

# Department of Computer Science and Engineering

**Global Campus, Jakkasandra Post, Kanakapura Taluk, Ramanagara District, Pin Code: 562 112**

**2023-2024**

**Fourth Semester Progress Report on**

**Facial Recognition Attendance System**

**Submitted for the partial fulfilment of Project Centric Learning activity of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted by**

**YASH KUMAR JAIN M**

**22BTRCN327**

**VARSHITHA Y**

**22BTRCN313**

**ANKIT CHOUDHARY**

**22BTRCN025**

**AFROJ ALAM**

**22BTRSN062**

**Under the guidance of**

**Dr. GAURAV KUMAR,**

Assistant Professor

Department of Computer Science & Engineering

School of Computer Science and Engineering

JAIN (Deemed-to-be University)



# Department of Computer Science and Engineering

**Global Campus, Jakkasandra Post, Kanakapura Taluk, Ramanagara District, Pin Code: 562 112**

# CERTIFICATE

This is to certify that the project centric learning work titled **Facial Recognition Attendance System** is carried out by **YASH KUMAR JAIN (22BTRCN327), VARSHITHA-Y(22BTRCN313),ANKIT-CHOUDHARY(22BTRCN025),AFROZ ALAM(22BTRSN062),LOKNATH ROY(22BTLCN004),** bonafide students of Bachelor of Technology at the Faculty of Engineering & Technology, Jain Deemed-to-be University, Bangalore in partial fulfillment for the project centric learning activity of degree in Bachelor of Technology in Computer Science & Engineering, during the year **2023-2024**.

|  |  |  |
| --- | --- | --- |
| ***Dr Gaurav Kumar*** | ***Dr. Mahesh T R*** | ***Dr. Geetha G*** |
| ***Assistant Professor***,  Dept. of CSE,  School of Computer Science and Engineering  JAIN (Deemed-to-be University)  Date: | ***Program Head,***  Dept. of CSE,  School of Computer Science and Engineering  JAIN(Deemed-to-be University)  Date: | ***Director***,  School of Computer Science and Engineering  JAIN (Deemed-to-be University)  Date: |

Name of the Examiner Signature of Examiner

1.

2.

# DECLARATION

We, **YASH KUMAR JAIN (22BTRCN327), VARSHITHA-Y(22BTRCN313),ANKIT-CHOUDHARY(22BTRCN025),AFROZ ALAM(22BTRSN062),LOKNATH ROY(22BTLCN004)** are student’s of fourth semester B.Tech in **Computer Science & Engineering**, JAIN(Deemed-to-be University), hereby declare that the project centric learning work titled titled **Facial Recognition Attendance System** has been carried out by us and submitted in partial fulfilment for the project centric learning activity of degree in **Bachelor of Technology in Computer Science & Engineering** during the academic year **2023-2024**.

Signature

|  |  |
| --- | --- |
| Name1: YASH KUMAR JAIN M  USN: 22BTRCN327 |  |
| Name 2: VARSHITHA Y  USN: 22BTRCN313 |  |
| Name 3: ANKIT CHOUDHARY  USN: 22BTRCN025 |  |
| Name 4: AFROJ ALAM  USN: 22BTRSN062  Name 5: LOKNATH ROY  USN: 22BTLCN004 |  |

Place : Banglore

Date :

## ACKNOWLEDGEMENT

*It is a great pleasure for us to acknowledge the assistance and support of School of Computer Science and Engineering, Jain (Deemed-to-be University) for the progress of this project centric learning work.*

*In particular we would like to thank* ***Director Dr Geetha G****,* ***Director,*** *School of Computer Science Engineering and*  ***Dr Mahesh T R, Program Head,*** *Department of Computer Science and Engineering, JAIN (Deemed-to-be University) for their constant encouragement and expert advice.*

*We would like to thank our guide* ***Dr Gaurav Kumar****,* ***Assistant Professor,******Department of Computer Science & Engineering****, JAIN (Deemed-to-be University), for sparing his valuable time to extend help in every step of our project work, which paved the way for smooth progress of the project.*

*We would like to thank our Project Coordinators* ***Dr Gaurav Kumar*** *and all the staff members of Computer Science & Engineering for their support.*

*We would like to thank one and all who directly or indirectly helped us in the progress of our Project.*

*Signature of Students*

# ABSTRACT

Our proposed automated student attendance system utilizes advanced face recognition technology to verify student identities efficiently and accurately. The system begins by capturing video frames through a camera, with the Viola-Jones algorithm employed to detect and segment the face region of interest (ROI) from each frame. This initial face detection step is critical for isolating the face from the background. Once the face ROI is obtained, the preprocessing stage commences, involving several key steps to enhance image quality. Firstly, the images are scaled to a standard size to ensure uniformity. Noise reduction is achieved through median filtering, which helps eliminate any unwanted noise that might affect the accuracy of the recognition process. Contrast Limited Adaptive Histogram Equalization (CLAHE) is then applied to improve the contrast of the images, making facial features more distinguishable.The classification and recognition of facial images are based on the optimal combination of algorithms. K-Nearest Neighbors (KNN) is employed for the recognition process due to its simplicity and effectiveness in classifying data based on similarity. This method ensures that each face is accurately recognized by comparing it to the nearest neighbors in the feature space.Attendance is automatically marked and stored in an Excel file, streamlining the process and minimizing manual errors. The system also includes provisions for on-the-spot registration, allowing new students to be added seamlessly. Additionally, duplicate entry detection mechanisms are implemented to prevent multiple entries for the same student, ensuring the integrity of the attendance data. By leveraging the combination of the Viola-Jones algorithm for face detection and K-Nearest Neighbors for face recognition, our system significantly enhances the efficiency of attendance tracking. This automation reduces the time and effort required for manual attendance recording, allowing more valuable class time to be utilized for educational activities. The proposed system not only improves accuracy and reliability but also provides a convenient and effective solution for managing student attendance in educational institutions.

