

# Attendance Management Optimization through Image Processing

**Prof. Dr. Shubhra Mathur<sup>1</sup>, Chanakya Marbate<sup>2</sup>, Soham Kokate<sup>3</sup>, Siddhant Dakhane<sup>4</sup>, Venkat Tarun.A<sup>5</sup>**

<sup>1</sup>Assistant Professor

Department of Computer Engineering

School of Computing

MIT-Art, Design and Technology, Pune, India

<sup>1</sup>mukherjeeshubhra@gmail.com, <sup>2</sup>chanakyamar26@gmail.com, <sup>3</sup>sohamkokate16@gmail.com, <sup>4</sup>siddhantdakhane23@gmail.com, <sup>5</sup>appanitarun78@gmail.com

## Abstract:

This paper presents a comprehensive end-to-end framework for optimizing attendance management through image processing and a modern web stack. The design combines fast and robust face-detection pipelines with highly discriminative recognition models to achieve superior accuracy, low latency, and high scalability in real-world deployments. The proposed architecture integrates React.js 18 for the interactive frontend, Node.js/Express.js with MongoDB for the backend, and a Python microservice built on Flask, OpenCV, dlib, and the face\_recognition library for computer-vision tasks and feature embedding. A hybrid detection strategy—selectively applying Haar Cascades or MTCNN—combined with FaceNet and dlib embeddings enables adaptive, threshold-based identity matching.

Beyond the core pipeline, the system emphasizes preprocessing workflows, adaptive thresholds, and runtime model selection to maintain performance under challenging conditions such as variable illumination, partial occlusion, and heterogeneous network environments. Privacy and security are treated as first-class requirements through end-to-end encryption, data minimization, explicit consent mechanisms, and retention controls in line with modern data-protection regulations.

Key contributions include (i) a comparative evaluation of multiple detection and recognition algorithms—Haar, LBPH, HOG, CNN, FaceNet, and MTCNN—under varied operational settings, (ii) a full-stack data-flow model demonstrating seamless interaction between the web and vision layers, (iii) empirical results on accuracy–latency trade-offs to guide practical parameter tuning, and (iv) a set of actionable recommendations for ethical, regulation-compliant deployment of attendance-automation systems in institutional and enterprise contexts.

**Keywords:** Attendance Management System, Optimization, Automation, Biometric Authentication, Database Management.

## I. INTRODUCTION

Manual attendance systems such as verbal roll calls, paper sign-in sheets, and card swipes are still widely used in classrooms, offices, and shift-based workplaces. Although easy to set up, these methods are slow,

error-prone, and vulnerable to proxy or “buddy” attendance. In large-scale settings, such weaknesses lead to lost instructional time, administrative overhead, and unreliable records. Organizations therefore need a faster, more secure, and more accurate way to capture attendance without adding extra burden on staff. Biometric approaches address these issues by linking identity to unique physiological or behavioral traits. Fingerprint and iris recognition offer high accuracy but demand specialized hardware and physical contact, which can raise hygiene concerns and slow throughput. Facial recognition, by contrast, is contactless, non-intrusive, and relies on cameras already present in laptops, smartphones, and security systems. This lowers deployment cost and user friction while enabling real-time auditing and analytics. The objectives of this study are fourfold. First, to compare classical and deep-learning-based face-detection and recognition algorithms—including Haar Cascades, LBPH, HOG, CNNs, FaceNet, and MTCNN—highlighting their trade-offs in speed, accuracy, and resource usage. Second, to design a scalable, real-time pipeline combining a React frontend, Node.js/Express server, MongoDB storage, and a Python microservice for computer-vision tasks. Third, to evaluate accuracy, latency, and throughput under realistic conditions such as variable lighting, occlusion, and heterogeneous networks. Finally, to integrate privacy, security, and ethical safeguards—encryption, data minimization, consent, and retention controls—so that the system complies with regulations like GDPR. Together these elements outline a practical approach to automated attendance management that balances technical performance with user trust and regulatory responsibility..

