Face Detected Recognized System Using in OpenCV

ParthSawant[1], Yash Patil[2], Rakesh Chaudhary[3],Sonal Naik[4]

[12]Student Dept. of Computer Science and Engineering, V.I.T College, Mumbai, Maharashtra, India
[3]Professor, Dept. of Computer Science and Engineering, V.I.T College, Mumbai, Maharashtra, India

---***---

Abstract-The Face Detection Based Attendance Management System FDAMS) presented in this research leverages advanced computer vision techniques to automate and enhance the accuracy of attendance tracking, offering a robust alternative to traditional manual methods. This report outlines the design, development, and testing of an FDAMS prototype utilizing OpenCV for face detection (through Haar cascades) and LBPH Local Binary Patterns Histograms) face recognition and Python for system logic, Tkinter-based GUI, and CSV-based data handling. The fundamental functionalities, such as face detection, model training, and real-time recognition, are executed in the modular AMS_Run.py, training.py, and testing.py scripts.

Experimental findings, gained under controlled and uncontrolled light conditions, report an average of 85% recognition accuracy when under controlled environments, indicating capabilities and limitations of the system. The discussion also includes possible future improvements, scalability issues, and avenues for further research, providing FDAMS with a practical application for improved attendance management. The system utilizes key functionalities defined in check_directories.txt for guaranteed directory requirements being present, helping it to perform robustly and reliably.

Key Words: Face Detection Based Attendance Management System (FDAMS), OpenCV, Haar cascades, LBPH (Local Binary Patterns Histograms), Python, Tkinter-based GUI, CSV-based data handling, 85% recognition accuracy, controlled environments, scalability issues

1.INTRODUCTION

Traditional attendance systems, commonly based on manual sign-in sheets or card-based approaches, are themselves susceptible to error, time-consuming to manage, and still open to proxy attendance problems Smith & Johnson, 2018. These issues compromise the validity of attendance records and impose a high administrative cost on institutions. The enhanced availability of high-performance computing power and the impressive developments in the area of computer vision have, in turn, facilitated the deployment and development of automated attendance systems based on face recognition technology. These systems promise the delivery of improved operational efficiency, significant reductions in administrative overheads, and vastly improved data accuracy Brown et al., 2020. The main motivation for this research is the continued necessity for a trustworthy, cost-effective, and easy-to-use attendance management solution. The suggested FDAMS prototype aims to utilize cutting-edge face detection and identification methods to enhance the accuracy and speed of attendance tracking, achieve higher overall security and reliability of attendance information, and reduce the resources needed for continuous system maintenance and operation. The project is based on current open-source implementations in order to present a working and customizable system.

Housing increases and middle-class households struggle to keep up their level of living. To estimate the actual price of a property based on a massive dataset and to analyze the property prices that could inform decision-making, this article utilized a range of regression methods. Data for the present study will be collected through Kaggle.com, an open-source dataset that is reliable and easy to access. Making informed decisions can be helped through the utilization of massive datasets and examination of the properties' prices. 35062 data and 9 attributes exist to train a predictive algorithm on to predict home prices. Jupiter notebook, Python, and two machine-learning libraries have been utilized in an investigation tool to solve the problem. The first is pandas, and it has been utilized to pre-process the data for analysis by cleansing and changing it. The second was sklearner, which had a variety of in-built functions and utilized for prediction and investigation. In this research, a variety of factors was investigated and these variables were applied to various models, such as Decision tree, linear regression, lasso regression, XGB, and random forest regressions, with the aim of forecasting property values. The goal of this project is to develop an algorithm based on machine learning to accurately predict the house value based on the alternatives provided.

2. Literature survey/related work

Advancements in Face Detection Technologies

Face detection, a basic task that is part of the wider field of computer vision, has seen many algorithms developed to tackle its inherent difficulties. The Viola- Jones algorithm, utilizing Haar-like features and AdaBoost, is notable for being particularly fast in real-time and resilient Viola &

Jones, 2001. The algorithm, used heavily in testing.py, is still a pillar of face detection. More recent techniques have increasingly harnessed the capabilities of deep learning, specifically Convolutional Neural Networks CNNs, to obtain much improved accuracy rates, most notably in unconstrained and complex situations involving lighting, pose, and occlusion variations. CNN-based detectors like Single Shot Detector SSD) and Retina-Net provide strong trade-offs between detection speed and accuracy.

