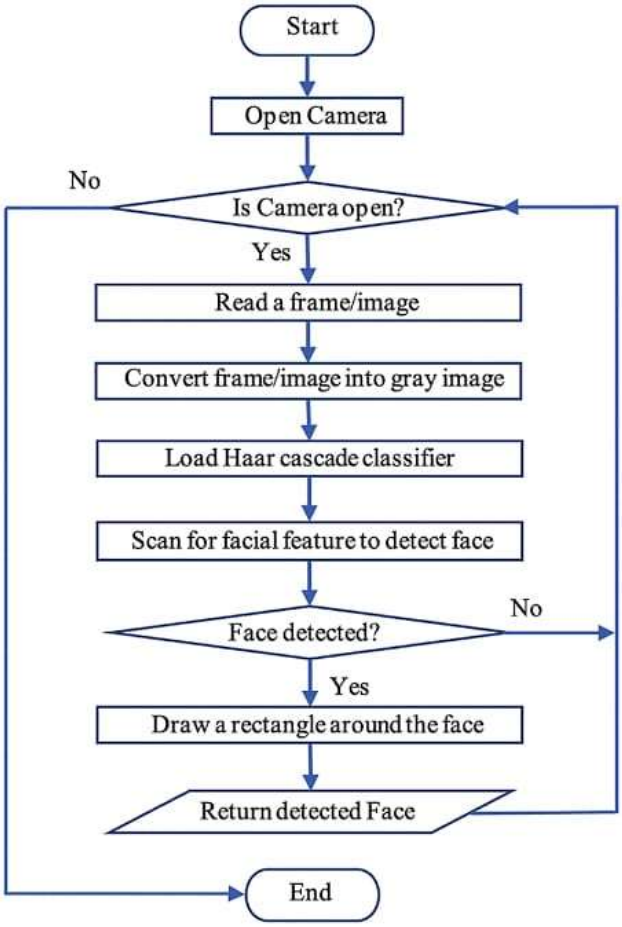

Fig 3. 2 Flowchart of methodology

3.3 System Architecture

To create an AI healthcare chatbot using VS Code, you'll need a system architecture that includes several key components. Start with a user interface where users interact with the chatbot through web. The backend application server, managed through an API gateway, processes user requests and handles different services. For natural language understanding, integrate an NLP engine that recognizes user intents and extracts relevant information. The Chabot's responses are powered by a retrieval-based model that searches a knowledge base, indexed by search technologies like Elasticsearch. Connect to external systems via APIs, and use databases (both relational and NoSQL) for data storage. Ensure robust security measures with encryption and compliance tools, and deploy using containerization (Docker) and orchestration (Kubernetes).

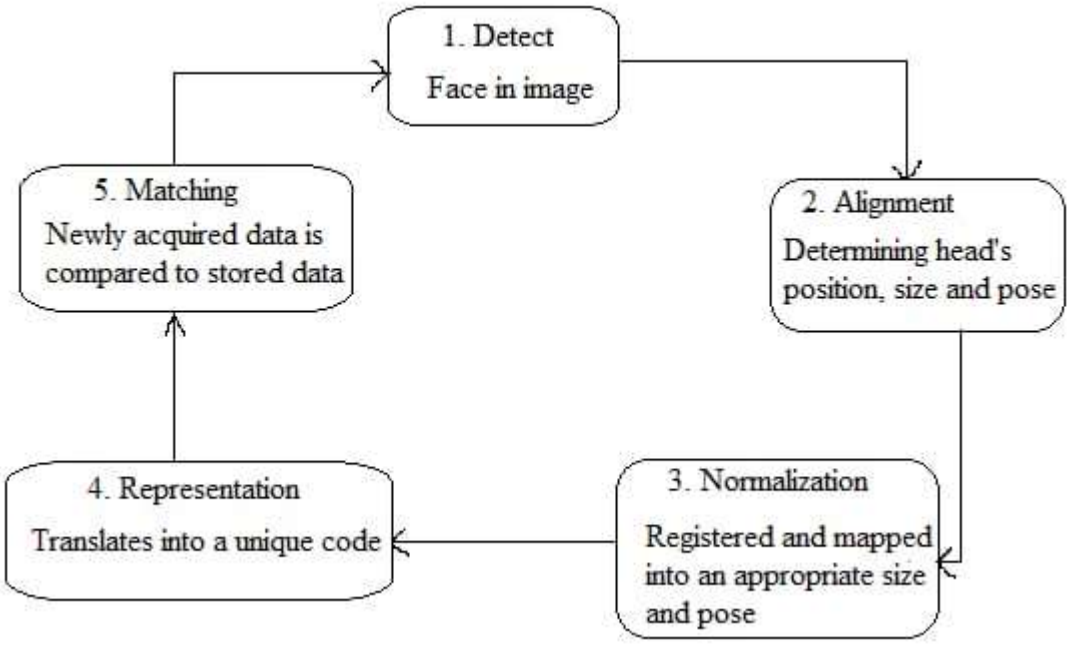

Fig 3. 3 system architecture

3.4 Data Flow Diagram

A Data Flow Diagram (DFD) for a "Face Recognition Attendance System" can be outlined in two primary levels to illustrate the flow of data and processing within the system.