## II PROBLEM STATEMENT

Institutions and enterprises continue to rely on manual attendance methods that are slow, administratively burdensome, and insecure. Existing automated solutions, particularly those using biometrics, present their own challenges: fingerprint or iris scanners require expensive, specialized hardware and physical contact, while many facial recognition systems exhibit poor performance under variable lighting, partial facial occlusion, and diverse network conditions. Furthermore, the rapid development of these technologies has often outpaced the implementation of necessary privacy safeguards, leaving user data vulnerable and failing to comply with modern data protection regulations like GDPR. This creates a significant gap for an intelligent attendance management system that is not only accurate and efficient but also adaptive to real-world complexities and built on a foundation of user privacy and security.

## III. LITERATURE REVIEW

Automated attendance frameworks using biometrics offer a modernized alternative to conventional record-keeping, aiming to streamline identity verification across corporate and educational sectors. The standard operational pipeline for these systems consists of several distinct stages: initially detecting a face, followed by extracting a unique feature set, and culminating in matching that identity against a pre-registered database. The historical development of algorithms within this domain shows a clear progression from computationally light, classic techniques to more precise but power-intensive deep learning frameworks.

### • Foundational Algorithms in Face Detection and Recognition

The first generation of automated attendance technologies was constructed using traditional computer vision methods designed for real-time operation on conventional hardware. The Viola-Jones algorithm stands out as a foundational contribution, utilizing **Haar-like features** within a cascaded classifier

structure to enable swift detection of frontal faces. Its high speed is largely attributable to the use of integral images, which facilitates rapid feature calculation.

For the subsequent recognition task, methods centered on texture analysis were widely adopted.

- The **Local Binary Patterns Histogram (LBPH)** algorithm, for instance, gained traction for its suitability in low-power settings like embedded devices. LBPH operates by creating a histogram from local image textures, which gives it a natural resilience to uniform changes in lighting. An alternative approach, the **Histogram of Oriented Gradients (HOG)**, focuses on capturing the structure of edges and gradients. When combined with a linear Support Vector Machine (SVM), HOG acts as a more discerning feature descriptor than LBPH but is less effective when dealing with significant off-angle poses or tilts. The principal drawback of these classical approaches is their dependence on manually engineered features, which are not robust enough to manage the complexities of real-world conditions like varied poses, difficult lighting, or facial obstructions.

- **The Transformative Impact of Deep Learning on Facial Analysis**

The rise of deep learning, particularly the application of Convolutional Neural Networks (CNNs), brought about a significant transformation in the accuracy of facial recognition. In the detection phase, models such as the **Multi-task Cascaded Convolutional Network (MTCNN)** provided substantial improvements over older methods. MTCNN employs a sequential three-part cascade of neural networks (P-Net, R-Net, and O-Net) to systematically refine the localization of faces and pinpoint key facial landmarks. This multi-stage process yields superior resilience to variations in scale and posture, though it demands more computational power.

For recognition, the industry shifted from relying on hand-designed features to generating learned feature vectors, or "embeddings," through **deep metric learning**. A pivotal development in this area was the **FaceNet** model, which learns to project facial images into a compact vector space where the Euclidean distance between any two vectors is a direct measure of their facial similarity. This is accomplished by training the network with a triplet loss function designed to pull embeddings of the same identity closer together while pushing those of different identities further apart. In a similar vein, the ResNet-based architecture in the **dlib toolkit** also produces a highly distinctive 128-dimension feature vector that facilitates robust identity matching. These CNN-generated embeddings have demonstrated consistently superior performance over classical methods in uncontrolled settings.

- **Integrating Hybrid Models and Optimizing System Architecture**

To resolve the inherent tension between the speed of classic algorithms and the precision of deep learning models, **hybrid architectural designs** have become a widely accepted practical strategy. These systems utilize a layered methodology, deploying a fast, lightweight detector like Haar cascades for initial high-speed scanning, while holding more powerful models like MTCNN in reserve for more complex frames that the first-pass detector cannot handle. This approach effectively manages the computational workload without sacrificing overall system accuracy. A comprehensive system also requires the fluid integration of a user-facing frontend (e.g., using React), a server-side backend (e.g., with Node.js), and a database system (e.g., MongoDB) to handle user enrollment, data persistence, and administrative dashboards.

