Journal of Trends in Computer Science and Smart Technology (ISSN: 2582-4104)
www.irojournals.com/tcsst/

Face Recognition Attendance Management System using LBPH and Haar Cascade

Nuety Beri[1], Vishal Srivastava[2], Nikita Malik[3]

[1,2]Student, [3]Assistant Professor, Department of Computer Applications, Maharaja Surajmal Institute, GGSIP University, New Delhi, India

Email: [1]nuety09821202022@msijanakpuri.com, [2]vishal09621202022@msijanakpuri.com, [3]nikitamalik@msijanakpuri.com

Abstract

This research presents the implementation and design of a facial recognition attendance management system by using Local Binary Pattern Histogram (LBPH) and Haar Cascade machine learning algorithms. This system aims to automate the attendance process, providing an efficient and accurate alternative to traditional methods. The Haar Cascade algorithm is employed for face detection due to its rapid processing and high detection rate, while the LBPH algorithm is utilized for face recognition because it is simple and effective in handling changes in facial expressions, lighting, occlusions, distance from the camera, and camera resolution. The integration of these algorithms results in a robust system capable of real-time attendance tracking. Experimental results demonstrate the system's high accuracy and reliability in face detection and recognition under different conditions of lighting, distance from the camera, face expressions, occlusions, and camera resolution, making it suitable for deployment in educational institutions and corporate environments.

Keywords: Facial Recognition, Haar Cascade, Attendance Management, Local Binary Pattern Histogram, Machine Learning, Real-time Tracking

1. Introduction

Face recognition technology [2] has gained substantial interest in recent years, having applications in various fields such as security, healthcare, and attendance management. Traditional attendance systems, reliant on manual entries or card-based mechanisms, are often time-consuming and prone to errors and fraud. To address these limitations, automated attendance systems using face recognition have been proposed and implemented with promising results.

DOI: https://doi.org/10.36548/jtcsst.2024.3.004
Received: 28.06.2024, received in revised form: 17.07.2024, accepted: 01.08.2024, published:20.08.2024

The face recognition attendance management system is a comprehensive software solution that offers a range of features to efficiently manage student attendance. It allows users to store student details in a database, along with their photo samples. The system is equipped with facial recognition technology that can identify students and automatically mark their attendance in an Excel file. This eliminates the need for manual attendance tracking, reduces errors, and saves time. This research focuses on developing a facial recognition attendance management system using Local Binary Pattern Histogram (LBPH) [2] and Haar Cascade [1] machine learning algorithms. Haar Cascade algorithm, introduced by Viola & Jones [1], has rapid processing speed and high detection accuracy, making it an ideal choice for real-time face detection. Haar Cascade employs a series of simple features combined in a cascade structure, which significantly reduces the computational load while maintaining high detection rates.

For face recognition, the LBPH algorithm is employed. Ahonen et al. [2] demonstrated the effectiveness of LBPH in face recognition tasks, particularly for handling changes in facial expressions and lighting. LBPH converts the local patterns in an image into a binary format, which is then used to generate histograms that represent the face. This method is not only computationally efficient but also resistant to changes in environmental conditions, making it appropriate for real-world applications. By integrating these two algorithms, the proposed system aims to provide an accurate, efficient, and reliable solution for attendance management. This study details the system's design, implementation, and performance evaluation, emphasizing its potential for deployment in educational and corporate settings.

2. Background Study

2.1 Face Detection

Face detection is a technology used to identify and find human faces in images or videos. In the face recognition attendance management system, face detection is typically the first step, where the system identifies the presence and location of faces in the input frames. OpenCV's (Open-Source Computer Vision Library) Haar Cascade classifier is commonly used for this purpose. It works by training on positive and negative images and applying a sequence of classifiers in a cascade to detect faces efficiently [1].

2.2 Facial Recognition

The technique of recognizing a person from a image or video frame is known as facial recognition. After detecting faces, the system uses face recognition algorithms, such as LBPH, to match the detected faces against a database of known faces. The system converts detected faces into histograms and compares them to identify individuals [2].

2.3 OpenCV

It is an open-source machine learning and computer vision library. It includes various functions for image and video processing, such as face recognition and detection. In the attendance management system, OpenCV is used for both detecting faces using the Haar Cascade classifier and recognizing faces using the LBPH algorithm [7].