At the Context Diagram (Level 0), the Face Recognition Attendance System interacts with two main external entities: the User (such as an employee or student) and the Administrator. The User provides face images and identification data to the system, which processes this information to confirm attendance and sends back an attendance confirmation. The Administrator interacts with the system to configure settings and manage user data, while the system provides them with attendance reports and alerts.

In the Level 1 DFD, the system's internal processes are detailed. The process begins with the Capture Face Image component, which receives face images from users. This image is then sent to the Process Image stage, where facial recognition technology matches it with data stored in the User Database. Upon successful recognition, the system proceeds to the Update Attendance process, recording the attendance information in the Attendance Records database. Additionally,

the Generate Reports process allows the Administrator to request and receive various reports and alerts based on the attendance data.

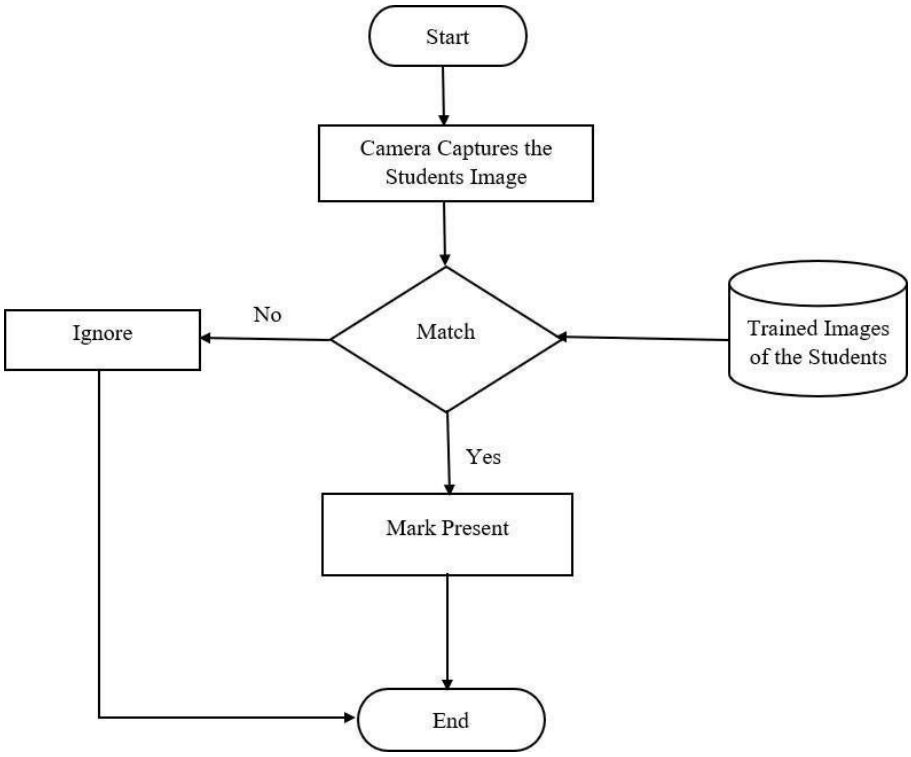

Fig 3. 4 Data flow diagram

3.5 System Requirements

Hardware Requirements:

1. Cameras:

   - High-resolution cameras capable of capturing clear face images in various lighting conditions.
   - Integration with the system for real-time face capture (e.g., USB webcams, IP cameras).

2. Server Hardware:

- A server with sufficient processing power (CPU), memory (RAM), and storage to handle image processing, data storage, and system operations.
- Specifications depend on the scale of the system and expected load (e.g., multi-core processors, 16GB+ RAM, SSD storage).

3. Workstations:

- Computers for administrators and users to interact with the system, including a web browser or application interface.
- These should meet minimum system requirements for running the user interface and management tools.

4. Networking Equipment:

- Reliable network infrastructure, including routers, switches, and network cables to ensure stable connectivity between cameras, servers, and workstations.
- Adequate bandwidth to handle the data traffic from face image capture and transmission.

5. Backup and Storage Devices:

- External storage or backup solutions for regular data backups (e.g., NAS devices, cloud storage services).
- Devices for data recovery in case of hardware failure.