## TABLE OF CONTENTS

|  |  |
| --- | --- |
|  | Page No |
| List of Figures | vii |
| Nomenclature used | vii |
| Abstract | v |
| **Chapter 1** | **01** |
| 1. INTRODUCTION   Overview  Background and Motivation | 01  01  01 |
| **Chapter 2** | **02** |
| **2.** Literature Survey | 02 |
| **Chapter 3** | **03** |
| **3.** Objective and Methodology | 03 |
| 3.1 Objective | 03 |
| 3.2 Methodology  3.3 Algorithm Used | 03  03 |
| **Chapter 4** | **4** |
| **4.** System Design | 4 |
| 4.1 System Architecture | 4 |
| 4.2 Use Case diagram | 5 |
| 4.3 Data Flow diagram | 6 |
| 4.4 Sequence diagram | 6 |
| 4.5 Implementation | 19 |
| **Chapter 5** | **7** |
| **5.** Hardware and Software requirement | 7 |
| 5.1 Hardware requirement | 7 |
| 5.2 Software requirement  **Chapter 6**  **6.** Result  Result Screenshots | 7  8  8  9 |
| **Conclusion** | 10 |
| **References**  **Appendix – 1**  **Appendix - 2** | 11  12  13 |

**LIST OF FIGURES**

[Figure 1 Title of figure 1 7](https://d.docs.live.net/03ed21d7ba1161bc/Documents/Final-Year-Project/Thesis/thesis-v3.docx)

[Figure 2 Title of figure 2 ……………………………………………………………………15](https://d.docs.live.net/03ed21d7ba1161bc/Documents/Final-Year-Project/Thesis/thesis-v3.docx)

[Figure N Title of figure N 35](https://d.docs.live.net/03ed21d7ba1161bc/Documents/Final-Year-Project/Thesis/thesis-v3.docx)

vii

# Chapter 1

# INTRODUCTION

# Certainly, let's delve into the intricate domain of facial recognition—a remarkable human ability that intertwines visual perception with cognitive analysis. At its core, this process begins with the capture of light waves containing visual data by our retinas. These data streams are then meticulously parsed, dissecting shapes, sizes, contours, and textures, forming a rich tapestry of facial features. Subsequently, this information undergoes a sophisticated comparison against stored representations within our memory, culminating in the act of recognition. Yet, transcending human limitations, contemporary facial recognition systems harness the immense power of computation and memory. Their applications span a vast spectrum, from bolstering security measures to facilitating social interactions, encompassing realms as diverse as criminal investigations, social media tagging, and online authentication. The genesis of facial recognition systems can be traced back to the 1960s, where early endeavours sought to emulate the nuanced processes of human cognition.

**Overview**

The face attendance recognition system automates attendance tracking using facial recognition. It captures live facial images, matches them with a database of registered students, and marks attendance automatically. Utilizing advanced computer vision and machine learning algorithms, it ensures accuracy and efficiency in attendance management. The system's user-friendly interface allows for easy addition of new students and access to attendance reports. With robust security measures in place, it enhances privacy protection while streamlining administrative tasks.

**Background and Motivation**

**Background:**

Traditional attendance methods are often laborious and error-prone, prompting a shift towards automated systems leveraging facial recognition technology to streamline processes and enhance efficiency.

**Motivation:**

The background of this endeavor lies in the contemporary educational and corporate landscape's demand for streamlined attendance management. Traditional methods are beset by inefficiencies and prone to errors, necessitating a paradigm shift towards automated systems. Leveraging facial recognition technology, this initiative aims to overcome the limitations of manual attendance taking, enhancing accuracy, efficiency, and transparency. By embracing innovation, institutions can reallocate resources, reduce administrative burdens, and foster a more conducive learning or work environment.

# Chapter 2

# LITERATURE SURVEY

In the seminal work presented by Zhao et al. (2003), a comprehensive exploration of the challenges inherent in facial identification was undertaken. Among these challenges lies the intricate task of discerning between known and unknown facial images, a nuanced endeavor fraught with complexities. Moreover, the scholarly contribution by Pooja G.R et al. (2010) delved into the intricacies of the training process within face recognition systems designed for student attendance monitoring. It was discerned that this process is not merely sluggish but also exceedingly time-intensive, posing a substantial hurdle in the optimization of such systems. Furthermore, the scholarly discourse advanced by Priyanka Wagh et al. (2015) illuminated additional impediments encountered within the domain of face recognition-based student attendance systems. Notably, the variability induced by diverse lighting conditions and varied head poses emerged as formidable challenges, significantly undermining the performance metrics of such systems. In the comprehensive analysis conducted by Katara et al. (2017), a critical examination of various biometric authentication systems was undertaken, elucidating their respective drawbacks.