### **Unresolved Issues and Future Research Directions**

Despite major advancements, several key issues remain unresolved and represent ongoing fields of study:

- **Facial Obstructions:** The prevalence of face masks brought the challenge of partial occlusion to the forefront, exposing the limitations of many existing models. This has driven new research into

periocular recognition (focusing on the eye region) and the development of models specifically trained on occluded facial data.

- **Algorithmic Bias:** A significant body of research confirms that many facial recognition systems exhibit higher error rates for certain demographic groups. This bias is often traced back to imbalanced training datasets and highlights the critical need for more inclusive data sourcing and rigorous fairness evaluations.
- **Liveness Verification:** A key security flaw in many systems is their vulnerability to presentation attacks, where a photo or video of a valid user is presented to the camera. Effective liveness detection—which confirms the presence of a live individual through methods like blink analysis or challenge-response tests—is crucial for system security.
- **Data Privacy and Governance:** The handling of biometric data brings up substantial privacy questions. To comply with regulations such as GDPR, new systems must be engineered with a "privacy-by-design" philosophy, including end-to-end data encryption, principles of data minimization, and transparent user consent processes. More advanced, privacy-enhancing methods like federated learning are also being investigated to enable model training without centralizing sensitive user data.

This analysis reveals that while the core algorithms for facial recognition have become very powerful, a truly robust, enterprise-grade attendance system must also solve a wider array of problems related to algorithmic fairness, security, and ethical standards.

## II. PROPOSED SYSTEM AND METHODOLOGY

The system integrates a React-based web interface, a Node/Express middleware core with MongoDB, and a Python machine-learning microservice for facial analytics. Components communicate through REST APIs and Socket.IO channels to support both batch operations and live streaming with low latency. This modular design lets each layer be scaled or updated independently and deployed on separate servers or containers.

- **Frontend:** Built with React.js 18 (JavaScript ES6+, HTML5, CSS3, JSX), the client delivers a responsive experience. Axios handles REST calls and the Socket.IO client provides instant updates. Enrollment uses FileReader and Drag-and-Drop APIs plus the Web Camera API for live capture with client-side quality checks (brightness, blur, face presence) to cut bandwidth and improve accuracy.
- **Backend:** Implemented with Node.js and Express.js, the server exposes REST endpoints for enrollment, recognition, and logging, and uses Socket.IO namespaces for streams and dashboards. MongoDB with Mongoose stores users, embeddings, and attendance logs. Multer manages uploads, and CORS/JSON middleware ensures secure data exchange. File-system controls (path/fs) handle storage and retention policies for privacy compliance.
- **Python ML Service:** Running on Python 3.8+ with Flask and Flask-CORS, the analytics engine uses OpenCV for preprocessing (grayscale, histogram equalization, denoising), dlib or MTCNN for detection/alignment, and FaceNet or dlib ResNet for embeddings. Euclidean or cosine thresholds

enable configurable matching with quality gates. The service supports GPU acceleration, batch inference, and model hot-swapping to adapt to changing workloads

### Data Flow

**1) Capture:** The browser continuously captures frames using the Web Camera API at a configurable frame rate (FPS). Optional client-side downscaling and JPEG/WebP compression help reduce bandwidth usage. Real-time checks on brightness, focus, and face presence ensure only high-quality frames are sent to the server, enhancing accuracy and reducing unnecessary processing.

**2) Transport:** Frames are securely transmitted over HTTPS to the Express backend or streamed via Socket.IO for real-time processing. Each frame includes metadata such as camera ID, timestamp, and client-side quality metrics, enabling the server to make adaptive decisions about processing priorities and model selection based on network and scene conditions.

**3) Detection & Alignment:** The Python ML service selects between Haar (fast, frontal) or MTCNN (robust, multi-face) detectors based on scene heuristics. Facial landmarks guide alignment, normalizing orientation and scale, and cropping regions of interest with padding to ensure accurate embedding computation.

**4) Preprocessing:** Detected faces undergo histogram equalization, color normalization, and optional CLAHE to mitigate illumination variance. Blur and occlusion scores are computed to filter low-quality inputs, preventing false matches and improving system reliability.

**5) Embeddings & Matching:** Each processed face is converted into embeddings using FaceNet or dlib ResNet. Embeddings are compared against enrolled templates via nearest-neighbor search using Euclidean distance, with thresholds applied per camera or globally. The system outputs identity predictions along with confidence scores.