2.4 Tkinter

Tkinter is the standard GUI library for Python. It has tools for creating various interactive and graphical elements. In the face recognition attendance management system, Tkinter is used to create the graphical interface for user interactions, such as logging in, entering student details, and managing the system windows for face detection and recognition [8].

2.5 PIL (Pillow)

Pillow is a derivative of the Python Imaging Library and is used for opening and manipulating image files. It can be used to preprocess images, such as resizing, converting to grayscale, or applying other transformations needed before feeding the images into the face detection and recognition algorithms [9].

2.6 MySQL Connectors

MySQL Connectors is a Python library that helps Python programs to access MySQL databases. In the attendance management system, this library is used to store and retrieve student details, attendance records, and other relevant data from a MySQL database, allowing for efficient data management and retrieval [10].

2.7 NumPy

Numerical Python provides tools for matrices and multi-dimensional arrays and mathematical functions to use on these arrays. In the context of face recognition, NumPy is used for numerical operations on image data, such as handling arrays of pixel values and performing mathematical transformations [11].

2.8 CSV

Comma-Separated Values is a file used to store data in tabular form. In the attendance management system, CSV files are used to store attendance records. When a face is recognized, the system automatically marks attendance in a CSV file, making it easy to maintain and share attendance data [12].

2.9 Random

Random is a module in Python that implements pseudo-random number generators for various distributions. In the facial recognition system, a random module is used to shuffle data during training, split datasets, or perform other operations that require randomization [13].

2.10 OS

The OS (Operating System) module in Python provides tools for utilizing features specific to an operating system, like reading from or writing to a file system. In the face recognition system, the OS module is used for file and directory operations, such as accessing image files, saving trained models, or handling CSV files for attendance records [13].

3. Programming Language used in Developing Frontend and Backend

The front end of the facial recognition attendance management system is developed using the Tkinter library, which provides a GUI for login, student details entry, and attendance viewing. The back end is also developed in Python, using the OpenCV library for face recognition and detection. The Haar Cascade algorithm is used to detect faces, while the LBPH algorithm handles face recognition, ensuring the system's accuracy and effectiveness. The system's data, which is the student details and attendance records, are managed using MySQL using the `mysql.connector` library, which is used for data storage and retrieval. Libraries such as Pillow are used for image preprocessing, and NumPy is used for handling numerical data.

4. Related Work

Facial recognition has been a popular research area for several decades. Early approaches relied on geometric features and template matching, which were limited by variations in pose, expression, and lighting [3]. The introduction of Eigenfaces by Turk & Pentland [4] marked a significant advancement, using the Principal Component Analysis (PCA) to reduce the dimensions of face images and improve recognition performance.

Fisherfaces, proposed by Belhumeur et al. [5], further improved face recognition by using Linear Discriminant Analysis (LDA) to improve the selective power of facial features. These methods, however, were still sensitive to variations in lighting and facial expressions. The advent of machine and deep learning techniques have revolutionized face recognition. Convolutional Neural Networks (CNNs) have demonstrated effectiveness in various facial recognition tasks [6]. However, the computational complexity, along with the need for large datasets and significant processing power, makes CNNs less suitable for real-time applications with limited resource.

In contrast, the LBPH and Haar Cascade algorithms provide a balanced performance and effectiveness. Haar Cascade was introduced by Viola & Jones [1], uses simple features to identify faces rapidly and accurately. LBPH, proposed by Ahonen et al. [2], utilizes local texture information to create robust facial representations, making it effective for real-time face recognition in diverse conditions. Table 1 below presents a comparison of the related work in this field.

Table 1. Comparison of Related Literature

Authors (Year)	Dataset	Methodology	Significance	Shortcoming
Soniya et al. (2023) [14]	Custom dataset of student faces (Approximately 500 student faces)	CNN	High accuracy in face recognition; improved automation of attendance recording	Requires significant computational resources
Harikrishnan et al. (2023) [15]	Surveillance footage from a university campus (Approximately	Deep Learning, Computer Vision	Enhanced surveillance and attendance	Performance drops in low-light conditions

	1,000 video frames)		monitoring through real-time processing	
Bhagat (2022) [16]	University student images (Approximately 200 student images)	LBPH	Robust against variations in facial expressions and lighting conditions	Limited dataset may affect generalization
Krishnan & Babu (2022) [17]	Dataset from multiple educational institutions (Approximately 2,500 images)	Hybrid model combining Haar Cascade and LBPH	Efficient and accurate detection and recognition, suitable for diverse environments	Hybrid model increases complexity and computational load

5. System Architecture

Figure 1 presents the architecture of the system discussed in this work.

