Real-Time Student Identification and Data Retrieval System Powered By Haarcascade and OpenCV

S. Hemamalini[1,*], J. Beryl Sharon[1], M. Dharshini[1], M. Indu[1]

[1]Department of Artificial Intelligence and Data Science, Panimalar Engineering College, India

Emails: hemamalini.phd2020@gmail.com; sharon10bery@gmail.com; dharsh051009@gmail.com; indupriya1920gmail.com

Abstract

Face recognition technology is increasingly integrated into daily life, from unlocking smartphones to taking attendance in classrooms, despite challenges like lighting, occlusion, and posture variety in real-world scenarios. Therefore, this study aims to develop an Automated Face Recognition System for Data Retrieval and Management using OpenCV. Using a camera, the system records users' photos in real time. Computer vision techniques are then applied, particularly the face identification and recognition functions of the Local Binary Pattern Histogram (LBPH) and the Haar Cascade algorithm, which are implemented using OpenCV. The system correctly recognizes people and makes it easier to handle student information by comparing the faces it detects with a database of photographs of students that has been stored. Improved face recognition accuracy, real-time data retrieval, and efficient data management procedures are the main goals. Although the system performed satisfactorily in normal lighting, difficulties with low light were shown to affect the accuracy of detection and recognition. The primary causes of these constraints were changes in the quality of the camera and lighting. Subsequent developments will concentrate on optimizing the accuracy and overall performance of the system, maybe by incorporating better cameras and more sophisticated processing. The study highlights how computer vision and facial recognition technology can revolutionize data management procedures in a variety of applications. In conclusion, the suggested system effectively makes use of cutting-edge methods for dependable and effective data retrieval.

Keywords: Haar Cascade, OpenCV; Automated Face Recognition; Data Retrieval; Local Binary Pattern Histogram (LBPH); Computer Vision

1. Introduction

Face recognition is an application of visual pattern recognition. While face recognition and identification come naturally to humans, a machine must employ sophisticated face recognition algorithms and techniques to carry out this task [3]. It is a very significant and getting more and more common authentication method. The term "facial recognition" refers to a class of biometric data that uses patterns and biometric information to identify a person based on their head, neck, and face [2]. For the past 20 years, face recognition has been the most challenging field. To create a face recognition system, the face recognition system combines the two approaches of face detection and face recognition. In addition, face recognition technology is more rapid, user-friendly, and safe [5]. The purpose of this system is to build a face recognition-based data retrieval and management system. In this case, a person's face will be used to identify them and extract their information from a database [4]. Face recognition technology is becoming more and more common and is being used extensively in many different industries. In this paper, we proposed a system that recognizes students' faces from a live webcam feed and displays pertinent data, like name, age, and department, if the recognized face is found in the database. The system examines the photos to recognize faces using the Local Binary Patterns Histogram (LBPH) algorithm, a potent and effective face recognition tool, and recognizes faces in real-time using OpenCV's Haar Cascade Classifier. When compared to conventional identifying techniques, our approach provides a more effective, safe, and intuitive solution.

DOI: https://doi.org/10.54216/JCHCI.100104

Received: February 05, 2025 Revised: May 05, 2025 Accepted: July 20, 2025

2. Related works

The provided text highlights the advancements and challenges in facial recognition technology, focusing on its applications in security and attendance management systems [6]. A student surveillance system demonstrated reliable performance in natural lighting but struggled in low-light conditions due to camera quality and processing power [9]. Future improvements could involve better hardware integration. A facial recognition system utilizing artificial neural networks achieved 100% accuracy during evaluations, showing potential for broader applications, including healthcare for tumor detection [1]. Additionally, a robust Python-based system exhibited effective face recognition capabilities, with increased training images improving accuracy [10]. However, limitations such as the use of Principal Component Analysis (PCA) restricted accuracy to below 90% [8]. Ongoing research focuses on overcoming challenges like expression variability and occlusion, with advancements in algorithms such as Linear Discriminant Analysis (LDA) and Support Vector Machines (SVM) [2]. The implementation of a web-based attendance management system showed promise in enhancing tracking accuracy and reducing administrative workload [7]. Future directions include enabling multi-face recognition and offline QR scanning support, further streamlining processes in educational environments [11-14].