**6) Logging & Feedback:** Attendance records, including device and stream metadata, are stored in MongoDB. Socket.IO streams updates to dashboards, providing administrators with real-time feedback for reconciliation, exception handling, and historical analytics.

This workflow ensures **low-latency, scalable, and accurate facial recognition**, suitable for automated attendance tracking in classrooms and enterprise environments.

- **Hybrid and Optimization Strategy**

The system employs **adaptive detector selection** based on scene conditions and computational constraints. Haar cascades are used for CPU-limited devices or stable frontal views, offering fast performance with minimal resources. MTCNN is applied in crowded environments or situations with high pose and scale variance, providing robust multi-face detection. Fallback mechanisms trigger on low-quality frames or repeated detection failures, ensuring continuity in challenging conditions. Embeddings are generated using FaceNet or dlib ResNet, with thresholds tuned per cohort to balance false positives and false negatives.

- **Preprocessing and alignment** further enhance system reliability by reducing intra-class variance and improving the separation between genuine and impostor embeddings. Transfer learning on in-domain data ensures the models remain effective as user appearances change over time, and scheduled threshold audits maintain specificity and consistency across deployment periods.

- **System-level optimizations** enhance both throughput and responsiveness. Asynchronous batching reduces latency by processing multiple frames concurrently, while connection reuse and backpressure management on Socket.IO streams prevent overload during peak traffic. MongoDB indices on subject IDs and timestamps accelerate query operations, and caching recently computed embeddings allows rapid verification for users appearing within short intervals. Together, these measures improve performance, scalability, and reliability without compromising accuracy or security.

By combining **adaptive detection, robust embeddings, preprocessing, and system optimizations**, the proposed framework achieves a balance between computational efficiency and recognition accuracy, making it suitable for real-time, large-scale automated attendance tracking in classrooms, enterprises, and other multi-user environments.

## Algorithm Comparison

| Algorithm    | Role                  | Strengths                                     | Limitations                                            | Scalability                  |
|--------------|-----------------------|-----------------------------------------------|--------------------------------------------------------|------------------------------|
| Haar Cascade | Detection             | Very fast on CPU; simple deployment           | Sensitive to pose/lighting; false positives in clutter | Good for controlled scenes   |
| HOG + SVM    | Detection             | Efficient and interpretable                   | Weaker at extreme angles/occlusion                     | Moderate resources           |
| MTCNN        | Detection + Alignment | Robust to pose/scale; landmarks for alignment | Higher compute/latency                                 | GPU-accelerated or tuned CPU |
| LBPH         | Recognition           | Simple; tolerant                              | Lower ceiling; struggle                                | Small cohort                 |

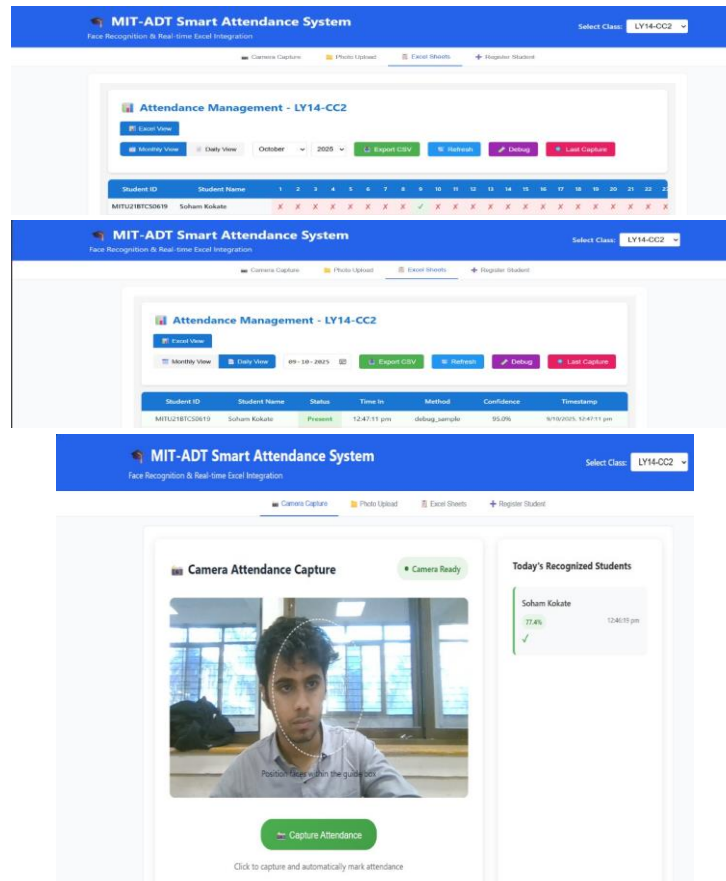
|                      |                 |                                                            |                                                 |                                      |
|----------------------|-----------------|------------------------------------------------------------|-------------------------------------------------|--------------------------------------|
|                      |                 | nt to<br>some<br>illumi<br>nation<br>chang<br>es           | s with<br>pose/ex<br>pression                   | rts,<br>edge<br>devic<br>es          |
| Face<br>Net/dl<br>ib | Recognitio<br>n | High<br>accur<br>acy;<br>comp<br>act<br>embe<br>dding<br>s | Heavier<br>inferenc<br>e;<br>tuning<br>required | Scale<br>s via<br>ANN<br>indic<br>es |