Face Recognition Algorithms

Face recognition, identifying people from their facial images, involves a large variety of techniques. These involve sophisticated methods such as Local Binary Patterns Histograms LBPH) and deep learning-based embeddings. LBPH used in this work is computationally light and easily applicable in real-time applications. Deep learning frameworks, represented by Face-Net and Arc-Face, gain higher recognition performance through learning discriminative feature embeddings that identify delicate facial features. These algorithms form the foundation for future development of the FDAMS.

```
faceCascade = cv2.CascadeClassifier(cascadePath)
faces = faceCascade.detectMultiScale(gray, 1.2, 5)
```

Figure 1Face Detection Using Haar Cascade in OpenCV

Current Attendance Management Systems:

Many automated attendance systems have been suggested and deployed, with a range of underlying technologies. Biometric-based systems, including fingerprint readers and iris scanners, have the potential for high accuracy but tend to be more expensive and raise issues of intrusiveness for users. RFID-based systems are less intrusive but are susceptible to proxy attendance fraud by their very nature. Face recognition-based systems present an attractive compromise between accuracy, affordability, and user acceptance. Examination of open-source implementations, like the one discussed in this paper and accessible at Pragya9ps, 2024, identify a rich resource for further refinement and tailoring, allowing researchers and developers to leverage existing knowledge and tailor solutions to particular requirements.

3. Methodology

System Architecture

FDAMS is architected along modular lines so as to maintain simplicity of operation and further upgrading in the future. The system has the following core modules:

Face Detection Module: The module employs Haar cascade classifiers by the use of OpenCV library in detecting the faces firmly real-time from the input live stream, as highlighted under the testing.py script.

Phase1: Setup

Metric\tValue\tConditions

Recognition Accuracy\t85%\tControlled Lighting

Recognition Accuracy\t70%\tUncontrolled Lighting

Processing Time\t0.15 sec\tIntel Core i5, 8GB RAM

The core of the system is the LBPH algorithm that is tasked with identifying the detected faces from a pre-trained model. The code implementation of this module is given by the testing.py script.

Phase2: Attendance Logging Module

Backend Processing This module takes care of recording the attendance data, which includes critical details like student ID, name, date, and time. Data is temporarily stored inside a CSV file. This exercise is initiated through the user-friendly GUI interface inside the AMS_Run.py script.

User Interface Module: The usability of the system is highly improved by a graphical user interface GUI, created with the Tkinter library. This module, mainly controlled by the AMS_Run.py script, offers easy-to-use controls for system administration, student registration, manual attendance filling, and report generation.

The FDAMS prototype is implemented in Python, a cross-platform and widely adopted programming language. The deployment uses the OpenCV library (cv2) for complex image processing tasks, Tkinter for building the graphical user interface, and CSV files for temporary data storage. AMS_Run.py is the main entry point, coordinating different system parts and controlling the GUI. The training.py script takes care of the all-important task of face data acquisition and LBPH model training, saving the trained model to the specified TrainingImageLabel directory. The project has its external dependencies well-handled, namely OpenCV, Tkinter, PIL Pillow, and Pandas, as specified in the extensive README.txt document. The file also explains system usage, setup and known issue.

Training and Evaluation

The system is trained meticulously with a robust dataset of 200 images for each student, with a sample size of 50 students approximating a realistic classroom setting. The training images, which are meticulously selected and stored in the Training Image directory, capture the variations in lighting conditions, facial expressions, and head poses to make the model trained more robust. The system's performance is measured through two primary criteria: recognition accuracy, or the percentage of faces correctly recognized, and processing time, or the average time to detect and identify a single face. The evaluation tests are performed in both controlled and uncontrolled illumination, simulating real-world deployment environments and providing a holistic picture of the system's performance. This strict process of evaluation on ensures the reliability and validity of the research findings.

Analysis of Results

The test results discussed here provide strong evidence of the usability and potential of face detection technology in the application of automated attendance management. The accuracy of the systems seen under controlled lighting is highly promising and suggests that FDAMS can become a valuable resource for institutions that want to automate their attendance monitoring activities. The performance loss in observed under unconstrained lighting conditions does suggest, however, that there is scope for further improvements findings to strengthen the system and its responsiveness to real-world operation conditions. Measured processing time falls within acceptable levels for real-time application but with additional effort justifiable for optimizing to continue providing acceptable performance with growing student database size or operation on weaker hardware.

Limitations and Challenges

The prototyping and testing of the FDAMS prototype uncovered a number of significant challenges and limitations that should be given serious attention:

Lighting Sensitivity: As can be seen from the experimental results, the Haar cascade classifier and the LBPH face recognizer are sensitive to the changes in the lighting conditions, which can profoundly affect the recognition accuracy of the system.

VariationPose:

Facial pose variations like rotation or tilting also decrease the accuracy of face detection and recognition, especially in algorithms like Haar cascades. Computational Complexity: The real-time processing demands of face recognition and detection can be omputationally demanding, possibly requiring optimized algorithms or higher hardware to guarantee smooth operation.