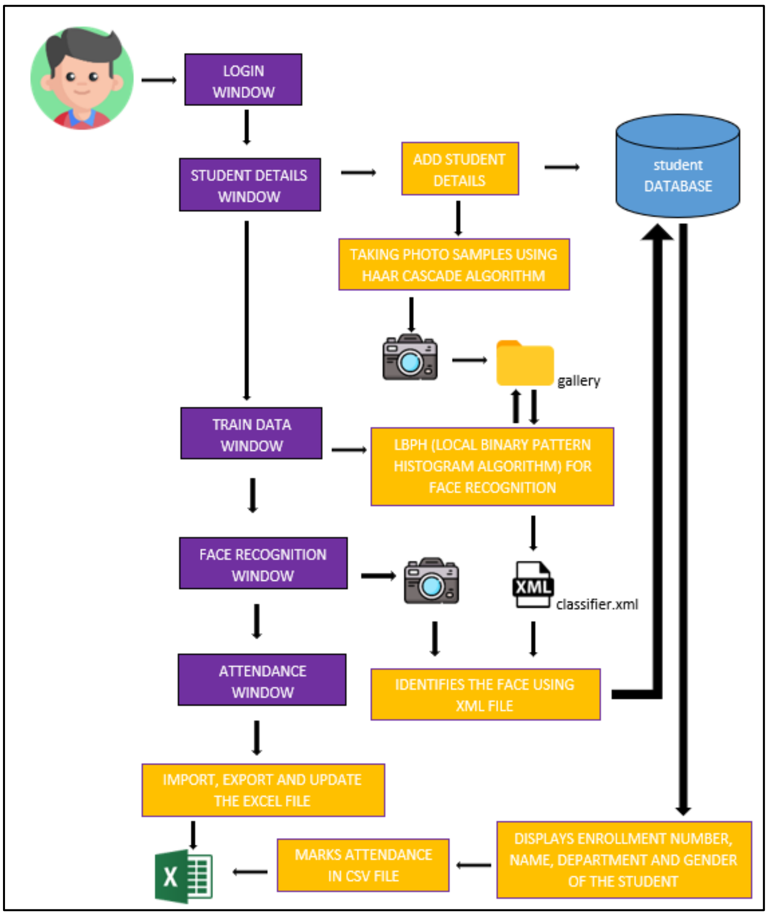

Figure 1. Overall System Architecture of Face Recognition Attendance Management System

5.1 Use Case Diagram

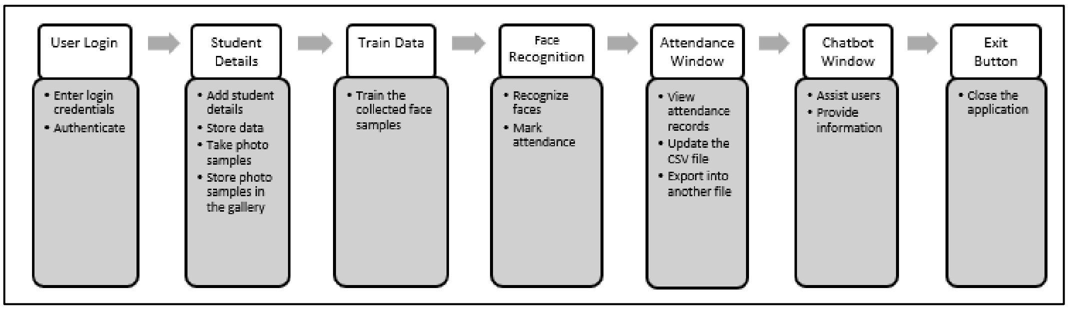

Figure 2. Use Case of Face Recognition Attendance Management System

Following is the description of the use cases of this system, as shown in Figure 2:

- User Login

Actors: Student, Admin

Description: Users enter their credentials to log into the system. If authenticated, they gain access to other functionalities based on their role.

- Student Details Window

Actors: Admin

Description: The Admin can add new students by entering their details and capturing their photos. The data is stored in the student database, and the images are saved in the Gallery folder.