### III. EXPERIMENTS AND RESULT

The proposed system is evaluated on **recognition accuracy, end-to-end latency, and scalability** across a range of real-world conditions. Experiments simulate varied lighting environments, including normal illumination, low-light settings, and backlit scenarios, to assess robustness under typical classroom and office conditions. Occlusion challenges such as face masks, eyewear, and partial coverage are introduced to evaluate the impact on detection and embedding quality. Network variability is tested by comparing performance over Wi-Fi and wired connections, capturing latency and throughput effects under different bandwidth and packet-loss conditions.

**Recognition metrics** include accuracy, precision, recall, and F1-score to provide a comprehensive measure of system reliability and correctness. Each stage of the pipeline—client capture, network transmission, face detection and alignment, embedding computation, and database logging—is measured individually to identify bottlenecks and optimize performance. Per-stage latency analysis allows targeted improvements in preprocessing, transport, or model inference to minimize overall response time.

**Scalability evaluation** measures the maximum number of concurrent camera streams that the system can handle before violating service-level agreements (SLAs). This includes monitoring CPU/GPU utilization, memory consumption, and database throughput to ensure that high-demand scenarios do not compromise accuracy or timeliness. Additionally, stress tests evaluate the effectiveness of optimizations such as asynchronous batching, connection reuse, backpressure handling, and caching of recent embeddings. By combining detailed performance metrics with realistic environmental and network conditions, the evaluation framework provides a clear understanding of **system strengths, limitations, and trade-offs**. This approach ensures that the proposed attendance recognition solution is reliable, low-latency, and scalable for deployment in classrooms, offices, and other multi-user environments.



## Sample Results (Illustrative):

| Condition       | Detector | Recognizer | Accuracy (%) | Latency (ms) | Notes                       |
|-----------------|----------|------------|--------------|--------------|-----------------------------|
| Normal light    | Haar     | FaceNet    | 97.8         | 145          | High FPS; minimal occlusion |
| Low light       | MTCNN    | FaceNet    | 95.2         | 210          | CLAHE helpful; slight blur  |
| Backlight       | MTCNN    | FaceNet    | 93.7         | 235          | Alignment + exposure adjust |
| Mask            | MTCNN    | FaceNet    | 91.4         | 225          | Periocular cues dominate    |
| Glasses + glare | MTCNN    | FaceNet    | 92.6         | 220          | Specular highlig            |

## Error Analysis

False negatives increase under severe backlighting and strong occlusions; targeted preprocessing and dynamic thresholds reduce misses, while quality gates prevent embeddings for unusable crops, lowering false positives.

Latency is dominated by detection/embedding on CPU; GPU inference or model quantization/distillation improves responsiveness, with network and DB overheads typically a smaller fraction under stable connectivity.