The intricate nature of iris data raises concerns regarding potential invasions of user privacy, thus engendering hesitancy among stakeholders. While voice recognition stands as an alternative, its efficacy pales in comparison to other modalities, exhibiting lower accuracy rates and susceptibility to ambient noise interference. In light of these deliberations, the face recognition system emerges as a compelling solution for integration within student attendance systems. Its inherent advantages, including the omnipresence of the human face and the balance between detail and privacy, position it as an optimal choice for ensuring both efficacy and user acceptance within educational contexts.

Scholarly contributions by S. Aanjanadevi et al. (2017) and Wei-Lun Chao (2007) have shed light on several pivotal factors exacerbating the challenges inherent in both face detection and recognition processes. These encompass a broad spectrum of complexities, including the influence of background environments, variations in illumination levels, diverse facial poses and expressions, occlusions obstructing facial features, rotational distortions, scaling disparities, as well as translational shifts within images.Such multifaceted considerations underscore the intricate nature of both face detection and recognition tasks, elucidating the myriad challenges that must be navigated to achieve robust and reliable performance within these domains.

Within the scholarly discourse, the research endeavours spearheaded by Akshara Jadhav et al. (2017) and P. Arun Mozhi Devan et al. (2017) converge on endorsing the Viola-Jones algorithm as the cornerstone for proficient face detection within student attendance systems.

Their empirical inquiries discrn the Viola-Jones algorithm as a standout choice amidst a panoply of methodologies, including face geometry-based analyses, Feature Invariant methodologies, and Machine Learning paradigms. Noteworthy is its exceptional swiftness and resilience, coupled with its propensity to yield superior detection rates, even in the face of variable lighting conditions. These findings find resonance in the findings of Rahul V. Patil and S. B. Bangar (2017), who further attest to the Viola-Jones algorithm's prowess in adapting to diverse lighting scenarios, underlining its adaptability and reliability in real-world deployment. Additionally, the research elucidated by Mrunmayee Shirodkar et al. (2015) underscores the algorithm's versatility, notably its adeptness in mitigating challenges stemming from illumination discrepancies, scaling intricacies, and rotational variations, fortifying its applicability across multifarious environments.Conversely, the fingerprint system, hailed for its effectiveness, grapples with inefficiency concerns. The verification process entails a significant time investment, necessitating users to queue up and undergo verification sequentially, thereby impeding the expediency of the authentication process. Meanwhile, the iris recognition system, characterized by its intricate detail and robust authentication capabilities, confronts privacy apprehensions. In a parallel vein, the scholarly exposition put forth by Naveed Khan Balcoh (2012) casts the Viola-Jones algorithm in a favourable light, positioning it as the paragon of efficiency amidst a landscape populated by various contenders, including AdaBoost, Float Boost, Neural Networks, S-AdaBoost, Support Vector Machines (SVM), and the Bayes classifier. This acknowledgment underscores the algorithm's eminence and enduring relevance, solidifying its stature as the quintessential choice for discerning practitioners seeking optimal performance in face detection endeavours.

# Chapter 3

# OBJECTIVE AND METHODOLOGY

* 1. **Objective**

The primary objective of this project is to develop and implement a Facial Recognition Attendance System. This system aims to automate the attendance tracking process, leveraging facial recognition technology to accurately identify and record student or employee attendance. The goal is to eliminate the need for manual attendance taking, thereby saving time, reducing errors, and improving overall efficiency in educational and corporate settings. Additionally, the system seeks to provide a user-friendly interface and ensure data security and privacy to meet the diverse needs and requirements of users.

## Methodology

## The methodology employed in the facial attendance recognition system encompasses a multifaceted approach. Initially, data acquisition entails the collection of facial images through sophisticated imaging apparatuses, followed by an intricate preprocessing phase aimed at refining image quality and standardization for subsequent analysis. Subsequently, feature extraction procedures, such as Principal Component Analysis (PCA) or Local Binary Patterns (LBP), are employed to distill salient facial characteristics into numerical representations, facilitating discriminative modeling. Leveraging machine learning paradigms like the K-Nearest Neighbors (KNN) algorithm, the system undergoes rigorous training on labeled datasets to discern intricate patterns and correlations between facial attributes and corresponding identities. Upon model training, face recognition entails a meticulous process of comparing input images against learned features, culminating in precise identity predictions. Attendance recording involves the seamless integration of recognized identities with temporal stamps, ensuring accurate attendance tracking. Continuous evaluation and refinement further optimize system efficacy, eventually enabling seamless deployment and integration within educational or corporate environments.