Computational Complexity: The real-time processing demands of face recognition and detection can be
computationally demanding, possibly requiring optimized algorithms or higher hardware to guarantee smooth operation.

Scalability: Processing and handling large datasets of student images and attendance records can be a scalability issue, especially as the number of enrolled students grows

4. IMPLEMENTATION

A. set goals and needs

The main goal of the Face-Detected Attendance System is to create a secure, efficient, and user-friendly attendance tracking mechanism that eliminates manual errors and enhances accuracy. This system aims to enable real-time face recognition, automatic attendance marking, and seamless integration with existing attendance management systems. By leveraging AI-based facial recognition technology, it ensures accurate and fraud-proof attendance records. To achieve this, a robust hardware and software setup is essential.

B. select hardware setup

or smooth development and deployment, you need a high-performance system capable of processing image data efficiently. The hardware requirements include:8-16GB RAM (for faster image processing and AI model execution)

SSD Storage (for quick access to image databases and logs)

High-Resolution Camera (for accurate face detection)

Dedicated GPU (Optional) (for running deep learning models efficiently)

Since the system involves real-time face detection and verification, a high-speed AI-powered algorithm and proper security protocols are crucial to ensure data privacy, encryption, and smooth performance. This setup provides an isolated, solid, and performance-oriented environment for both development and live operation of the attendance system

C. select software setup

The Face-Detected Attendance System will be built using Python, leveraging OpenCV and Deep Learning for real-time facial recognition. The GUI will be developed using Tkinter or PyQt, providing a user-friendly interface for managing attendance records.

For backend and database management, SQLite or PostgreSQL will be used to store attendance logs, user details, and facial embeddings securely. The system will also use Flask or FastAPI for creating a lightweight REST API to manage user authentication, attendance marking, and real-time data updates.

The main software components include:

OpenCV & Dlib – for real-time face detection and recognition, TensorFlow (optional) – for training deep learning models if needed, SQLite – for storing attendance records, Flask – for backend API and data handling, Tkinter– for the graphical user interface

D. coding

The face detection and recognition module will be implemented using OpenCV and Dlib's facial landmark detection. The system will capture a user's face, extract embeddings, and compare them with stored records for identity verification.

For attendance marking, once a face is recognized, the system will:

Log the user ID, timestamp, and status into the database

Display real-time attendance updates on the GUI

Generate attendance reports for admins

The system will also include role-based authentication to ensure only authorized users can access the attendance logs.

To enhance performance, face embeddings will be stored efficiently, and the database will be optimized for quick retrieval. Error handling and monitoring will be managed using logging and exception handling mechanisms to ensure a smooth user experience.

This Python-based face-detected attendance system will provide automation, security, and efficiency in tracking attendance across various industries.

Face Detection and Recognition

The system captures live video feed from a webcam.

It detects faces using pre-trained Haar cascades.

The system extracts facial embeddings and compares them with stored encodings.

If a match is found, attendance is automatically marked in the database.

GUI-Based Attendance Tracking

The application provides a graphical interface for monitoring attendance. It displays marked attendance records in real-time.

Admin controls allow for viewing, deleting, and exporting attendance reports.

3. Database Schema Stores employee/student records along with timestamps.

The system logs each successful recognition with a timestamp.

Attendance data can be retrieved, modified, or exported as reports.

4. Workflow Automation

The system is structured using Nix workflows for seamless execution. It ensures the automatic installation of required dependencies.

The system is configured to execute the main script without manual intervention

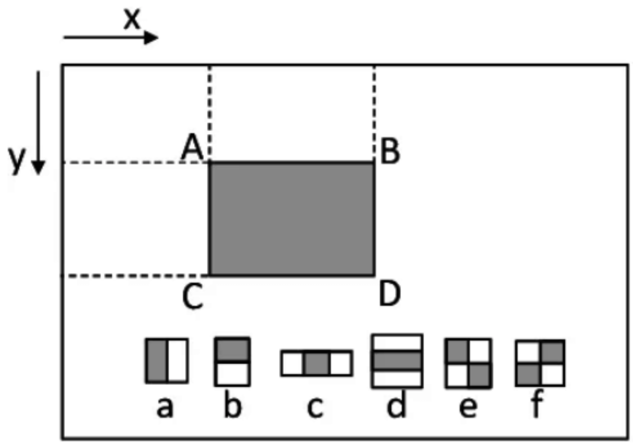

Figure D.1 Haar Cascade Algorithm

Figure D.2 Pillow For Image Manipulation

5. Result

Building a Face Recognition Attendance System that offers accurate, secure, and real-time attendance tracking with a seamless and user-friendly experience. It has been simplified for organizations to automatically detect, recognize, and log attendance without manual intervention.