6. Power Backup:

- Uninterruptible Power Supply (UPS) systems to ensure continuous operation and protect against power outages.

7. Environmental Controls:

- Proper ventilation and cooling systems for server rooms to maintain optimal operating conditions

Software Requirements:

1. Operating System:

   - Compatible with Windows, macOS, or Linux for server-side software.
   - Support for mobile operating systems (iOS, Android) if there are mobile applications involved.

2. Face Recognition Software:

   - Software capable of capturing and processing face images for recognition (e.g., OpenCV, Dlib, commercial APIs like Microsoft Azure Face API or AWS Rekognition).
   - Algorithms for facial feature extraction, matching, and recognition.

3. User Interface:

   - A web-based or mobile-based user interface for both users and administrators, designed for ease of use and accessibility.
   - Interfaces for real-time feedback, report generation, and system configuration.

4. Security Software:

   - Encryption tools for securing data in transit and at rest (e.g., SSL/TLS for web communication, AES for data storage).
   - Authentication and authorization mechanisms for secure access control.

5. Data Analytics and Reporting Tools:

   - Software for generating and exporting attendance reports (e.g., data visualization libraries, reporting tools).

6. Integration Interfaces:

- APIs or middleware for integrating with other systems or services if needed (e.g., HR systems, school management systems).

Chapter 4

IMPLEMENTATION

4.1 Installing procedure

Install PyCharm:

- Download and install PyCharm Community Edition from the JetBrains website.
- Launch PyCharm and create a new project.

Installing python and all required libraries

- Tkinter: For the GUI (usually included with Python standard libraries).
- OpenCV: For image capture and processing.
- Dlib: For facial recognition.
- Pickle: For saving and loading data.
- face_recognition: library is used for face detection and recognition.

Run the application

- Start PyCharm:
  - Open main.py in PyCharm.
- Run the Application:
  - Click the run button in PyCharm or use the terminal command python main.py.
- Test the System:
  - The application window should appear. Click "Start Attendance" to begin face recognition using your webcam.