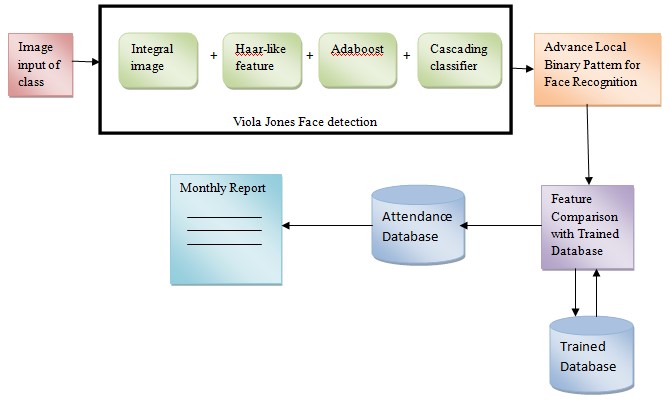
* 1. **Algorithm Used:**

The implemented algorithm is the K-Nearest Neighbors (KNN) algorithm, a robust instance-based learning method within the domain of supervised machine learning. This approach operates by storing all available training data points and classifying new instances based on a similarity measure, often calculated through Euclidean distance or other metrics within the feature space. In the context of facial recognition and attendance tracking, KNN undertakes the task of categorizing facial images into predefined classes, representing registered students. By leveraging the similarity between facial features, the algorithm effectively discerns and matches faces to their respective identities, facilitating automated attendance marking based on recognized individuals.

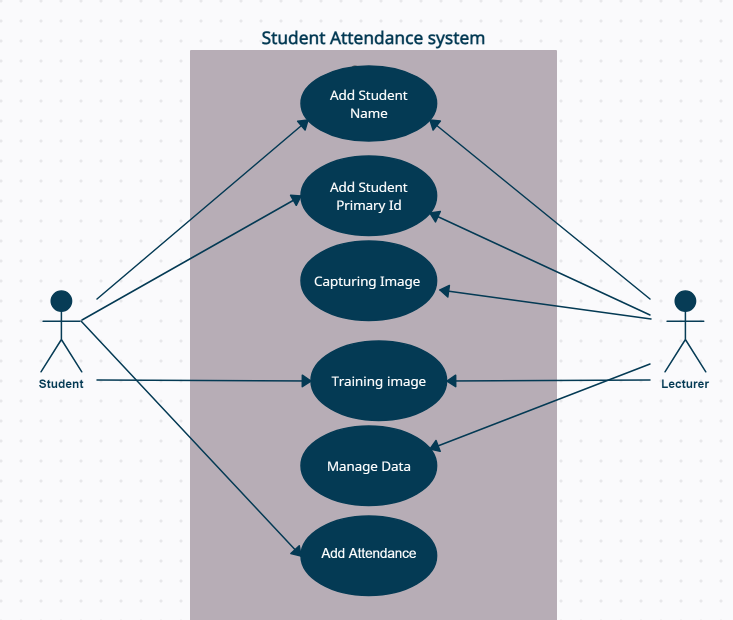
# Chapter 4

# SYSTEM DESIGN

**4.1 Proposed System Architecture**

This diagram provides an overview of the system's architecture, illustrating the components, modules, and their interactions. It serves to visually represent the structural design of the system, offering insights into how different elements work together to achieve the system's objectives. 

**4.2 Use case Diagram**

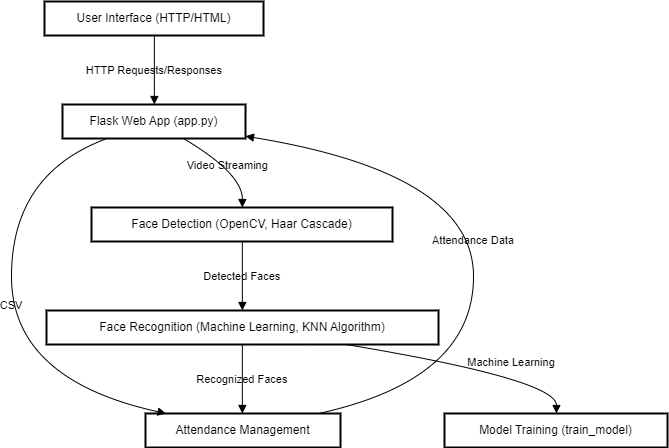
****



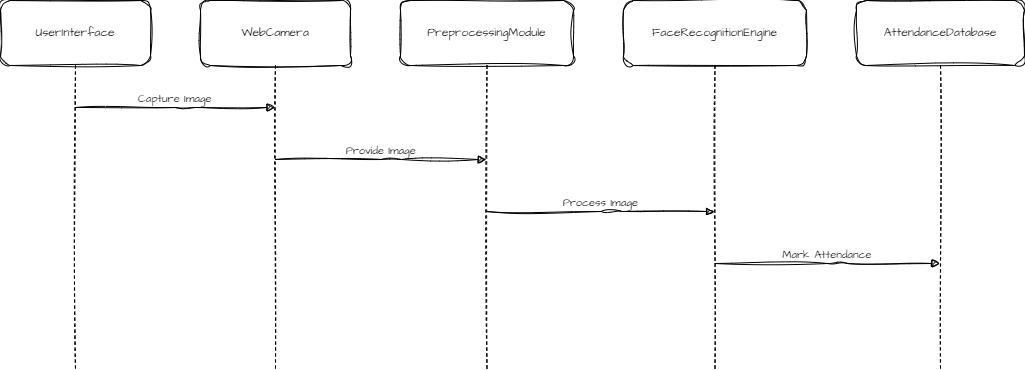
The use case diagram depicts the various interactions between users and the system, showcasing the different functionalities or use cases of the system. It provides a visual representation of how users interact with the system to accomplish specific tasks, such as marking attendance, accessing reports, or managing user accounts.