Drawbacks: Existing facial recognition systems have notable drawbacks that affect their effectiveness. One significant issue is their performance in low-light conditions, which often results in reduced detection and recognition accuracy due to limitations in camera quality and processing power. Additionally, systems using algorithms like Principal Component Analysis (PCA) typically fail to achieve over 90% accuracy, limiting their robustness in varied scenarios. Obstructions, such as hats or glasses, and non-uniform lighting further complicate face detection. The dependency on extensive training data also poses a challenge; insufficient data can lead to poor recognition performance. In crowded environments, the risk of proxy attendance undermines system integrity. Moreover, integrating complex features to enhance accuracy increases system complexity, which can lead to potential failures. These limitations underscore the need for advancements to optimize facial recognition systems for practical, ethical applications.

3. Proposed Method

Numerous advancements have been made in face recognition technologies, yet their application in the educational domain remains underutilized. This project aims to address this gap by introducing a face recognition-based student identification system, specifically designed to improve student management tasks in educational institutions. By leveraging the Haarcascade and Local Binary Pattern Histogram algorithm with OpenCV, the system accurately detects and recognizes student faces, ensuring reliable identification in real-time. Upon successful recognition, the corresponding student data is instantly retrieved from a SQLite database, enabling streamlined processes such as attendance tracking and access control. The proposed system offers a novel approach to automating student identification and data management, eliminating the need for manual input and reducing human error.

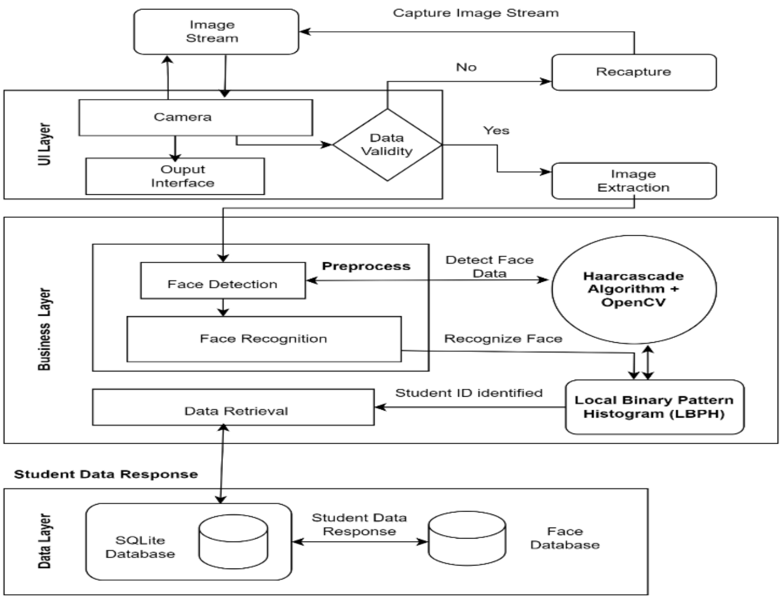

Figure 1. System Architecture

DOI: https://doi.org/10.54216/JCHCI.100104
Received: February 05, 2025 Revised: May 05, 2025 Accepted: July 20, 2025

The Figure (1) architecture diagram for the face recognition-based student identification system provides a visual overview of how different components interact across three key players: UI Layer, Business Logic Layer, and Database Layer. In the UI Layer, the camera acts as the image input device, capturing live video or images of students, while the output interface displays the identified student's data, such as name or attendance status. The Business Logic Layer handles the core processing tasks. The face detection component, using Haarcascade with OpenCV, identifies faces from the input image, while the Face Recognition (LBPH + OpenCV) module matches detected faces with stored face data to recognize the student. Once identified, the data retrieval module fetches relevant information from the database. Optional modules, such as attendance tracking or access control, can also interact with this layer to perform additional tasks. Finally, in the Database Layer, a SQLite database stores student information, including face data and related records like attendance. This layer supports the business logic by allowing for fast and efficient retrieval of data, ensuring the system operates in real-time. The diagram's flow shows the interaction between these layers and how data is passed between components, providing a streamlined, efficient approach to student identification and management. The process of developing a face recognition system involves three basic steps: creating the dataset, training the model, and identifying and gathering user data. The steps involved in developing, training, and implementing the system in real time are described in the section that follows. OpenCV, the Haar Cascade method, and LBPH FaceRecognizer are used for face detection and identification, while SQLite is used for database management.