4.2 Code Snippets

```
import os.path

import datetime

import pickle

import tkinter as tk

import cv2

from PIL import Image, ImageTk

import face_recognition

import util

from test import test

class App:

    def _init_(self):

        self.main_window = tk.Tk()

        self.main_window.geometry("1200x520+350+100")

        self.login_button_main_window = util.get_button(self.main_window, 'login', 'green',
self.login)

        self.login_button_main_window.place(x=750, y=200)

        self.logout_button_main_window = util.get_button(self.main_window, 'logout', 'red',
self.logout)

        self.logout_button_main_window.place(x=750, y=300)
```

```
        self.register_new_user_button_main_window = util.get_button(self.main_window, 'register
new user', 'gray',

        self.register_new_user, fg='black')

        self.register_new_user_button_main_window.place(x=750, y=400)

        self.webcam_label = util.get_img_label(self.main_window)

        self.webcam_label.place(x=10, y=0, width=700, height=500)

        self.add_webcam(self.webcam_label)

        self.db_dir = './db'

        if not os.path.exists(self.db_dir):

            os.mkdir(self.db_dir)

        self.log_path = './log.txt'

    def add_webcam(self, label):

        if 'cap' not in self._dict_:

            self.cap = cv2.VideoCapture(0)

        self._label = label

        self.process_webcam()

    def process_webcam(self):

        ret, frame = self.cap.read()
```

```
        self.most_recent_capture_arr = frame

        img_ = cv2.cvtColor(self.most_recent_capture_arr, cv2.COLOR_BGR2RGB)

        self.most_recent_capture_pil = Image.fromarray(img_)

        imgtk = ImageTk.PhotoImage(image=self.most_recent_capture_pil)

        self._label.imgtk = imgtk

        self._label.configure(image=imgtk)

        self._label.after(20, self.process_webcam)

    def login(self):

        label = test(

                image=self.most_recent_capture_arr,

                model_dir='C:\\Users\\Rahul Poojary\\PycharmProjects\\pythonProject4\\Silent-Face-
Anti-Spoofing-master\\Resources\\Anti_spoof_models',

                device_id=0

                )

        if label == 1:

            name = util.recognize(self.most_recent_capture_arr, self.db_dir)

            if name in ['unknown_person', 'no_persons_found']:

                util.msg_box('Ups...', 'Unknown user. Please register new user or try again.')

            else:
```

```
            util.msg_box('Welcome back !', 'Welcome, {}.'.format(name))
            with open(self.log_path, 'a') as f:
                f.write('{},{},in\n'.format(name, datetime.datetime.now()))
                f.close()
        else:
            util.msg_box('Hey, you are a spoofer!', 'You are fake !')
    def logout(self):
        label = test(
            image=self.most_recent_capture_arr,
            model_dir='C:\\Users\\Rahul Poojary\\PycharmProjects\\pythonProject4\\Silent-Face-Anti-Spoofing-master\\Resources\\Anti_spoof_models',
            device_id=0
            )
        if label == 1:
            name = util.recognize(self.most_recent_capture_arr, self.db_dir)
            if name in ['unknown_person', 'no_persons_found']:
                util.msg_box('Ups...', 'Unknown user. Please register new user or try again.')
            else:
                util.msg_box('Hasta la vista !', 'Goodbye, {}.'.format(name))
```

```
            with open(self.log_path, 'a') as f:
                f.write('{},{},out\n'.format(name, datetime.datetime.now()))
                f.close()
        else:
            util.msg_box('Hey, you are a spoofer!', 'You are fake !')
    def register_new_user(self):
        self.register_new_user_window = tk.Toplevel(self.main_window)
        self.register_new_user_window.geometry("1200x520+370+120")
        self.accept_button_register_new_user_window =
util.get_button(self.register_new_user_window, 'Accept', 'green', self.accept_register_new_user)
        self.accept_button_register_new_user_window.place(x=750, y=300)
        self.try_again_button_register_new_user_window =
util.get_button(self.register_new_user_window, 'Try again', 'red',
self.try_again_register_new_user)
        self.try_again_button_register_new_user_window.place(x=750, y=400)
        self.capture_label = util.get_img_label(self.register_new_user_window)
        self.capture_label.place(x=10, y=0, width=700, height=500
        self.add_img_to_label(self.capture_label)
        self.entry_text_register_new_user = util.get_entry_text(self.register_new_user_window)
        self.entry_text_register_new_user.place(x=750, y=150)
```

```
        self.text_label_register_new_user = util.get_text_label(self.register_new_user_window,
'Please, \ninput username:')

        self.text_label_register_new_user.place(x=750, y=70)

    def try_again_register_new_user(self):

        self.register_new_user_window.destroy()

    def add_img_to_label(self, label):

        imgtk = ImageTk.PhotoImage(image=self.most_recent_capture_pil)

        label.imgtk = imgtk

        label.configure(image=imgtk)

        self.register_new_user_capture = self.most_recent_capture_arr.copy()

    def start(self):

        self.main_window.mainloop()

    def accept_register_new_user(self):

        name = self.entry_text_register_new_user.get(1.0, "end-1c")

        embeddings = face_recognition.face_encodings(self.register_new_user_capture)[0]

        file = open(os.path.join(self.db_dir, '{}.pickle'.format(name)), 'wb')

        pickle.dump(embeddings, file)

        util.msg_box('Success!', 'User was registered successfully !')

        self.register_new_user_window.destroy()
```

```
if _name_ == "_main_":
    app = App()
    app.start()
import os
import pickle
import tkinter as tk
from tkinter import messagebox
import face_recognition
def get_button(window, text, color, command, fg='white'):
    button = tk.Button(
                        window,
                        text=text,
                        activebackground="black",
                        activeforeground="white",
                        fg=fg,
                        bg=color,
                        command=command,
                        height=2,
                        width=20,
```

```
                    font=('Helvetica bold', 20)

                )

    return button

def get_img_label(window):

    label = tk.Label(window)

    label.grid(row=0, column=0)

    return label

def get_text_label(window, text):

    label = tk.Label(window, text=text)

    label.config(font=("sans-serif", 21), justify="left")

    return label

def get_entry_text(window):

    inputtxt = tk.Text(window,

                       height=2,

                       width=15, font=("Arial", 32))

    return inputtxt

def msg_box(title, description):

    messagebox.showinfo(title, description)

def recognize(img, db_path):
```

```
    # it is assumed there will be at most 1 match in the db
    embeddings_unknown = face_recognition.face_encodings(img)
    if len(embeddings_unknown) == 0:
        return 'no_persons_found'
    else:
        embeddings_unknown = embeddings_unknown[0]
    db_dir = sorted(os.listdir(db_path))
    match = False
    j = 0
    while not match and j < len(db_dir):
        path_ = os.path.join(db_path, db_dir[j])
        file = open(path_, 'rb')
        embeddings = pickle.load(file)
        match = face_recognition.compare_faces([embeddings], embeddings_unknown)[0]
        j += 1
    if match:
        return db_dir[j - 1][:-7]
    else:
        return 'unknown_person'
```

Chapter 5

RESULTS

The Face Recognition Attendance System project effectively demonstrates the integration of facial recognition technology with automated attendance management. Upon running the system, it successfully captures and analyzes real-time video from a webcam to identify and recognize individuals, recording attendance promptly and accurately. The user-friendly interface built with Tkinter facilitates easy operation, while the system provides immediate visual feedback by labeling recognized faces and flagging unknown ones. These results showcase the system's ability to enhance efficiency, reduce errors, and streamline attendance tracking, making it a valuable tool for various environments such as educational institutions and workplaces.