**4.3 Data flow Diagram**

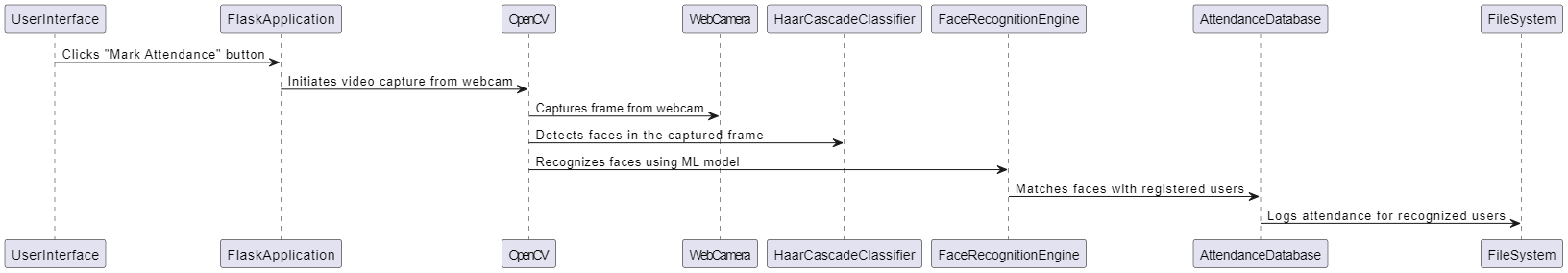
****

This diagram illustrates the flow of data within the system, showcasing how information moves through different processes and components. It provides insights into the input, processing, and output of data, offering a visual representation of how data is handled and utilized within the system.

**4.4 Sequence Diagram**

The sequence diagram presents a dynamic view of the system, depicting the sequence of interactions between different components or objects to accomplish specific tasks. It showcases the chronological order of messages or actions exchanged between system elements, offering a visual representation of system behavior.

**4.5 IMPLEMENTATION**

****

While not a diagram, the implementation section likely provides details about how the system is implemented, including the technologies, programming languages, frameworks, and tools used in the development and deployment of the Facial Recognition Attendance System.

# Chapter 5

### HARDWARE AND SOFTWARE REQUIREMENTS

**Hardware Requirements:**

Web Camera: A webcam or any camera device capable of capturing live video feed is essential for the face recognition system to capture images of individuals.

Computer: A computer system with sufficient processing power and memory to run the face recognition algorithm and related software components smoothly.

Storage Device: Sufficient storage space is needed to store the face images of registered users, as well as any logs or databases generated during the attendance marking process.

Internet Connectivity: While not mandatory, internet connectivity can be beneficial for accessing updates, downloading libraries, or integrating cloud-based services.

**Software Requirements:**

OpenCV: OpenCV (Open Source Computer Vision Library) is a critical software library for image processing tasks, including face detection and recognition.

Flask: Flask is a lightweight web framework used to create the user interface and handle HTTP requests for initiating attendance marking and user management tasks.

Python: Python serves as the primary programming language for developing the face recognition system due to its simplicity, extensive libraries, and support for machine learning.

Haar Cascade Classifier: A pre-trained Haar Cascade classifier is utilized for face detection within images or video streams, enabling the system to identify faces accurately.

Machine Learning Libraries: Libraries such as scikit-learn or TensorFlow are employed for training machine learning models to recognize faces based on extracted features.

Operating System: The system can run on various operating systems, including Windows, macOS, or Linux distributions, depending on the developers' preferences and compatibility requirements.