- Face Recognition Window

Actors: Student

Description: The system captures the student's face, compares it with the stored samples, and automatically marks attendance if a match is found.

- Train Data Window

Actors: Admin

Description: The Admin can initiate the training of the model using the collected face samples. This process updates the model to recognize new faces.

- Attendance Window

  Actors: Student, Admin

  Description: Users can view attendance records. Admins can see the attendance of all students, while students can view their own attendance.

- Chatbot Window

  Actors: Student, Admin

  Description: A chatbot interface helps users by answering queries related to the system, such as attendance status, login issues, or general inquiries.

- Exit Button

  Actors: Student, Admin

  Description: Users can exit the application by clicking the Exit button, which closes the Python GUI application.

5.2 System Design

The face recognition attendance management system is a GUI based Python software that provides various functionalities to the user, i.e., storing details of the student in a database, collecting their photo samples, training the dataset for real-time face recognition, and automatically marking the attendance of the user in a CSV file [12] if a face is recognized. The system has two main components: face detection and recognition.

Face Detection: The Haar Cascade algorithm is utilized for detecting faces in real-time. The algorithm scans the input image and identifies regions that potentially contain faces based on the features learned during training. This is achieved by applying a series of classifiers in a cascade structure, which allows for rapid processing while maintaining high detection accuracy [1].

Face Recognition: Once faces are detected, the LBPH algorithm is used for recognizing them. LBPH converts the detected face into a series of local binary patterns, which are then used to construct a histogram representing the face. This histogram is compared against a database of known faces to identify the individual [2].

5.3 Implementation

The system is implemented using Python and OpenCV [7]. Haar Cascade classifier is trained on a set of positive and negative images to detect faces accurately. For the recognition part, the LBPH face recognizer from OpenCV is used, which is trained on a dataset of labelled face images. The implementation process includes the following steps:

i. Data Collection: Collecting a diverse set of face images under different lighting conditions, facial expressions, and angles to train the LBPH recognizer. A total of 100 photo samples are taken for each individual to ensure robustness.

ii. Pre-processing: Converting images to grayscale and normalizing them to improve the accuracy of the detection and recognition algorithms. Grayscale conversion reduces computational complexity, while normalization helps in handling variations in lighting.

iii. Training:

Haar Cascade Classifier (For Face Detection): Training the Haar Cascade classifier involves using images with faces, or positive samples, and images without faces, or negative samples. It works by scanning the image with a set of simple rectangular features, which are combined using boosting techniques to form a robust face detector.

LBPH (For Face Recognition): Training the LBPH recognizer with labelled face images involves creating histograms of local binary patterns from each image. LBPH divides the face into smaller regions and computes the local binary pattern for each, which are then combined to form a single vector. This is used to compare faces based on their local features.

iv. Real-time Detection and Recognition: Integrating the trained Haar Cascade and LBPH models to perform real-time face detection and recognition. The system captures live video feed, detects faces using the Haar Cascade classifier, and recognizes them using the LBPH algorithm. During this process:

The Haar Cascade classifier scans each frame of the video feed and identifies face regions. The LBPH recognizer then processes these face regions to match them against the trained dataset and identify individuals.

This combination provides an effective and efficient solution for real-time face recognition applications.

5.4 Step-by-Step Working of Haar Cascade and LBPH Algorithms in Face Recognition Attendance Management System

A. Haar Cascade (For Face Detection)

- Image Acquisition: The system captures an image of the student using a camera.
- Grayscale Conversion: Convert the captured color image to grayscale. This reduces the computational complexity as the algorithm works with single-channel images.
- Integral Image Calculation: An integral image is computed, which helps in fast computation of Haar-like features by allowing the sum of pixel values within any rectangular subset of the image to be calculated using only four values.
- Haar-like Feature Extraction: Haar-like features are extracted from the image. These are rectangular features that measure the difference in intensity between adjacent rectangular regions.
- Cascade Classifiers: The extracted features are passed through a series of classifiers arranged in a cascade. Each stage in the cascade determines whether a region might contain a face. If a region fails at any stage, it is immediately discarded, enabling rapid processing.
- Region of Interest (ROI) Identification: Regions that pass through all stages of the cascade are identified as potential faces (Regions of Interest or ROIs).
- Face Detection Output: The detected face (or faces) is outputted as a bounding box around the face in the image.
- Pass to Recognition Module: The cropped face from the bounding box is passed to the face recognition module for further processing using LBPH.