5.1 Initialization of login page

The image shows a login page interface for what appears to be an attendance system. On the left side is a photo of a man wearing a white shirt with a lanyard. He has short dark hair and facial hair. The background seems to be an office or workspace with exposed ceiling elements visible.On the right side of the image are three buttons: a green "login" button at the top, a red "logout" button in the middle, and a gray "register new user" button at the bottom. This layout suggests it's an interface for an attendance or access control system where users can log in, log out, or register as a new user.

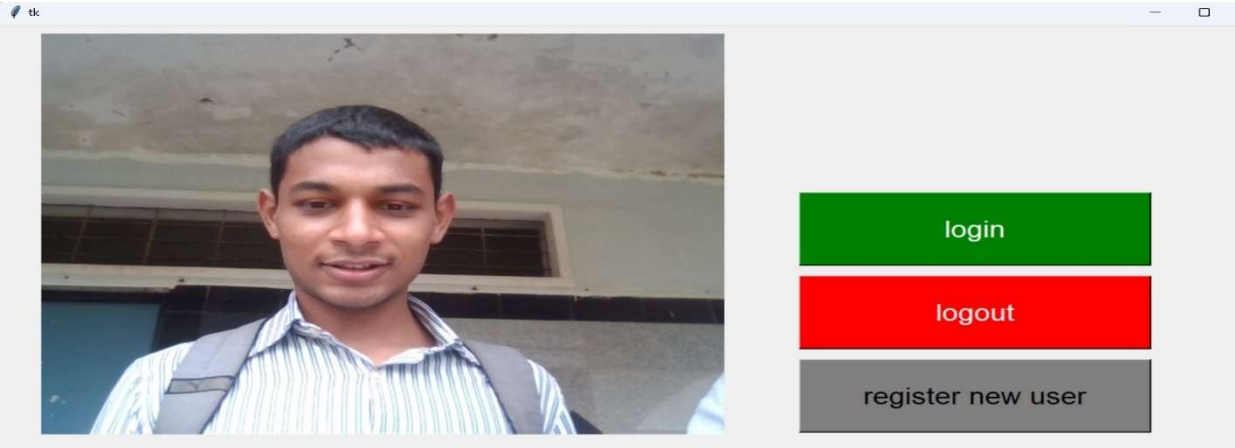

Fig 5. 1 Initialization of Login page

5.2 Output for successful user registration

The image shows a face recognition attendance system interface with a success message. The interface displays a photo of a man wearing a white shirt with a lanyard in an office setting. A pop-up window in the center of the image indicates "Success!" with the message "User was registered successfully!" This suggests that the system has successfully registered a new user, likely using the photo and facial data of the person shown in the image. This interface appears to be part of an attendance or access control system that utilizes facial recognition technology to register and authenticate users. The success message implies that the registration process has been completed, allowing the newly registered user to now use the login function for attendance tracking or access control purposes.

