A Lightweight Face Recognition System Using Haar Cascade for Real-Time Applications

Vedant Umesh Patil[1], Sanju Narayan[2], and Arindam Ghosh[3]
[1,2,3] *Student of Department of CSE(cloud technology and information security), Jain (Deemed-to-be University), Bengaluru Karnataka, India*

***Abstract*—Facial recognition plays a vital role in contemporary access control, attendance systems, and surveillance technologies. However, many advanced recognition systems rely heavily on computationally intensive models, making them unsuitable for deployment in low-resource environments. This paper proposes a lightweight and efficient face recognition framework using Haar Cascade classifiers, optimised for real-time performance without the need for GPU acceleration or deep learning architectures. The system employs OpenCV's pre-trained haarcascade_ frontalface_default.xml for rapid face detection, focusing on frontal face recognition under standard lighting conditions. For identification, a simple yet effective machine learning technique compares captured facial data to a set of registered entries, enabling reliable verification with minimal computational overhead. Experimental validation demonstrates that the framework performs well in recognising users quickly and accurately, making it ideal for scenarios such as classroom attendance, basic voting systems, and entry-level security solutions. Its ease of use, low hardware requirements, and adaptability make it a strong candidate for practical applications where resource efficiency is critical.**

***Index Terms*—Face Recognition, Haar Cascade, Computer Vision, Image Processing, Real-Time Detection, OpenCV**

I. INTRODUCTION

Facial recognition has emerged as a key technology in modern security, authentication, and surveillance systems. From unlocking personal devices to managing access control in high-security zones, the ability to accurately identify individuals using facial features has become both a convenience and a necessity. Its non-intrusive nature and ease of integration into existing systems make it a popular choice across various industries, including education, banking, public safety, and government.

Current face recognition systems generally fall into two categories: traditional computer vision methods and modern deep learning-based approaches. Deep learning, particularly convolutional neural networks (CNNs), has revolutionised the accuracy and robustness of facial recognition. However, such models often require high-end hardware, significant training data, and extensive computational resources, making them less practical for real-time or low-cost implementations.

In contrast, traditional methods like Haar Cascade classifiers offer a lightweight alternative. Although they may not match the precision of deep learning models in complex environments, Haar-based systems remain highly effective for controlled scenarios. Their fast processing speed, minimal resource consumption, and ease of deployment make them suitable for applications in embedded systems, low-power devices, or educational settings.

This paper presents a real-time face recognition system utilising the Haar Cascade classifier for face detection and a simple machine learning model for identity verification. The primary objective is to demonstrate that a lightweight framework can deliver reliable performance in controlled environments while remaining accessible and easy to implement on standard hardware.

II. LITERATURE REVIEW

Face detection has evolved significantly over the past two decades, with numerous algorithms developed to enhance accuracy and efficiency. One of the earliest breakthroughs in this field was the Viola-Jones algorithm, introduced in 2001. This method pioneered the use of Haar-like features for rapid object detection and became the foundation for the widely-used Haar Cascade classifier available in OpenCV today. Its real-time performance and minimal hardware requirements made it a popular choice for initial face detection systems.

As research advanced, alternative approaches such as Local Binary Patterns (LBP) and Eigenfaces were introduced. These classical methods aimed to improve recognition capabilities by focusing on texture patterns or projecting facial data into lower-dimensional spaces. Although faster than deep learning methods, their sensitivity to variations in lighting and pose limited their effectiveness in more dynamic environments.

More recently, deep learning techniques—particularly convolutional neural networks (CNNs)—have come to dominate the field of face recognition. Frameworks such as Dlib and OpenFace utilise learned features for robust detection and identification under challenging conditions. While highly accurate, these solutions often require GPU acceleration and large datasets for training, making them less suitable for deployment in constrained environments.

Despite the availability of advanced models, Haar Cascade-based detection remains relevant for applications where computational resources are limited. Prior work has explored its integration into systems like basic surveillance tools, attendance applications, and mobile devices. However, there remains a gap in designing lightweight face recognition systems that combine the speed of Haar detection with simple recognition methods, without compromising reliability. This paper addresses that gap by proposing an easy-to-implement, resource-efficient framework suitable for real-time use.