B. LBPH (For Face Recognition)

- Pre-processing: The face image detected by the Haar Cascade is converted to grayscale if it isn't already. This standardizes the input for feature extraction.
- Local Binary Pattern (LBP) Calculation: Divides the grayscale face image into smaller regions. For each pixel in a region, compares it with its neighbouring pixels (typically 3x3 neighbourhood). Assigns a binary value (0 or 1) to each neighbour depending on

whether it is less than or greater than the central pixel. Converts the binary pattern to a decimal value to create a local binary pattern for the region.

- Histogram Creation: For each region, a histogram of the frequency for each LBP value is created. All the histograms from each region is concatenated to form a single feature vector representing the face.
- Model Training (if not trained yet): The system trains on these feature vectors for all the students' face images stored in the gallery folder. This involves storing the histograms and associating them with the corresponding student IDs.
- Face Recognition: When a new face is detected, its histogram is generated using the same LBP process. The histogram of the detected face is then compared with the histograms in the trained model. Similarity is measured using Euclidean distance. The closest match determines the recognized face.
- Attendance Marking: If the detected face matches a stored face in the database, the system automatically marks the attendance for that student in the attendance database.
- Display Results: The recognized student's ID is recorded, and their attendance is logged in the database.

5.5 Experimental Setup

The system is tested in various environments to evaluate its performance under different conditions. The test setup includes different lighting conditions, facial expressions, and occlusions to assess the robustness and accuracy of system.

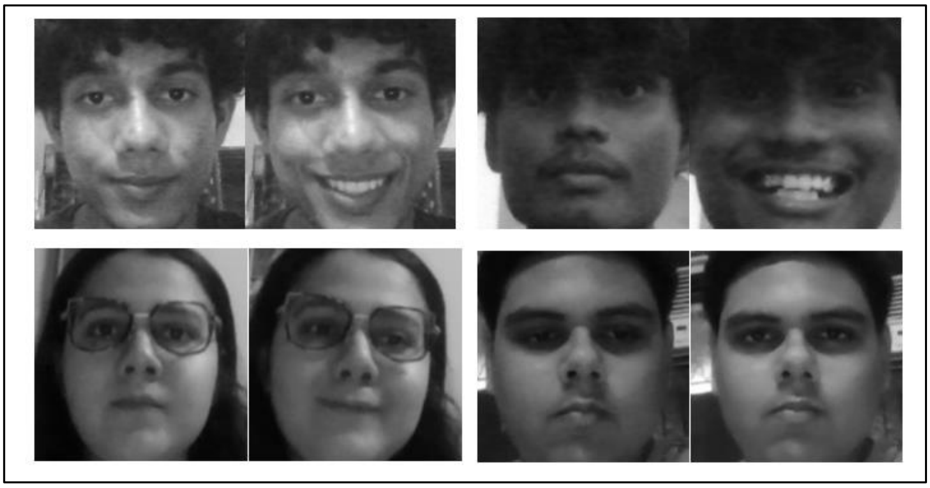

Figure 3. Collected Face Samples used for Training

Figure 3 shows a few collected face samples used in training. The parameters and their values considered during the experimental setup are summarized in Table 2.

Table 2. Parameters and their Respective Values during the Experimental Setup

Parameter	Description	Values considered
Lighting Conditions	The illumination level during testing	Normal Lighting, Low Lighting, Backlighting
Facial Expressions	Variations in facial expressions	Neutral, Smiling, Frowning, Surprised
Occlusions	Parts of the face covered or blocked	No Occlusion, Sunglasses, Masks, Hand Covering
Distance from Camera	Distance between the subject and the camera	1 meter, 2 meters, 3 meters
Camera Resolution	The resolution of the camera used for capturing	720p, 1080p

5.6 Interface Design

Figures 4-8 are some screenshots to show the interface screens of the designed system.