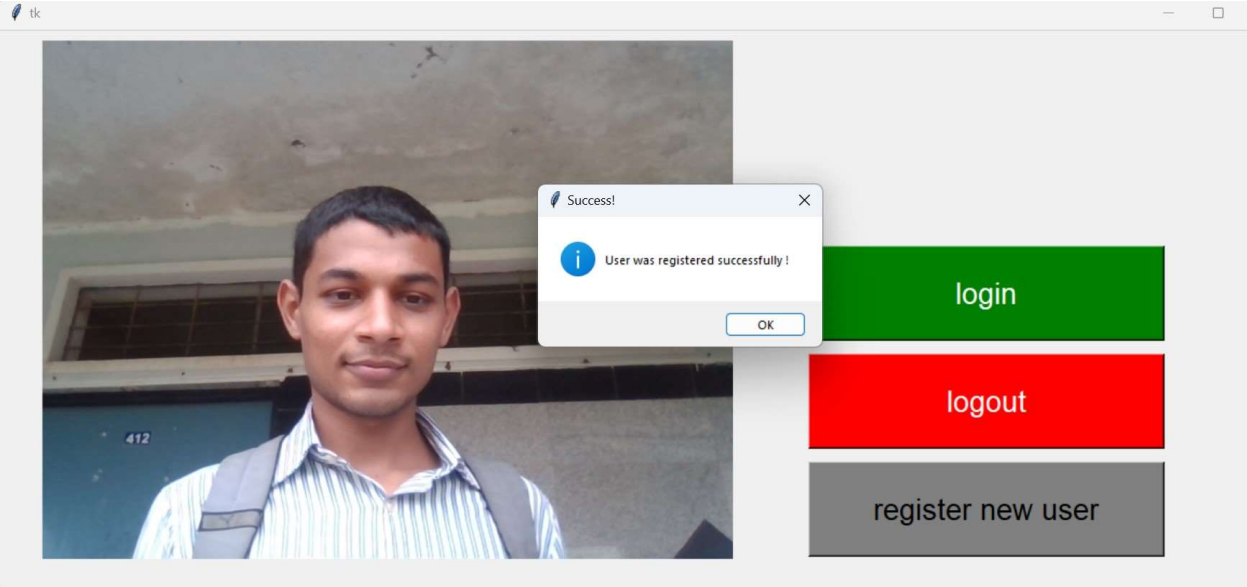

Fig 5. 2 Output for successful user registration

5.3 Facial recognition of user

The image shows the result of a successful login on a facial recognition attendance system.A pop-up window in the center of the image says "Welcome back!" with the message "Welcome, Arun." This indicates that the system has successfully recognized the user as Arun and logged him in.This result suggests that the facial recognition system has matched the current image with a previously registered user profile for Arun, allowing him to log in without needing to enter additional

credentials. The welcome message implies that the system is now ready to record Arun's attendance or grant him access to whatever the system controls.

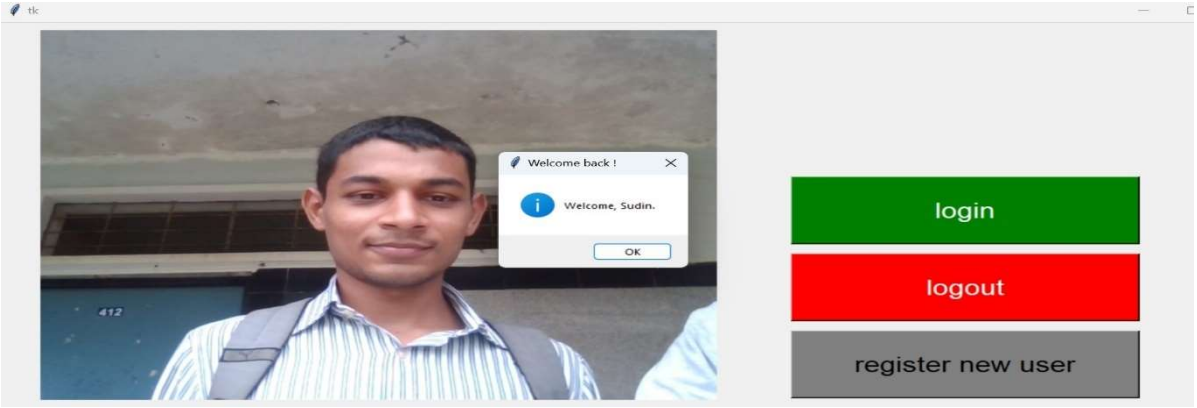

Fig 5. 3 successful facial recognition of user

5.4 Logout by the User

The image shows the result of a successful logout from a facial recognition attendance system.A pop-up window in the center of the image says "Hasta la vista !" (Spanish for "See you later!") with the message "Goodbye, Arun." This indicates that the system has successfully logged out the user named Arun. This result suggests that the facial recognition system has processed a logout request for Arun, likely initiated by clicking the red "logout" button. The farewell message confirms that Arun's session has ended and he has been signed out of the attendance or access control system. The system is now ready for the next user to log in or for a new user to be registered.