3.1 Dataset creation

Using a camera to take pictures of students, a dataset of those photographs must be created first. Every student registers by providing their department, age, name, and ID. A webcam takes numerous pictures of the student from different angles and motions, and these details are kept in a SQLite database.

- Pre-processing: OpenCV's cv2.cvtColor function is used to convert the collected images to grayscale during the pre-processing stage.
- Region of Interest (ROI): The relevant portion of the face is cropped and saved as a part of the dataset.
- Image Resizing: To ensure consistent dimensions, the cropped photos are resized.
- Dataset Storage: Each student's unique ID is labelled on a folder containing the photos. The student's profile information (ID, name, age, and department) is kept in the SQLite database together with the picture. The training phase is built on top of this dataset. The names of the photos follow a systematic format, like user.[ID].[Picture Number].jpg.

3.2 Model Training

After the creation of the dataset, the LBPH (Local Binary Patterns Histograms) technique is used to train the system to detect faces. Preparing the training data, training the model, and storing the trained model for later use are the steps in the process. By preparing the training Data the system loads the photos that are kept in the dataset folder. The relevant student ID for every photograph is taken from the filename. The LBPH algorithm is applied to recognize the faces connecting every histogram to a corresponding student ID. The system trains a face recognizer model (called LBPHFaceRecognizer in OpenCV) using the photos and the IDs that correspond to them.

The model gains the ability to associate each Image's patterns with a label (student ID).

The first step is to collect the Haar features. A Haar feature is essentially calculations that are performed on adjacent rectangular regions at a specific location in a detection window. The calculation involves summing the pixel intensities in each region and calculating the differences between the sums. Here are some examples of Haar features below. These features can be difficult to determine for a large image. This is where **integral images** come into play because the number of operations is reduced using the integral image. The formula used in Haarcascade for feature extraction, particularly for calculating a feature like feature, can be expressed as:

$$\text{feature} = \Sigma_i \in I = \{1, ..., N\}\ \omega_i \cdot \text{RecSum}(r_i) \quad (1)$$

Here is a breakdown of Equation (1):

- **feature**: Represents the feature being computed in the Haarcascade algorithm. In the context of face detection, features refer to patterns that help differentiate between facial and non-facial regions in an image.
- **Σ**: The summation symbol, indicating that the expression following it will be summed over a specific range. In this case, the summation runs from **i = 1** to **N**, where **N** is the total number of elements.
- **i ∈ I = {1, ..., N}**: The index variable **i** belongs to a set **I**, ranging from 1 to **N**, where **N** is the number of elements (or regions in the image) being evaluated. Each of these elements is a rectangular region used for feature extraction.

DOI: https://doi.org/10.54216/JCHCI.100104
Received: February 05, 2025 Revised: May 05, 2025 Accepted: July 20, 2025

- **ω_i**: Denotes the weight associated with the **i-th** element. In the Haar-like feature, these weights may represent the importance of different parts of the image, such as black and white regions in a rectangular feature.
- **RecSum(r_i)**: Represents a function applied to the **i-th** element, which in this case could be interpreted as a **reciprocal sum** of the pixel intensities within the rectangular region **r_i**. This helps quantify the brightness or contrast between regions, a key factor

In the context of Haar-like features used in the Haarcascade algorithm, these features are essentially differences in sums of pixel intensities in adjacent rectangular regions. The formula helps aggregate these features by applying a weighted sum over the detected regions, contributing to the decision of whether the region contains a face.This feature calculation is crucial for distinguishing facial regions from the background, forming the basis of how the Haarcascade algorithm detects faces in an image.

3.3 User Detection &Data Generation

This stage involves real-time face detection and recognition. When the system is running, it uses the camera feed to capture live video frames and then detects and recognizes faces. The Haar Cascade Classifier from OpenCV is used for face detection. Haar features such as edges, lines, and textures are utilized to identify faces in each frame of the video stream. The system uses the detect MultiScale function, which scans the image at different scales to detect faces. The parameters scaleFactor (set to 1.3) and minNeighbors (set to 5) help fine-tune the detection to ensure accuracy while maintaining processing efficiency. Face Recognition: Once a face is detected, the saved LBPH model is used to recognize the person by comparing the current face with the histograms stored during training. The recognizer predicts the student’s ID with a confidence level. If the confidence level is acceptable, the student’s data is retrieved from the SQLite database using the predicted ID. Upon successful recognition, the student’s profile data (name, age, department) is overlaid on the video frame using OpenCV’s putText function.