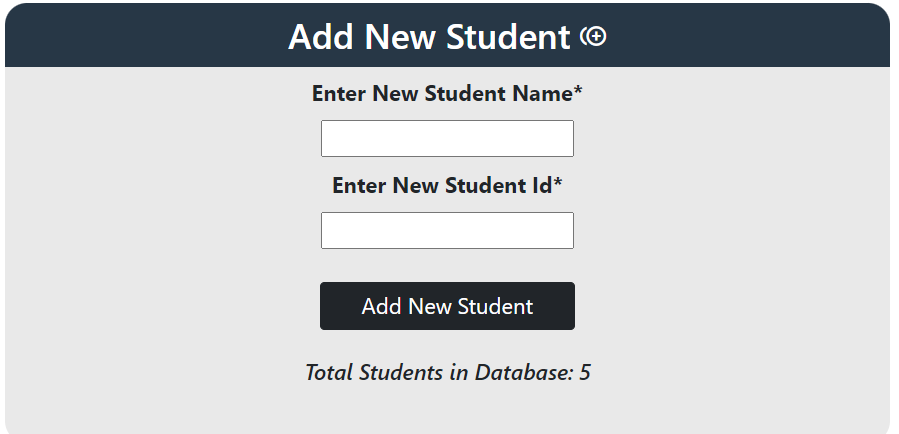
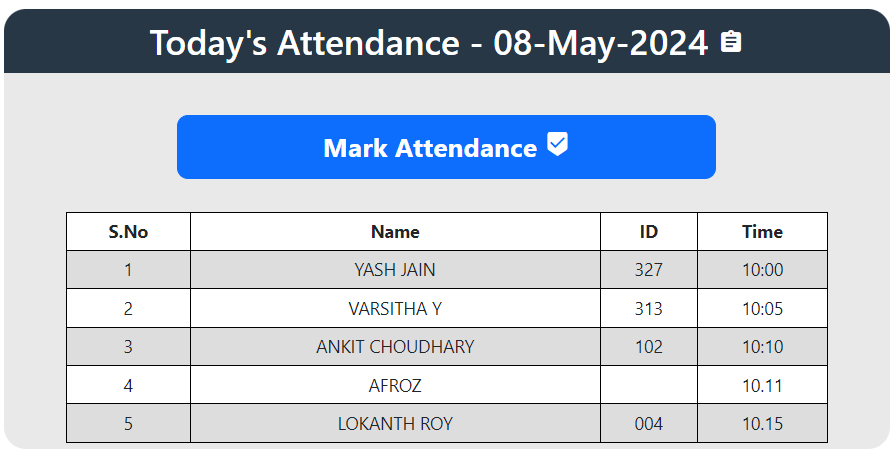
Text Editor or IDE: A text editor or integrated development environment (IDE) like Visual Studio Code, PyCharm, or Sublime Text is used for writing, editing, and debugging the project code.

# Chapter 6

# Result

# Gathering Identity Signatures for Dataset Formation

# It include screenshots to visually represent the performance or output of the system. These screenshots showcase the system's ability to accurately identify individuals based on facial features and record attendance. The images included in this chapter are expected to visually demonstrate the system's functionality, potentially showing the process of gathering identity signatures for dataset formation, as well as the system's ability to recognize and record attendance based on facial recognition. These images are likely to provide a visual representation of the system's performance and the practical application of facial recognition technology in automating attendance tracking.



Attendance Successfully Logged: Students' Presence Noted and Visualized

The provided screenshots depict the project interface, comprising the functionality to input student details such as name and ID, along with a feature to display the count of successfully added students. Upon error-free data entry, the system seamlessly updates and showcases the total count of enrolled students. Furthermore, the interface hosts a "**Mark Attendance**" button, leveraging advanced facial recognition technology to record attendance. Upon successful marking, pertinent details including the student's name, ID, and timestamp of attendance are promptly displayed. This sophisticated interface not only facilitates the seamless addition of student information but also employs cutting-edge facial recognition algorithms for efficient attendance tracking. In the event of a successful attendance mark, it offers a comprehensive display of relevant student details, enhancing administrative efficiency and accuracy.

**CONCLUSION**

In conclusion, the face recognition system represents a remarkable fusion of advanced technology and practical application, offering a sophisticated solution to various real-world challenges. With its ability to seamlessly identify individuals based on facial features, this system transcends conventional methods, revolutionizing security protocols, law enforcement practices, and everyday conveniences. However, amid its myriad benefits, the system is not devoid of intricacies and considerations. The intricate algorithms and meticulous preprocessing techniques underscore the complexity involved in accurately capturing, processing, and interpreting facial data. Furthermore, the reliance on robust datasets and continuous model refinement underscores the dynamic nature of this technology, demanding ongoing adaptation and optimization. Yet, despite these complexities, the face recognition system stands as a testament to human ingenuity, offering unparalleled efficiency, accuracy, and convenience in diverse domains. As we navigate the evolving landscape of technological innovation, the face recognition system serves as a beacon of progress, continually pushing the boundaries of what is possible and reshaping our interactions with the world around us.

# REFERENCES

1. Naveed Khan Balcoh. (2012). Algorithm for Efficient Attendance Management: Face Recognition based approach. International Journal of Computer Science Issues, V9 (4), No 1.
2. Rahul V. Patil and S. B. Bangar. (2017). Video Surveillance Based Attendance system. IJARCCE, 6(3), pp.708-713.
3. P. Arun Mozhi Devan et al., (2017). Smart Attendance System Using Face Recognition. Advances in Natural and Applied Sciences. 11(7), Pages: 139 144
4. Aanjanadevi, S. et al. “Face Attribute Convolutional Neural Network System for Data Security with Improved Crypto Biometrics.” Comput. Syst. Sci. Eng. 45 (2023): 2351-2362.
5. Zhao, W., Chellappa, R., Phillips, P. and Rosenfeld, A. (2003). Face recognition. ACM Computing Surveys, 35(4), pp.399 458.
6. Paul, Happy & Garg, Pooja. (2014). FACTOR STRUCTURE OF THE RESILIENCE SCALE-14: INSIGHTS FROM AN INDIAN SAMPLE 71 Factor Structure of the Resilience Scale-14: Insights from an Indian Sample. South Asian Journal of Management. Vol.21,. pp.70-86..
7. Wagh, Priyanka & Thakare, Roshani & Chaudhari, Jagruti & Patil, Shweta. (2015). Attendance system based on face recognition using eigen face and PCA algorithms. 303-308. 10.1109/ICGCIoT.2015.7380478.
8. Prof. Arun Katara1, Mr. Sudesh V. Kolhe2, Mr. Amar P. Zilpe3, Mr. Nikhil D. Bhele4, Mr. Chetan J. Bele5. Attendance System Using Face Recognition and Class Monitoring System. ISSN: 2321-8169, 273 -276. International Journal on Recent and Innovation Trends in Computing and Communication. Vol.5
9. Wei-Lun Chao, Jian-Jiun Ding, Jun-Zuo Liu,
10. Facial expression recognition based on improved local binary pattern and class-regularized locality preserving projection,Signal-Processing,Volume-117,2015,Pages-1-10,ISSN-0165-1684, <https://doi.org/10.1016/j.sigpro.2015.04.007>.(https://www.sciencedirect.com/science/article/pii/S0165168415001425)