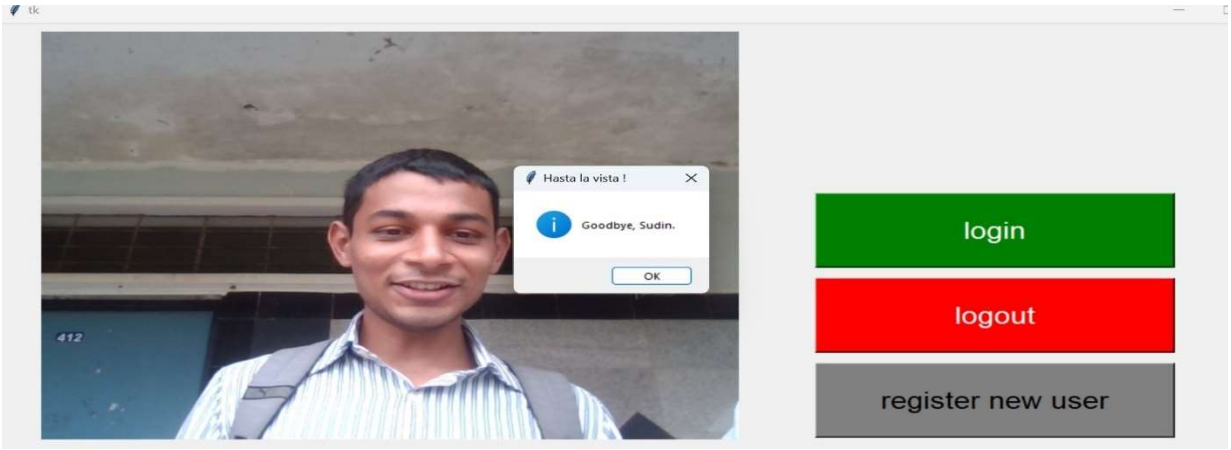

Fig 5. 4 Successful logout by user

5.5 Unknown user identified

The image shows the result of an attempted login by an unregistered user on a facial recognition attendance system. A pop-up window in the center of the image says "Ups..." with the message "Unknown user. Please register new user or try again." This indicates that the system does not recognize the person in the photo as a registered user.

This result suggests that the facial recognition system has failed to match the current image with any existing user profiles in its database. The system is prompting the user to either register as a new user (likely using the "register new user" button) or to try the login process again in case there was an error in capturing the image or in the facial recognition process.

The message implies that this is either a new employee or visitor who hasn't been registered in the system yet, or possibly an existing user who the system failed to recognize for some reason.

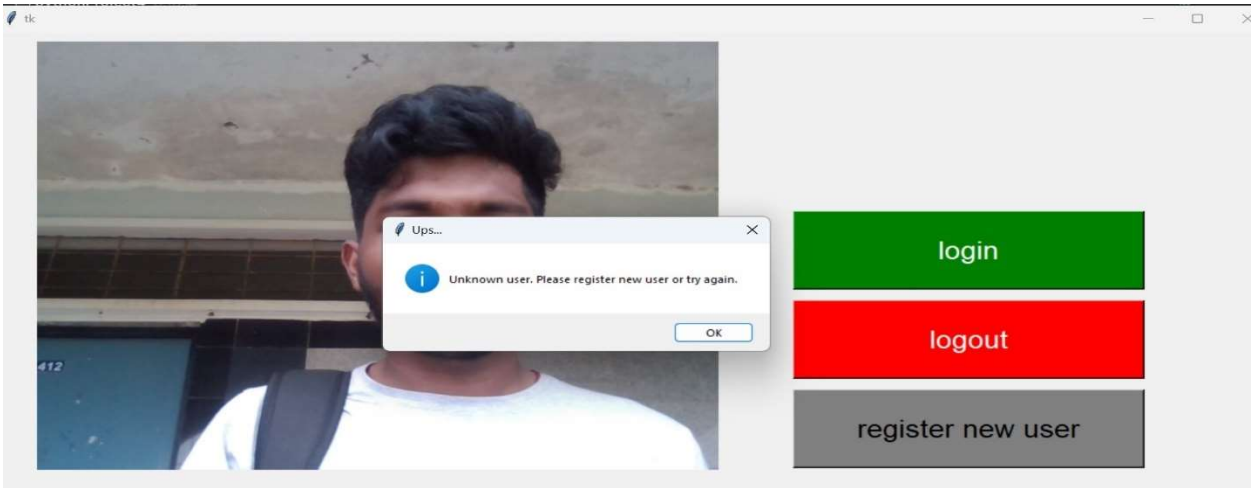

Fig 5. 5 Unknown user identified

5.6 Fake User

The image shows the result of an attempted login by a photo of the person in the facial recognition attendance system. A pop-up window in the center of the image says "Hey, you are a spoofer!" with the message "You are fake." This indicates that the system does not recognize the person in the photo as a live user.

This result suggests that the facial recognition system has detected that the user is trying to spoof the system.

The message implies that this is an unauthorized individual is attempting to gain access by presenting a fake image or using a method to spoof the system..