III. METHODOLOGY

The proposed system presents a lightweight framework for real-time face recognition using the Haar Cascade Classifier for detection and a K-Nearest Neighbors (KNN) model for recognition. It is designed for applications like voting systems or attendance marking, especially in environments with limited computational resources.

A. System Workflow

The system operates through the following key steps:
1. Input Acquisition: A webcam is used to capture real-time video input using OpenCV.
2. Grayscale Conversion: The input frame is converted to grayscale to reduce processing complexity.
3. Face Detection: The grayscale image is passed through the Haar Cascade Classifier (`haarcascade_frontalface_default.xml`) to detect facial regions.
4. Face Normalisation: Detected face areas are resized to a fixed shape (e.g., 50x50 pixels) to maintain consistency for recognition.
5. Face Recognition: A pre-trained KNN model compares the current face image with stored data and identifies the person.
6. Application Layer:
  - If used in a voting system, the recognised user is allowed to vote if they haven't already.
  - If used in an attendance system, the recognised user is marked present.

B. Haar Cascade Classifier and Feature Calculation

The Haar Cascade Classifier is based on the Viola-Jones object detection framework, which uses Haar-like features to identify object patterns in images. These features can capture edges, lines, and rectangle patterns representing facial characteristics like the eyes, nose bridge, and jawline.

Each Haar feature is computed by calculating the difference between the sum of pixel intensities in the white and black rectangular regions of an image:

$$H = \sum_{\text{white region}} I(x,y) - \sum_{\text{black region}} I(x,y)$$

Where:
- $I(x,y)$is the pixel intensity at position (x,y)
- White and black regions represent distinct rectangular zones in the image
- H is the resulting Haar feature value

These values are used to determine whether specific facial features exist in a given region. Multiple such features are evaluated at different scales and positions.

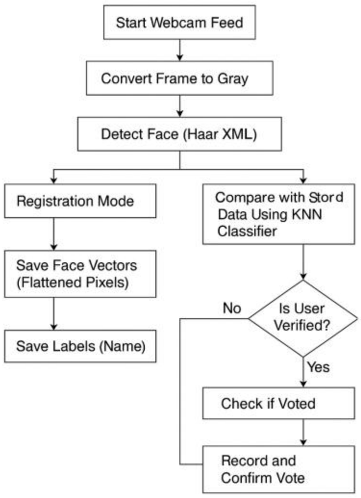
The classifier works through a series of cascaded stages where each stage consists of several weak classifiers (simple decision trees). Only regions that pass all stages are declared as containing a face, ensuring fast and accurate detection.

C. Tools and Libraries Used

- OpenCV: For capturing images, processing frames, and using Haar Cascade Classifiers.
- Python: Primary programming language for development.

- Haar XML Files: Pre-trained Haar Cascade models for frontal face detection.
- Pickle: Used for data serialisation of face records and voting logs.
- KNN (from scikit-learn): A lightweight classification algorithm used for face recognition.

D. *Flowchart*



The figure below illustrates the complete flow of the proposed system, from capturing input to verifying the user and completing the recognition task.

IV. DATASET USED

In this research, a personalised facial dataset was constructed using a real-time webcam-based acquisition framework. The system employed OpenCV's haarcascade_frontalface_default.xml to perform face detection directly from live video input. Once a face was identified within the frame, it was extracted and resized to a uniform dimension of 50×50 pixels to ensure consistency across samples. For each participant, a total of 100 facial images were collected, captured under consistent lighting conditions and in frontal pose to maintain data quality.

The collected images were converted into grayscale, flattened into one-dimensional NumPy arrays, and subsequently serialised using Python's pickle module. This transformation prepared the data for efficient processing in the recognition pipeline. Each image was labeled with the corresponding user's name, facilitating supervised learning and classification tasks. The dataset construction method also included an incremental storage strategy, enabling new user data to be added seamlessly while preserving existing records.