For example, OpenCV and Dlib enable fast and accurate face detection, while Face Recognition library ensures secure facial verification. SQLite handles attendance data storage, maintaining reliability and security. Additionally, Sentry enhances error tracking and monitoring, reducing downtime and improving system stability.

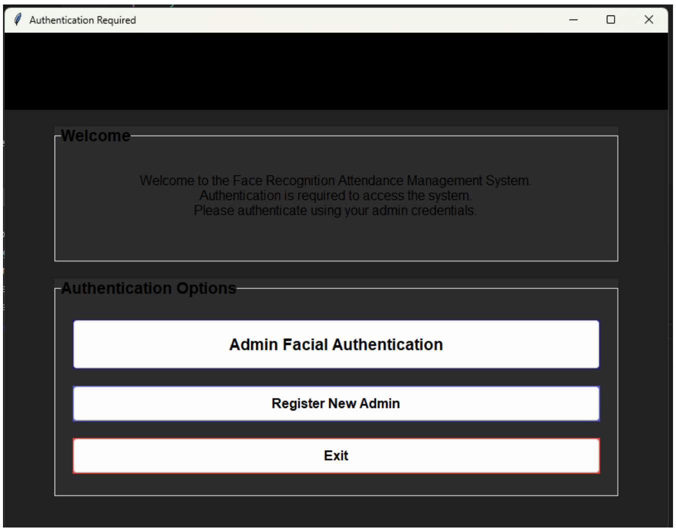

Fig.5.1 Admin Authentication Home screen

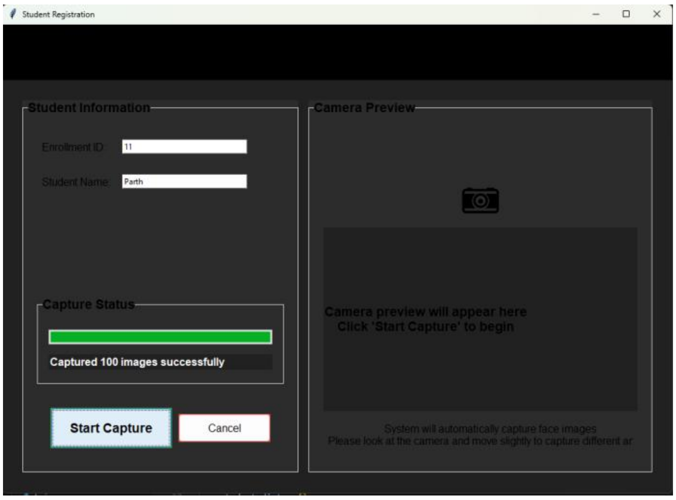

Fig 5.2 Registering Student Face

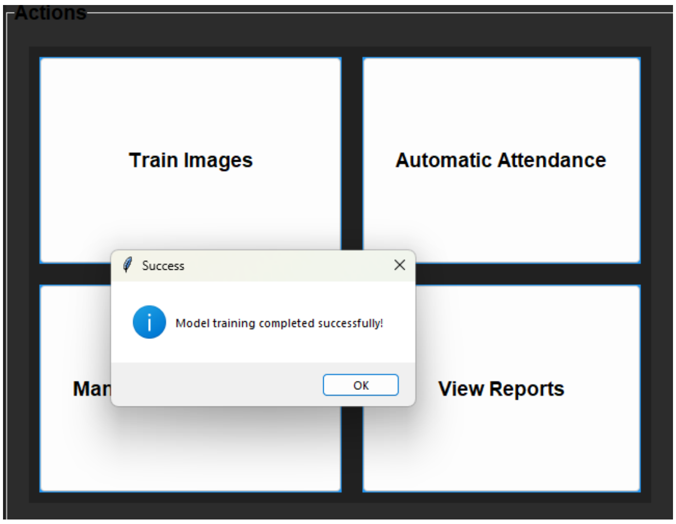

Fig 5.3 Training the Model

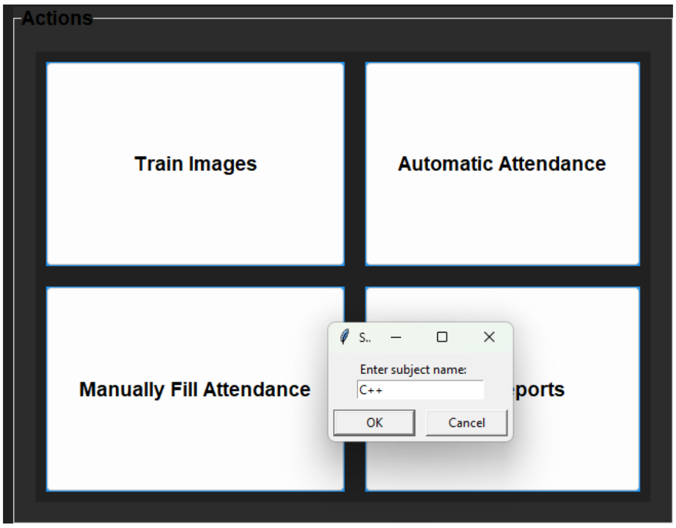

Figure 5.4 Entering the Subject

Fig 5.5 Attendance marked and updated in CVS file

6. CONCLUSIONS

Building a Face Recognition Attendance System that offers accurate, secure, and real-time attendance tracking with a seamless and user-friendly experience. It has been simplified for organizations to automatically detect, recognize, and log attendance without manual intervention.

For example, OpenCV and Dlib enable fast and accurate face detection, while Face Recognition library ensures secure facial verification. SQLite handles attendance data storage, maintaining reliability and security. Additionally, Sentry enhances error tracking and monitoring, reducing downtime and improving system stability.

7. Acknowledgement

Face Recognition Attendance System that offers superior accuracy in attendance tracking with a secure, fast, and easy-to-use experience. It has been simplified so that organizations can automate attendance logging, eliminate manual errors, and enhance security.

For example, OpenCV and Dlib power the face detection process, while the Face Recognition library ensures precise identity verification. SQLite/PostgreSQL manages attendance records with reliable data security. Additionally, Sentry enhances error tracking and monitoring, helping to reduce downtime and improve system stability

8. Reference

[1] Y.V.S.S. Avinash Kumar, CH. Venkata Lakshmi, G.B. Harish, Pattan Abdulla Khan, "FACE RECOGNITION BASED ATTENDANCE SYSTEM," Project Report, Department of Electronics and Communication Engineering, Andhra University, Visakhapatnam, 2021. Retrieved from https://ece.anits.edu.in/Project%20Reports%202020-21/Sec-B/B-13.pdf.Anits ECE