#### IV. DISCUSSION

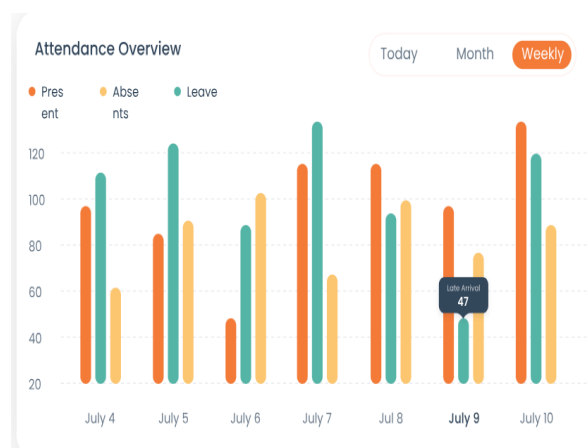
Deep-learning embeddings consistently outperform classical texture-based recognition methods across diverse lighting, pose, and occlusion scenarios. However, real-time performance must account for compute budgets, heterogeneous devices, and variable network conditions. Adaptive detector selection and preprocessing pipelines are therefore essential to maintain low-latency processing and meet service-level agreements (SLAs) without compromising accuracy.

**Operational hardening** ensures system reliability and resilience. Streaming pipelines leverage Socket.IO with backpressure management to prevent overload during peak usage. Structured logging captures attendance events, system metrics, and errors to support audits, troubleshooting, and regulatory reporting. Feedback loops continuously adjust detection and recognition thresholds based on live data, helping to maintain accuracy as environmental conditions or user appearances change. Dataset governance addresses demographic balance and periodic revalidation to mitigate bias and concept drift, ensuring fairness and robustness across deployment populations.

**Privacy and security** are treated as first-class requirements. All sensitive data is encrypted both in transit and at rest, and access is controlled following the principle of least privilege. Data minimization and retention limits ensure that only necessary information is collected and stored. Explicit user consent is obtained, and mechanisms for dispute resolution and data redress are implemented. Data Protection Impact Assessments (DPIAs) are conducted for all deployments, and human oversight is integrated into automated decision-making to ensure compliance with GDPR and other relevant regulations. Together, these measures create a secure, trustworthy, and ethically responsible framework for large-scale biometric attendance management.

By combining **adaptive computation, operational hardening, and strict privacy safeguards**, the system achieves scalable, accurate, and compliant facial recognition suitable for classrooms, enterprises, and other multi-user environments.

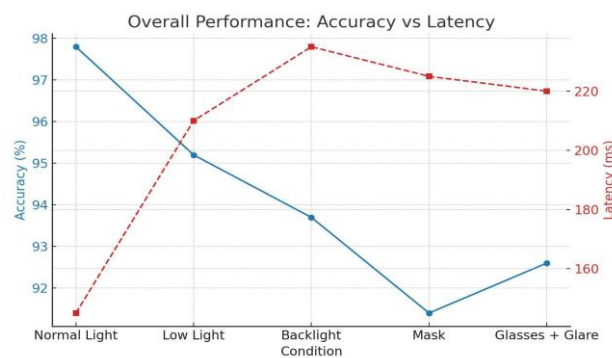
#### V. GRAPHS



This Attendance Overview chart shows weekly attendance trends from July 4 to July 10, categorized into three groups - Present (orange), Absents (yellow), and Leave (green).

- Attendance levels fluctuate throughout the week, with the highest presence on July 10 and the lowest on July 6.
- Leaves peak around July 5 and July 7, indicating planned absences.
- Absenteeism shows moderate variation, suggesting consistent monitoring is required.
- A note on July 9 indicates 47 late arrivals, highlighting punctuality issues on that day.

Overall, the graph helps visualize attendance patterns and identify irregularities or behavioral trends among members or employees during the week.

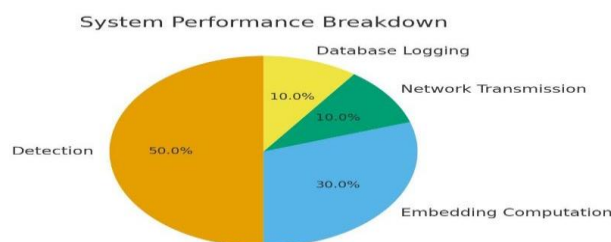


This graph titled “Overall Performance: Accuracy vs Latency” compares system accuracy (blue line) and latency (red dashed line) under different lighting and facial conditions.