**APPENDIX – 1**

**PHOTOGRAPHS**

**(With Student and Guide Information)**

|  |  |
| --- | --- |
| Team Member | Photograph |
| YASH KUMAR JAIN M  22BTRCN327  Dept. of Computer Science & Engineering |  |
| VARSHITHA Y  22BTRCN313  Dept. of Computer Science & Engineering |  |
| ANKIT CHOUDHARY  22BTRCN025  Dept. of Computer Science & Engineering |  |
| AFROJ ALAM  22BTRSN062  Dept. of Computer Science & Engineering |  |
| LOKNATH ROY  22BTLCN004  Dept. of Computer Science & Engineering |  |
| GAURAV KUMAR  Assistant Professor  Dept. of Computer Science & Engineering |  |

**APPENDIX - 2**

**SOURCE CODE**

App.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

app = Flask(\_\_name\_\_)

nimgs = 10

datetoday = date.today().strftime("%m\_%d\_%y")

datetoday2 = date.today().strftime("%d-%B-%Y")

face\_detector = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')

if not os.path.isdir('Attendance'):

    os.makedirs('Attendance')

if not os.path.isdir('static'):

    os.makedirs('static')

if not os.path.isdir('static/faces'):

    os.makedirs('static/faces')

if f'Attendance-{datetoday}.csv' not in os.listdir('Attendance'):

    with open(f'Attendance/Attendance-{datetoday}.csv', 'w') as f:

        f.write('Name,Roll,Time')

def totalreg():

    return len(os.listdir('static/faces'))

def extract\_faces(img):

    try:

        gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

        face\_points = face\_detector.detectMultiScale(gray, 1.2, 5, minSize=(20, 20))

        return face\_points

    except:

        return []

def identify\_face(facearray):

    model = joblib.load('static/face\_recognition\_model.pkl')

    return model.predict(facearray)

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, (50, 50))

            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')

def extract\_attendance():

    df = pd.read\_csv(f'Attendance/Attendance-{datetoday}.csv')

    names = df['Name']

    rolls = df['Roll']

    times = df['Time']

    l = len(df)

    return names, rolls, times, l

def add\_attendance(name):

    username = name.split('\_')[0]

    userid = name.split('\_')[1]

    current\_time = datetime.now().strftime("%H:%M:%S")

    df = pd.read\_csv(f'Attendance/Attendance-{datetoday}.csv')

    if int(userid) not in list(df['Roll']):

        with open(f'Attendance/Attendance-{datetoday}.csv', 'a') as f:

            f.write(f'\n{username},{userid},{current\_time}')

def getallusers():

    userlist = os.listdir('static/faces')

    names = []

    rolls = []

    l = len(userlist)

    for i in userlist:

        name, roll = i.split('\_')

        names.append(name)

        rolls.append(roll)

    return userlist, names, rolls, l

def deletefolder(duser):

    pics = os.listdir(duser)

    for i in pics:

        os.remove(duser+'/'+i)

    os.rmdir(duser)

# Our main page

@app.route('/')

def home():

    names, rolls, times, l = extract\_attendance()

    return render\_template('home.html', names=names, rolls=rolls, times=times, l=l, totalreg=totalreg(), datetoday2=datetoday2)

@app.route('/listusers')

def listusers():

    userlist, names, rolls, l = getallusers()

    return render\_template('listusers.html', userlist=userlist, names=names, rolls=rolls, l=l, totalreg=totalreg(), datetoday2=datetoday2)

## Delete functionality

@app.route('/deleteuser', methods=['GET'])

def deleteuser():

    duser = request.args.get('user')

    deletefolder('static/faces/'+duser)

    ## if all the face are deleted, delete the trained file...

    if os.listdir('static/faces/')==[]:

        os.remove('static/face\_recognition\_model.pkl')

    try:

        train\_model()

    except:

        pass

    userlist, names, rolls, l = getallusers()

    return render\_template('listusers.html', userlist=userlist, names=names, rolls=rolls, l=l, totalreg=totalreg(), datetoday2=datetoday2)