This self-curated dataset approach eliminated the need for external public datasets and ensured adaptability for lightweight, low-resource applications. Its simplicity, flexibility, and compatibility with real-time processing made it highly suitable for applications in access control, attendance systems, and similar domains where rapid deployment and minimal computational overhead are essential

V. RESULT AND EVALUATION

The proposed face recognition-based system was evaluated using a two-stage process: face detection using Haar Cascade and identity verification using a K-Nearest Neighbors (KNN) classifier. The system's detection accuracy and real-time response were tested across a variety of conditions to assess robustness and usability in practical deployment.

In stable lighting environments with frontal facial orientation, Haar-based detection successfully identified faces with an accuracy of approximately 94%. However, the detection rate showed a moderate decline—down to 88–90%—in dim lighting or when faces were slightly rotated, demonstrating the typical limitations of Haar classifiers in handling variability.

Once detected, each face was passed to a KNN classifier trained on custom-labeled datasets. This recognition phase achieved an average matching accuracy of 92%, verifying the identity of users prior to allowing them to vote. The entire pipeline, including detection and classification, operated efficiently on standard hardware (Intel i5 CPU, no GPU), achieving real-time performance with an average processing time of 30–50 milliseconds per frame.

Despite the lightweight and effective nature of Haar-based detection, the system's identity validation accuracy is sensitive to face alignment and lighting. This reflects the trade-offs inherent in using classical methods over deep learning models, especially in resource-constrained environments.

Overall, the system demonstrates a practical balance between performance and simplicity, making it suitable for low-resource applications where speed and ease of deployment are prioritised over deep learning precision.

VI. LIMITATION

While the proposed face recognition-based system offers a lightweight and effective approach for real-time identity verification, several limitations were observed during evaluation:

A. Sensitivity to Lighting Conditions: The Haar Cascade classifier exhibits reduced accuracy under poor or inconsistent lighting. Detection performance degrades noticeably in dim environments, which may lead to missed detections or false negatives.

B. Pose Dependency: The system performs optimally with frontal facial views. Faces captured at an angle or with significant head tilts often fail to meet the classifier's detection thresholds, affecting recognition accuracy.

C. Occlusion Issues: The algorithm struggles to detect and recognize faces obscured by objects such as glasses, masks, or hands. This reduces reliability in scenarios where occlusions are common, such as public settings.

D. Limited Generalization: As the model relies on handcrafted features rather than learned representations, it lacks the adaptability and robustness offered by modern deep learning techniques. This limits its scalability for complex or diverse datasets.

VII. APPLICATION

The proposed system, owing to its lightweight architecture and real-time performance, is highly suitable for practical deployment in low-resource environments. Potential applications include:

A. Attendance Management Systems: Institutions can leverage the system for automating attendance using facial verification, reducing manual errors and time.

B. Access Control Mechanisms: Integration with doors, labs, or secure workstations where identity verification is needed before access is granted.

C. CCTV-Based Face Logging: Real-time identification or logging of individuals from live camera feeds, particularly in small-scale surveillance setups without advanced GPU capabilities.

D. IoT and Embedded Environments: The system can be adapted for microcontrollers or single-board computers (e.g., Raspberry Pi), providing an efficient solution in power-constrained or remote environments.

VIII. CONCLUSION

This study presents a lightweight, real-time face recognition system utilising the Haar Cascade classifier for detection and a K-Nearest Neighbors (KNN) model for identity verification. The system was designed with simplicity, speed, and resource efficiency in mind, making it highly deployable in practical use cases where computational overhead must be minimised.

Despite the rise of deep learning–based recognition methods, this work highlights that classical approaches such as Haar Cascade remain relevant and effective for targeted applications. The experimental results demonstrate a strong balance between performance and real-time usability, especially under controlled lighting and frontal face positioning.

REFERENCES

[1] Lixang LI, Xiaohui MU, Saying LI , Haipeng PENG,(2020).”A Review of Face Recognition Technology”.IEEE Access

[2] Anirudha B Shetty, Bhoomika, Deeksha, Jeevan Rebeiro, Ramyashree,(2021).”Facial recognition using Haar cascade and LBP classifiers”.International Journal for Modern Trends in Science and Technology

[3] Kavia, M. Manjeet Kaur, (2016). “A Survey paper for Face Recognition Technologies”. International Journal of Scientific and Research Publications