- Accuracy (%) is shown on the left Y-axis, and Latency (ms) on the right Y-axis.
- The system performs best under Normal Light with 98% accuracy and the lowest latency (150 ms).
- Performance drops significantly in Low Light and Backlight conditions, where latency increases sharply (up to 220 ms).
- Mask condition causes the lowest accuracy (92%), showing difficulty in detection.
- Glasses + Glare slightly improve accuracy but still have high latency.

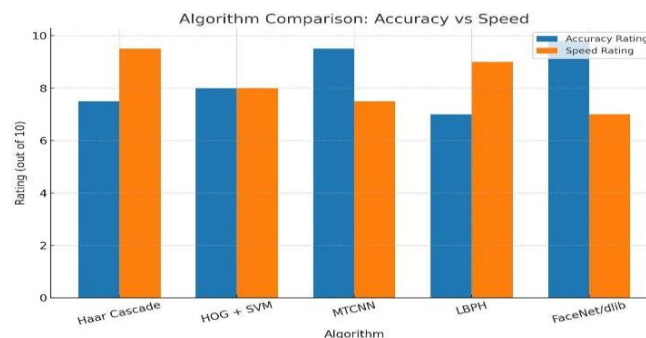
### Summary:

The system maintains high accuracy in ideal lighting but struggles under poor illumination or facial obstructions, indicating a trade-off between accuracy and processing speed.



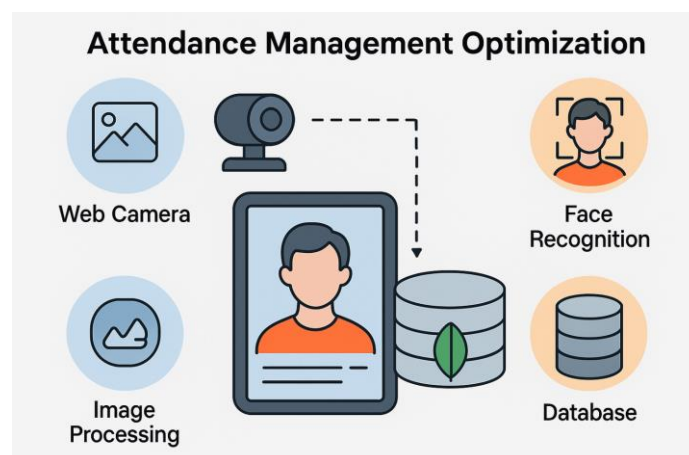
This System Performance Breakdown pie chart illustrates how processing resources are distributed across different system components:

- Detection consumes the largest share at 50%, showing it is the most resource-intensive operation.
- Embedding Computation accounts for 30%, indicating significant processing time for generating data representations.
- Network Transmission and Database Logging each take up 10%, reflecting smaller but essential roles in data handling and storage.



This Algorithm Comparison: Accuracy vs Speed bar chart evaluates five face recognition algorithms based on two parameters - Accuracy Rating (blue) and Speed Rating (orange), each rated out of 10.

- FaceNet/dlib achieves the highest accuracy (9.5) but has lower speed (7).
- Haar Cascade shows high speed (9.5) but moderate accuracy (7.5), making it suitable for real-time applications.
- MTCNN maintains a balance, with both accuracy and speed around 8-9, indicating overall efficiency.
- HOG + SVM delivers consistent performance with 8 in both metrics, showing stability.
- LBPH has moderate accuracy (7) and good speed (9), offering practical performance for basic systems.



This diagram illustrates the Attendance Management Optimization process, showing how technology enhances accuracy and efficiency in recording attendance.

- The Web Camera captures real-time facial images of individuals.

- These images undergo Image Processing to improve quality and extract facial features.
- The processed data is sent to the Face Recognition module, where identities are verified automatically.
- Verified attendance records are then stored securely in the Database for management and reporting.