# Our main Face Recognition functionality.

# This function will run when we click on Take Attendance Button.

@app.route('/start', methods=['GET'])

def start():

    names, rolls, times, l = extract\_attendance()

    if 'face\_recognition\_model.pkl' not in os.listdir('static'):

        return render\_template('home.html', names=names, rolls=rolls, times=times, l=l, totalreg=totalreg(), datetoday2=datetoday2, mess='There is no trained model in the static folder. Please add a new face to continue.')

    ret = True

    cap = cv2.VideoCapture(0)

    while ret:

        ret, frame = cap.read()

        if len(extract\_faces(frame)) > 0:

            (x, y, w, h) = extract\_faces(frame)[0]

            cv2.rectangle(frame, (x, y), (x+w, y+h), (86, 32, 251), 1)

            cv2.rectangle(frame, (x, y), (x+w, y-40), (86, 32, 251), -1)

            face = cv2.resize(frame[y:y+h, x:x+w], (50, 50))

            identified\_person = identify\_face(face.reshape(1, -1))[0]

            add\_attendance(identified\_person)

            cv2.putText(frame, f'{identified\_person}', (x+5, y-5),

                        cv2.FONT\_HERSHEY\_SIMPLEX, 1, (255, 255, 255), 2)

        cv2.imshow('Attendance', frame)

        if cv2.waitKey(1) == 27:

            break

    cap.release()

    cv2.destroyAllWindows()

    names, rolls, times, l = extract\_attendance()

    return render\_template('home.html', names=names, rolls=rolls, times=times, l=l, totalreg=totalreg(), datetoday2=datetoday2)

@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)

    i, j = 0, 0

    cap = cv2.VideoCapture(0)

    while 1:

        \_, frame = cap.read()

        faces = extract\_faces(frame)

        for (x, y, w, h) in faces:

            cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 20), 2)

            cv2.putText(frame, f'Images Captured: {i}/{nimgs}', (30, 30),

                        cv2.FONT\_HERSHEY\_SIMPLEX, 1, (255, 0, 20), 2, cv2.LINE\_AA)

            if j % 5 == 0:

                name = newusername+'\_'+str(i)+'.jpg'

                cv2.imwrite(userimagefolder+'/'+name, frame[y:y+h, x:x+w])

                i += 1

            j += 1

        if j == nimgs\*5:

            break

        cv2.imshow('Adding new User', frame)

        if cv2.waitKey(1) == 27:

            break

    cap.release()

    cv2.destroyAllWindows()

    print('Training Model')

    train\_model()

    names, rolls, times, l = extract\_attendance()

    return render\_template('home.html', names=names, rolls=rolls, times=times, l=l, totalreg=totalreg(), datetoday2=datetoday2)

if \_\_name\_\_ == '\_\_main\_\_': # main function

    app.run(debug=True)

Home.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://cutewallpaper.org/21/1920-x-1080-gif/1920x1080-Wallpapercartoon-Wallpapers-Driverlayer-Search-.gif');

        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>Face Recognition Based Attendance System</title>

</head><body>

<div class='mt-3 text-center'>

        <h1 style="width: auto;margin: auto;color: white;padding: 11px;font-size: 44px;">Face Recognition Based

            Attendance System</h1>

    </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);margin:0px 10px 10px 10px;min-height: 400px;">

            <h2 style="border-radius: 20px 20px 0px 0px;background-color: #273746 ;color: white;padding: 10px;">Today's

                Attendance - {{ datetoday2 }} <i class="material-icons">assignment</i></h2>

            <a style="text-decoration: none;max-width: 300px;" href="/start">

                <button

                    style="font-size: 24px;font-weight: bold;border-radius: 10px;width:490px;padding: 10px;margin-top: 30px;margin-bottom: 30px;"

                    type='submit' class='btn btn-primary'>Mark Attendance <i

                        class="material-icons">beenhere</i></button>

            </a>

            <table style="background-color: white;">

                <tr>

                    <td><b>S.No</b></td>

                    <td><b>Name</b></td>

                    <td><b>ID</b></td>

                    <td><b>Time</b></td>

                </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> </div>

<div class="col"

            style="border-radius: 20px;padding: 0px;background-color:rgb(211,211,211,0.5);margin:0px 10px 10px 10px;height: 400px;">

            <form action='/add' method="POST" enctype="multipart/form-data">

                <h2 style="border-radius: 20px 20px 0px 0px;background-color: #273746 ;color: white;padding: 10px;">Add

                    New Student <i class="material-icons">control\_point\_duplicate</i></h2>

                <label style="font-size: 20px;"><b>Enter New Student 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 Student 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 Student

                </button>

                <br>

                <h5 style="padding: 25px;"><i>Total Students in Database: {{totalreg}}</i></h5>

            </form>

        </div>

    </div>

    <script>

        // Function to get today's date and update the HTML element

        function updateDate() {

            var today = new Date();

            var options = { weekday: 'long', year: 'numeric', month: 'long', day: 'numeric' };

            var formattedDate = today.toLocaleDateString('en-US', options);

            document.getElementById("date").innerHTML = "Today's Date: " + formattedDate;

        }

        // Call the function to update the date when the page loads

        updateDate();

        // Update the date every day at midnight

        setInterval(updateDate, 86400000); // 86400000 milliseconds = 1 day

    </script>

</body>

</html>