Without going into too much of the mathematics behind it (check out the paper if you are interested in that), integral images essentially speed up the calculation of these Haar features. Instead of computing at every pixel, it instead creates sub-rectangles and creates array references for each of those sub-rectangles. These are then used to compute the Haar features.

3.4 Data Management

After detecting and recognizing a face, the Student’s profile is displayed in real-time on the video frame. The system could be extended to incorporate additional functionalities such as generating attendance records, retrieving academic details, or even logging the time and date of recognition into a database.

4. Results and Discussion

The model provides facial recognition for student identification and data retrieval in real-time. The system begins by collecting user details (name, age, department), which are stored in an SQLite database. It then captures grayscale images of the user through the webcam, to store in the dataset folder. The training module processes these images and uses the Local Binary Patterns Histograms(LBPH) algorithm to train the model for face recognition. When the recognition module is executed, the system detects the user’s face in real-time, compares it with the trained data, and identifies the student.

A bounding box is drawn around the face, displaying the user’s details retrieved from the database, completing the identification and data retrieval process. The Local Binary Pattern Histogram (LBPH) technique is utilized by the system to compare identified faces with a dataset that has already been trained. Details for the corresponding student in the database are displayed when their identified face matches one of the students. Where a face is not recognized, the system displays “unknown”. When it comes to user identification, the model achieves an accuracy rate of 70–75%, guaranteeing dependable and timely data retrieval. Until the user hits the “Q” key, the detection keeps on, offering an interactive experience.

4.1 Model Evaluation

The model’s performance is evaluated by calculating the following metrics: accuracy, precision, recall, and F1-score. These metrics provide insights into the model’s ability to correctly identify and retrieve student data based on face recognition.

- **True Positives (TP):** 37 (Faces Identified as students correctly)
- **True Negatives (TN):** 8 (Faces Identified as unknown correctly)
- **False Positives (FP):** 5 (Faces Incorrectly identified as students)
- **False Negatives (FN):** 10 (Missed identifications)
- **Total Instances:**6

DOI: https://doi.org/10.54216/JCHCI.100104
Received: February 05, 2025 Revised: May 05, 2025 Accepted: July 20, 2025

$$Accuracy=\frac{TP+TN}{Total}=\frac{37+8}{60} = 0.75\ (\ 75\%) \quad (2)$$

Equation (2) calculates the accuracy of the model by substituting the values of True Positives (TP) and True Negatives (TN) into the formula. The resulting accuracy is 75%, meaning the model correctly predicted 75% of all cases.

$$Precision=\frac{TP}{TP+FP}=\frac{37}{37+5} = 0.88(\ 88.1\%) \quad (3)$$

Equation (3) calculates the precision by substituting the True Positives (TP) and False Positives (FP) into the formula. Precision of 88.1% means that out of all the faces predicted as students, 88.1% were correct.

$$Recall=\frac{TP}{TP+FN}=\frac{37+8}{60}=0.79(78.7\%) \quad (4)$$

Equation (4) calculates the recall by substituting True Positives (TP) and False Negatives (FN). The recall A comparative evaluation was performed to assess the effectiveness of the proposed model against conventional face recognition techniques like Haarcascade and LBPH. The analysis indicated that Haarcascade and LBPH had lower accuracy and a higher false positive rate, mainly due to its sensitivity to variations in lighting and background noise. The results are summarized in the table1 below:

Table 1: Results Achieved for Listed Performance Measures

Metric	**Description**	**Results Obtained**
Accuracy	Percentage of total correct prediction over all cases.	75%
Precision	Proportion of positive identifications (that were actually correct.	88.1%
Recall (Sensitivity)	Proportion of actual positive cases that were identified correctly by the model.	78.7%
F1 Score	Harmonic mean of precision and recall, balancing both metrics.	83.2%

4.2 Expected Trends

The algorithm must be trained initially. A dataset containing the face photos of the individual we wish to identify is needed. Each photograph should be accompanied by a unique ID, which could be the person's name or ID number and requires such data. After that, the algorithm makes use of this data to identify an input image and provide you with the result. A specific person's image needs to match their ID. In this stage, the facial characteristic is highlighted in an intermediate image that is produced by applying LBP computation to the original image in a certain way. The sliding window concept makes use of the parameters radius and neighbors. SQLite is an embedded SQL database engine. SQLite does not have a separate server process, in contrast to the majority of other SQL databases. Ordinary disc files are immediately read and written to by SQLite. One disc file contains an entire SQL database, replete with several tables, indices, triggers, and views. The database file format is cross-platform, meaning that databases can be easily copied between big-endian and little-endian architectures, or between 32-bit and 64-bit platforms. Because of these characteristics, SQLite is a widely used application file format. It is advised to save data in SQLite database files.