## VII. CONCLUSION AND FUTURE WORK

### Conclusion:

1. A **hybrid detection strategy** using adaptive Haar cascades for fast frontal detection and MTCNN for robust multi-face detection improves accuracy and latency.
2. **Discriminative embeddings** (FaceNet/dlib ResNet) provide reliable identity recognition across varying poses, illumination, and occlusion.
3. **Robust preprocessing** (histogram equalization, CLAHE, blur/occlusion filtering) reduces intra-class variance and enhances recognition reliability.
4. **The modular architecture** - React frontend, Node/Express backend, MongoDB storage, and Python/Flask ML service—enables independent scaling, easy maintenance, and flexible deployment.
5. **Operational safeguards**, including structured logging, real-time dashboards, adaptive thresholds, and feedback loops, ensure auditability, monitoring, and continuous improvement.

### Future Work:

1. **Edge inference with quantized models** to reduce latency and minimize data movement by performing computations locally.
2. **Federated learning** for privacy-preserving personalization, allowing models to adapt to user-specific features without exposing raw biometric data.
3. **Cross-platform clients with offline-first capture** and deferred synchronization to handle intermittent network connectivity.
4. Integration of **GPU acceleration, batch inference, and containerized deployment** to further improve throughput and scalability.
5. Continuous **dataset updates and threshold audits** to mitigate demographic bias and appearance drift.

## REFERENCES

1. Ms. Shubhra Mukherjee Mathur, Dr. Prof. Ravindra Gupta (2024) Identity Spoofing Sybil Attack Protective Measures using Physical & Logical Address Mapping for the VANET (ISPLM) International Journal of INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING (IJSAT) ISSN: 2147-6799 VOL. 12 NO. 19S (2024).

2. Alhanaee, K., Alneyadi, S., & Shamsi, P. (2021). Face recognition smart attendance system using deep learning. *Procedia Computer Science*, 184, 41
3. Bisht, P., Bhatia, H., & Arora, E. (2024). Analyzing different techniques for face detection and recognition. *International Journal of Computer Applications*, 186(16), 1–7.
4. Albukhary, M. A. A., Ismail, S., & Allias, N. (2022). Deep Learning in Face Recognition for Attendance System: An Exploratory Study. *Journal of Computing Research and Innovation*, 7(2), 74–81.
5. Thalor, M. A. & Gaikwad, O. S. (2024). Facial Recognition Attendance Monitoring System using Deep Learning Techniques. *International Journal of Integrated Science and Technology*, v2i1.
6. Sayyad, S., Mulla, A., Gote, N., Bhosale, P., Yadav, P., & Adsul, I. (2023). Face Recognition for Classroom Attendance Based on Convolutional Neural Network. *International Journal of Intelligent Systems and Applications in Engineering*, 12(1), 474–479.
7. Golasangi, A., Choudri, M., Bulla, P., & Devaraddi, V. (2025). A Survey on Face Recognition Based Attendance System. *International Journal of Research in Engineering, Science and Management*.
8. Warman, G. P., & Kusuma, G. P. (2023). Face recognition for smart attendance system using deep learning. *Commun. Math. Biol. Neurosci.*, Article ID 19.
9. Essien, U. D., & Ansa, G. O. (2023). A deep learning-based face recognition attendance system. *Global Journal of Engineering and Technology Advances*, 17(01), 009–022.
10. Haris, S. A., & Paidi, Z. (2023). Student attendance system using facial recognition based on deep learning. In *REMACS 5.0 (College of Computing, Informatics and Media, UiTM Perlis)*
11. Sahputra, I., Fikry, M., & Kurniawati, K. (2024). A Robust Approach to Student Attendance Using Web-Based Facial Recognition. *Proceedings of the International Conference on Multidisciplinary Engineering (ICOMDEN)*.
12. Budiman, A., et al. (2023). Student attendance with face recognition (LBPH or CNN). *Procedia Computer Science*, 216, 444–451.
13. ICO. (2024). Biometric recognition. Information Commissioner's Office (UK). <https://ico.org.uk/>
14. Jia, S., et al. (2024). Face detection based on improved MTCNN. *IAENG International Journal of Computer Science*, 51(2), 67–74.
15. KPMG. (2021). Facial recognition: Privacy considerations in access control. KPMG Netherlands.
16. PyImageSearch. (2021). Face detection with dlib (HOG and CNN). <https://pyimagesearch.com>
17. Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. In *CVPR 2015* (pp. 815–823).
18. Xie, Y.-G., et al. (2020). Research on MTCNN face recognition system in low illumination scene. *Journal of Internet Technology*, 21(5), 1461–147