IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155271

A Lightweight HSV Histogram-Based Algorithm for Real-Time Face Recognition on Edge Devices

Prof. Roshni Gawande[1], Dr. S. B. Patil[2], Prof. Sneha Dhere[3]

Zeal Education Society, Nahre Pune[1-3]

Abstract: Real-time face recognition systems are increasingly deployed in mobile and edge environments, yet most deep learning approaches demand GPU acceleration and large memory footprints. This paper presents a lightweight algorithm combining Haar Cascade detection, HSV histogram feature extraction, and cosine similarity classification. Implemented via a Flutter mobile client and Flask REST API backend, the system achieves an average end-to-end latency of 82.5 ms and a False Acceptance Rate (FAR) of 0.25% without GPU support. Experimental evaluation demonstrates that algorithmic efficiency and minimal infrastructure overhead can outweigh marginal accuracy gains of deep learning models in controlled access-control scenarios.

1. INTRODUCTION

Biometric authentication has become a cornerstone of modern security systems. Facial recognition, in particular, offers non-intrusive verification with high user acceptance. While deep learning models such as FaceNet and ArcFace achieve near-perfect accuracy, their reliance on GPUs and large-scale infrastructure limits deployment in resource-constrained environments. This work addresses the need for a lightweight, CPU-only solution optimized for small-scale, real-time applications.

2. RELATED WORK

Early approaches such as Eigenfaces and Fisherfaces relied on dimensionality reduction techniques. Local Binary Patterns Histograms (LBPH) improved robustness but remained sensitive to illumination. Deep learning methods like FaceNet and ArcFace revolutionized accuracy but introduced heavy computational demands. The Viola-Jones Haar Cascade framework remains relevant for CPU-based detection due to its speed and efficiency.

3. PROPOSED METHODOLOGY

The system consists of three sequential stages:

- **Face Detection**: Haar Cascade applied to grayscale images with optimized parameters.
- **Feature Extraction**: HSV histograms (50 bins per channel) form a 150-dimensional normalized vector.
- **Classification**: Cosine similarity reduces to a dot product due to pre-normalization, with a threshold of 0.40 for acceptance.

4. SYSTEM ARCHITECTURE

- **Mobile Client**: Flutter application for image capture and Base64 encoding.
- **Backend**: Flask REST API for detection, extraction, and classification.
- **Data Transport**: JSON payloads ensure lightweight communication.
- **Storage**: Dual-layer persistence with RAM for vectors and filesystem for raw images.

5. EXPERIMENTAL SETUP

- **Hardware**: Intel Core i7 server, Samsung Galaxy A53 client.
- **Dataset**: 25 subjects, 10 training and 5 testing images each.
- **Conditions**: Controlled indoor lighting, frontal poses.

6. RESULTS AND DISCUSSION

- **Recognition Metrics**: Precision = 89.09%, Recall = 78.40%, FAR = 0.25%, FRR = 21.60%.
- **Latency**: Haar detection consumes 83.5% of server time; total server-side latency = 34 ms; end-to-end latency = 82.5 ms.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155271

- **Threshold Trade-off**: Lowering threshold to 0.30 improves recall to 92.8% but increases FAR to 1.56%.

7. CONCLUSION

The proposed HSV histogram-based algorithm demonstrates that efficient, CPU-only recognition is feasible for small-scale deployments. While accuracy is lower than deep learning models, the trade-off in speed, memory footprint, and security makes it suitable for edge applications. Future work includes integrating Presentation Attack Detection (PAD), scalability testing, and hybrid approaches combining handcrafted and learned features.

REFERENCES

[1]. M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
[2]. P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE TPAMI*, vol. 19, no. 7, pp. 711–720, 1997.
[3]. T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns: Application to face recognition," *IEEE TPAMI*, vol. 28, no. 12, pp. 2037–2041, 2006.
[4]. F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE CVPR*, 2015, pp. 815–823.
[5]. J. Deng et al., "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE CVPR*, 2019, pp. 4690–4699.