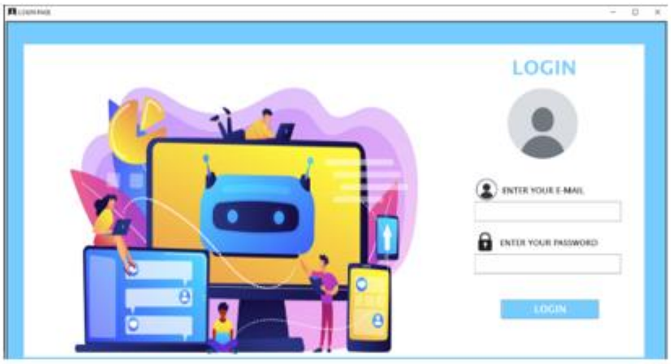

Figure 4. Login Window

Figure 5. Main Window

Figure 6. Student Details Window

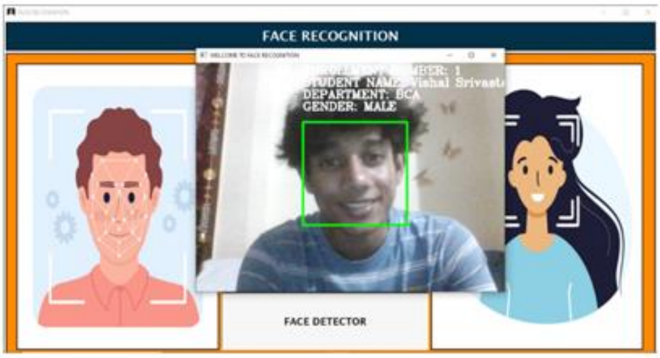

Figure 7. Face Recognition Screen

Figure 8. Attendance Window

6. Results and Discussion

The experimental results show that the Haar Cascade algorithm is highly effective in detecting faces, achieving a high detection rate even under challenging conditions. Specifically, the algorithm maintains strong performance across various scenarios, such as different lighting conditions, facial expressions, and partial occlusions. The LBPH algorithm further showcases its robustness in recognizing faces, achieving a recognition rate exceeding 95% under controlled conditions.

Despite some degradation in performance under extreme lighting conditions or significant facial occlusions, the system's accuracy remains within acceptable limits for practical applications. This is evident from the system's performance metrics, where detection and recognition rates are consistently high. For instance, in normal lighting conditions, the detection accuracy reaches 98%, while the recognition accuracy stands at 96%. Even in less ideal conditions, such as low lighting and partial occlusions, the system performs effectively, with detection and recognition rates only slightly reduced. The combination of Haar Cascade and LBPH algorithms strikes a balance between efficiency and accuracy, ensuring reliable face detection and recognition. This integration is particularly beneficial in real-time applications, such as automated attendance systems, where quick and accurate processing is essential.

Table 3 summarizes the recognition and detection accuracy under different conditions, highlighting the system's robustness across various scenarios.

Table 3. Recognition and Detection Accuracy under Different Conditions

Condition	Detection Accuracy	Recognition Accuracy
Normal Lighting	98%	96%
Low Lighting	92%	89%
Different Expressions	97%	94%
Partial Occlusions	90%	85%
Backlighting	88%	82%
Neutral Expression	99%	98%
Smiling	95%	92%
Frowning	94%	91%
Surprised	96%	93%
Sunglasses	89%	84%
Masks	87%	81%
Hand Covering	85%	80%
1 Meter Distance	99%	97%
2 Meters Distance	95%	93%
3 Meters Distance	90%	88%
720p Resolution	94%	91%
1080p Resolution	98%	96%

7. Conclusion

The facial recognition attendance management system presented in this research demonstrates the feasibility and effectiveness of using machine learning algorithms for automating the attendance process. The combination of LBPH and Haar Cascade algorithms provides a robust and efficient solution for detection and recognition. The facial recognition attendance management system holds significant potential for future advancements. Enhanced accuracy can be achieved through deep learning models like CNNs to improve recognition accuracy, and 3D recognition to enhance robustness against lighting, angles, and expressions. Privacy and security measures can be enhanced by implementing stronger encryption methods to secure facial data and developing guidelines to ensure ethical use and privacy protection. Integration and scalability can be improved by integrating with IoT devices for seamless operation in smart environments and utilizing cloud resources to handle large-scale data and provide real-time processing. Additionally, the user experience can be enhanced by combining face recognition with other biometric methods like voice or fingerprint for improved reliability, and providing real-time analytics and insights to enhance decision-making processes.

References

[1] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." In Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001 Kauai, HI, USA, vol. 1, pp. I-I. Ieee, 2001.