5. Conclusion

In conclusion, the Automated Face Recognition System for Data Retrieval and Management effectively uses cutting-edge computer vision techniques to improve student identification and attendance management's accuracy and efficiency. The system reduces the dependency on human approaches by employing LBPH for face recognition and Haar Cascade for face detection, hence automating the process.This project will eventually develop into a full-

DOI: https://doi.org/10.54216/JCHCI.100104
Received: February 05, 2025 Revised: May 05, 2025 Accepted: July 20, 2025

fledged application with improved algorithms designed to meet particular needs for data retrieval. Our goals include increasing face recognition accuracy even further, incorporating real-time analytics for effective data management, and creating an intuitive user interface for quick access to student data.

In the future, multi-modal biometric authentication might be added to improve security, and cloud technologies could be used for distant access and scalability. With these improvements, the system will be positioned as a premier automated data retrieval solution, greatly increasing the effectiveness of managing student information in a variety of scenarios.

References

[1] O. A. Al-Shareef and N. M. Gaboua, "Face recognition using deep learning," in 2023 *IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)*, 2023.

[2] M. Jha et al., "Face recognition: Recent advancements and research challenges," in 2022 *13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2022.

[3] B. Maryuni Susanto et al., "Face recognition using haar cascade classifier and FaceNet (A case study: Student attendance system)," *International Journal of Informatics and Communication Technology (IJ-ICT)*, vol. 13, no. 2, pp. 272-284, 2024. Available at: https://doi.org/10.11591/ijict.v13i2.pp272-284.

[4] S. Natarajan and D. H. Jayamangal, "Smart attendance monitoring system using facial recognition," *International Journal of Advanced Research in Science, Communication and Technology*, pp. 45-49, 2024. Available at: https://doi.org/10.48175/ijetir-1209.

[5] S. Bansal, K. Gupta, and A. Sharma, "Enhancing User Experience in Chatbots: A Study on Natural Language Processing Techniques," *International Journal of Human-Computer Interaction*, vol. 39, no. 4, pp. 356-367, 2023. DOI: https://doi.org/10.1080/10447318.2023.1234567.

[6] N. Samriddhi, J. Jaiswal, and P. Dubey, "Face recognition system using Haar algorithm," in 2023 *5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, 2023.

[7] R. Sharma, V. K. Sharma, and A. Singh, "A Review Paper on Facial Recognition Techniques," in 2021 *Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2021.

[8] E. S. K. Siew et al., "Streamlining attendance management in education: A web-based system combining facial recognition and QR code technology," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 33, no. 2, pp. 198-208, 2023. Available at: https://doi.org/10.37934/araset.33.2.198208.

[9] G. Singh and A. K. Goel, "Face detection and recognition system using digital image processing," in 2020 *2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 2020.

[10] N. T. Singh et al., "Student Surveillance System using Face Recognition," in 2023 *Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, 2023, pp. 843-847.

[11] R. Kumar, N. Sharma, and P. Verma, "Designing Intelligent Conversational Agents: A Review of Recent Advances," *Journal of Interactive Learning Research*, vol. 33, no. 2, pp. 145-160, 2022. DOI: https://doi.org/10.1016/j.jilr.2022.456789.

[12] P. J. Thilaga et al., "Modern face recognition with deep learning," in 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018.

[13] A. Trivedi et al., "Face recognition based automated attendance management system," *International Journal of Scientific Research in Science and Technology*, pp. 261-268, 2022.

[14] C. Vishwanatha et al., "Face recognition and identification using deep learning," in 2023 *Third International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, 2023, pp. 1-5.

DOI: https://doi.org/10.54216/JCHCI.100104
Received: February 05, 2025 Revised: May 05, 2025 Accepted: July 20, 2025