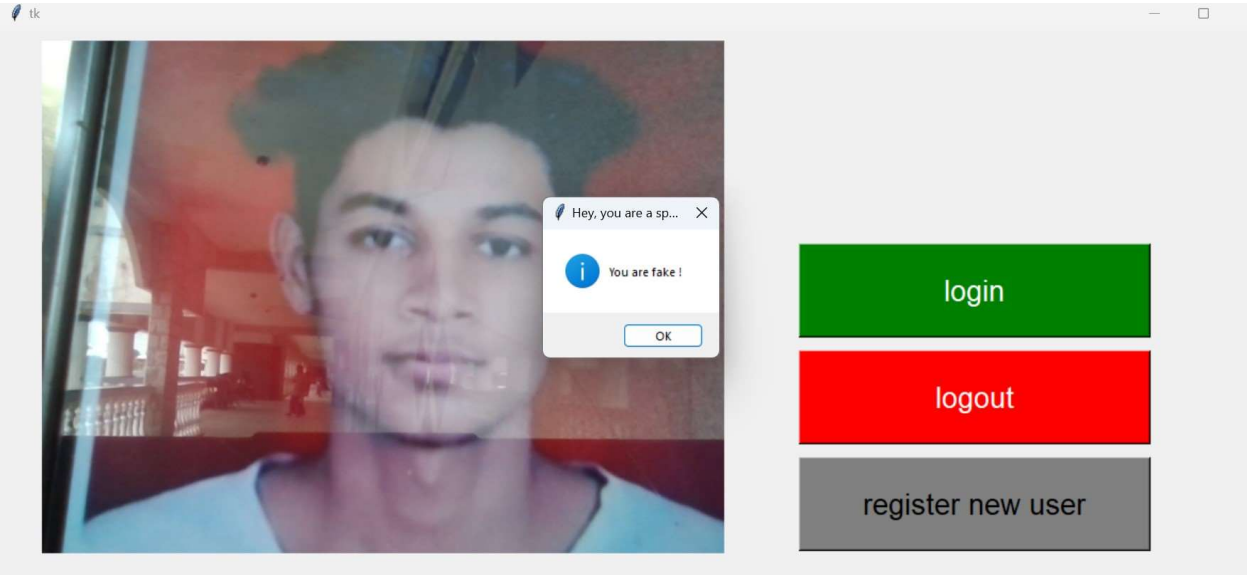

Fig 5. 1 Fake user

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENTS

6.1 Conclusion

In conclusion, the Face Recognition Attendance System is a modern and efficient way to keep track of attendance. By using facial recognition technology, it automates the process of marking who is present, making it faster and more accurate than traditional methods. With a user-friendly interface created with Tkinter, and real-time image processing done through OpenCV, the system is easy to use and dependable. It helps reduce errors and saves time, making it useful in schools, offices, and other settings. Overall, this system simplifies attendance tracking and has the potential to be improved and adapted for even more uses in the future.

6.2 Future Enhancement

To enhance this Face Recognition Attendance system in the future, consider the following improvements:

1. Better Accuracy and Speed:
    - New Technology: Use more advanced technology to make facial recognition more accurate and faster.
    - Extra Features: Add other ways to identify people, like using voice or eye scans, for more reliable results.
2. Easier to Use:
    - Mobile Access: Create apps or mobile-friendly websites so users can manage their attendance from their phones.
3. Improved Data Management:
    - Cloud Storage: Store attendance data online for better access and backup.

- Advanced Reports: Offer detailed reports and analysis on attendance trends.

4. Scalability:

- System Integration: Make it easier to connect with other systems like HR or school management tools.
- Handle Growth: Build a system that can easily handle more users and data.

5. Better Security:

- Encrypt Data: Use strong encryption to protect data.
- Follow Rules: Make sure the system follows all data protection laws.

6. More Features:

- Automatic Check-In: Allow automatic check-ins when users enter certain areas.
- Fix Errors: Improve the system's ability to handle errors and poor quality images.

7. Adapt to Different Environments:

- Lighting: Make the system work well in different lighting conditions.
- Weatherproof Cameras: Use cameras that can handle outdoor or harsh conditions.

8. Real-Time Alerts:

- Notifications: Send instant alerts or reminders about attendance issues or missed check-ins.

9. Use of New Technologies:

- AI: Use artificial intelligence to make the system smarter and more adaptive.
- IoT: Integrate with other smart devices for additional features like automated entry control

REFERENCES

[1] Preethi, K. and Vodithala, S., 2021, May. Automated smart attendance system using face recognition. In 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 1552-1555). IEEE.

[2] Sawhney, S., Kacker, K., Jain, S., Singh, S.N. and Garg, R., 2020, January. Real-time smart attendance system using face recognition techniques. In 2020 9th international conference on cloud computing, data science & engineering (Confluence) (pp. 522-525). IEEE.

[3] Alhanaee, K., Alhammadi, M., Almenhali, N. and Shatnawi, M., 2021. Face recognition smart attendance system using deep transfer learning. Procedia Computer Science, 192, pp.4093-4102.

[4] Dev, S. and Patnaik, T., 2020, September. Student attendance system using face recognition. In 2020 international conference on smart electronics and communication (ICOSEC) (pp. 90-96). IEEE.

[5] Luo, M., Cao, J., Ma, X., Zhang, X. and He, R., 2021. FA-GAN: Face augmentation GAN for deformation-invariant face recognition. IEEE Transactions on Information Forensics and Security, 16, pp.2341-2355.