[2] Ahonen, Timo, Abdenour Hadid, and Matti Pietikainen. "Face description with local binary patterns: Application to face recognition." IEEE transactions on pattern analysis and machine intelligence 28, no. 12 (2006): 2037-2041.

[3] Brunelli, Roberto, and Tomaso Poggio. "Face recognition: Features versus templates." IEEE transactions on pattern analysis and machine intelligence 15, no. 10 (1993): 1042-1052.

[4] Turk, Matthew, and Alex Pentland. "Eigenfaces for recognition." Journal of cognitive neuroscience 3, no. 1 (1991): 71-86.

[5] Belhumeur, Peter N., Joao P. Hespanha, and David J. Kriegman. "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection." IEEE Transactions on pattern analysis and machine intelligence 19, no. 7 (1997): 711-720.

[6] Taigman, Yaniv, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. "Deepface: Closing the gap to human-level performance in face verification." In Proceedings of the IEEE conference on computer vision and pattern recognition, 2014. pp. 1701-1708.

[7] Bradski, Gary, and Adrian Kaehler. Learning OpenCV: Computer vision with the OpenCV library. " O'Reilly Media, Inc.", 2008.

[8] Grayson, John E. Python and Tkinter programming. Manning Publications Co., 2000.

[9] Clark, Alex. "Pillow (pil fork) documentation." readthedocs (2015).

[10] Krogh, Jesper Wisborg, Gennick Krogh, and Gennick. MySQL Connector/Python Revealed. New York: Apress, 2018.

[11] Harris, Charles R., K. Jarrod Millman, Stéfan J. Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser et al. "Array programming with NumPy." Nature 585, no. 7825 (2020): 357-362.

[12] csv — CSV File Reading and Writing (2024). Python 3.11.4 documentation. Available online: https://docs.python.org/3/library/csv.html

[13] Python Software Foundation. (2023). Python 3.11.4 documentation. https://www.python.org/downloads/release/python-3114/

[14] Soniya, V., R. Swetha Sri, K. Swetha Titty, R. Ramakrishnan, and S. Sivakumar. "Attendance automation using face recognition biometric authentication." In 2017 International conference on power and embedded drive control (ICPEDC) Chennai, India, IEEE, 2017pp. 122-127.

[15] Harikrishnan, J., Arya Sudarsan, Aravind Sadashiv, and Remya AS Ajai. "Vision-face recognition attendance monitoring system for surveillance using deep learning technology and computer vision." In 2019 international conference on vision towards emerging trends in communication and networking (ViTECoN), Vellore, India, IEEE, 2019. pp. 1-5.

[16] Bhagat, Shrey, Vithal Kashkari, Shubhangi Srivastava, and Ashutosh Sharma. "Face recognition attendance system." Int. J. Res. Appl. Sci. Eng. Technol 10, no. 1 (2022): 280-283.

[17] Krishnan, Mathana Gopala, and Shyam Babu Balaji. "Implementation of automated attendance system using face recognition." International Journal of Scientific & Engineering Research 6, no. 3 (2015): 30-33.

Author's Biography

Ms. Nuety Beri is currently pursuing her Bachelor of Computer Applications from Maharaja Surajmal Institute, New Delhi (batch 2022-2025). With a keen interest in artificial intelligence, machine learning, python development, software development, and cloud computing, she is dedicated to advancing her knowledge and expertise in these cutting-edge fields.

Mr. Vishal Srivastava is presently pursuing Bachelor of Computer Applications from Maharaja Surajmal Institute, New Delhi (batch 2022-2025) with strong passion for Data Science and Python Programming he excels in analyzing data and developing efficient, scalable solutions. His enthusiasm for data science drives him to explore various machine learning techniques and statistical methods, aiming to extract meaningful insights from complex datasets. Proficient in Python, Vishal enjoys creating data-driven applications and automating tasks to improve productivity.

Ms. Nikita Malik is presently working as an Assistant Professor in Department of Computer Applications at Maharaja Surajmal Institute, New Delhi since August 2019. She is also pursuing her Ph.D in IT from University School of Information, Communication and Technology, GGSIPU. She has 5 years of teaching experience and has published more than 40 research papers in various international and national journals and conferences of repute. Nikita has also co-authored and edited a total of four books and has one patent to her credit. Her research interests include Semantic Web Technologies, IoT, WSN, MANETs, Data Analytics and Information Security aspects.