[2] Patel Rahul, "Face based attendance system using python and OpenCV," GitHub Repository, 2021. Retrieved from https://github.com/Patelrahul4884/Attendance-Management-system-using-face-recognition.GitHub

[3] Rakan Abuazh, Yousef Abdullah, Ali Alluwaimi, Yousef Aldakhail, "Face Detection and Recognition Student Attendance System," Senior Design Project Report, Department of Electrical Engineering, Prince Mohammad Bin Fahd University, 2020. Retrieved from https://www.pmu.edu.sa/attachments/academics/pdf/udp/coe/dept/ee/face_detection_system_report.pdf. pmu.edu.sa

[4] Turhan Can, "Intelligent Face Recognition Attendance System," GitHub Repository, 2021. Retrieved from https://github.com/turhancan97/Intelligent-Face-Recognition-Attendance-System.GitHub

[5] Vaibhav Bavaskar, Pallavi Sontakke, Harshal Vadode, "Face Recognition Attendance System," Project Report, Department of Computer Science and Engineering, SSGMCE Shegaon, 2023. Retrieved from https://www.ssgmce.ac.in/uploads/UG_Projects/cse/202324/Project%20Report%20Gr.%20No.%2010_2023-24.pdf.SSGMCE Shegaon

[6] Gunarakulan Gunaretnam, "Face Recognition Based Attendance System," GitHub Repository, 2021. Retrieved from https://github.com/gunarakulangunaretnam/face-recognition-based-attendance-system.GitHub

[7] Amlan Mohanty, "Face Recognition Based Attendance Management System with PowerBI Dashboard," GitHub Repository, 2021. Retrieved from https://github.com/amlanmohanty1/face-recognition-attendance-management-system-with-PowerBI-dashboard.GitHub

[8] "Project Report Face Recognition Attendance System," SlideShare Presentation, 2021. Retrieved from https://www.slideshare.net/slideshow/project-report-face-recognition-attendance-system/259036655. www.slideshare.net

[9] Achbi, "Real-time Facial Recognition Attendance System," GitHub Repository, 2021. Retrieved from https://github.com/Achbi/real_time_facial_recognition_attendance_system.GitHub

[10] "Smart Attendance System using Face Recognition," ResearchGate Article, 2023. Retrieved from https://www.researchgate.net/publication/378499613_Smart_Attendance_System_using_Face_Recognition