



Similarity-based Identity Verification System

Harsh¹, Tejasva², Santosh Kailash Galve³, Naman Narang⁴, Ms.Hemlata⁵

Department of CSE-AI&DS,

Panipat Institute of Engineering & Technology, Samalkha, India

Mail ID: hemplata.cse.ai-ds@piet.co.in

Abstract—Identity verification is a fundamental task in security-sensitive applications such as authentication, access control, digital forensics, and identity management. This paper presents a similarity-based identity verification system that verifies a claimed identity by comparing extracted biometric or profile features with stored reference representations. The proposed workflow performs data acquisition, preprocessing, feature extraction, similarity computation, threshold-based decision making, and verification reporting. Distance and similarity measures such as Euclidean distance and cosine similarity are considered for measuring the closeness between an input representation and a stored identity profile. The system is designed to remain computationally practical while supporting variations in illumination, pose, expression, image quality, and other acquisition conditions. The paper describes the system architecture, feature-processing pipeline, verification algorithm, evaluation methodology, security considerations, and practical deployment issues. Rather than relying only on a raw similarity score, the framework separates feature extraction from decision policy so that thresholds can be calibrated for different operating environments. The proposed approach provides a transparent foundation for identity verification and can be extended with modern embedding models, liveness detection, multimodal authentication, and privacy-preserving storage.

Keywords—Identity verification, similarity measures, feature extraction, biometric authentication, Euclidean distance, cosine similarity, machine learning, security.

INTRODUCTION

Reliable identity verification is essential in systems where access or a transaction must be associated with the correct individual. Conventional password-based authentication is simple to deploy, but passwords can be guessed, shared, stolen, or reused. Biometric verification provides another approach by using measurable characteristics of a person, while profile-based verification can use structured attributes or learned representations. In both cases, the central problem is to determine whether a newly presented sample is sufficiently similar to a stored reference belonging to the claimed identity.

Similarity-based verification treats the decision as a comparison problem. An input sample is transformed into a feature representation, and that representation is compared with one or more stored representations. The comparison produces a score or distance. A decision threshold then determines whether the claimed identity is accepted or rejected. This separation is useful because feature extraction and decision policy can be improved independently and because the final decision can be related to a measurable similarity value.



The main objective of this work is to design a clear and computationally manageable identity verification pipeline based on feature similarity. The framework considers preprocessing, discriminative feature extraction, normalization, similarity computation, threshold selection, and evaluation. Euclidean distance and cosine similarity are used as representative comparison measures because they are widely applicable to numerical feature vectors and are straightforward to implement.

The contributions of this work are: (1) an end-to-end architecture for similarity-based identity verification; (2) a preprocessing and feature-extraction workflow that produces comparable representations; (3) a comparison layer supporting distance and angular similarity measures; (4) a threshold-based verification mechanism; (5) an evaluation protocol based on false acceptance, false rejection, accuracy, and related operating characteristics; and (6) a discussion of deployment, privacy, and security considerations.

RELATED WORK

Biometric authentication has been studied extensively using fingerprints, face images, iris patterns, voice, and other measurable characteristics. Jain et al. described biometrics as automated recognition based on physiological or behavioral characteristics and emphasized the importance of representation, matching, and decision stages [1]. Face recognition research has subsequently moved from handcrafted descriptors toward learned feature embeddings, where samples from the same identity are encouraged to occupy nearby locations in an embedding space.

Deep convolutional architectures have played an important role in visual representation learning. ResNet introduced residual learning to enable deeper networks while improving optimization [2]. FaceNet proposed learning a compact embedding in which Euclidean distance can directly represent face similarity [3]. ArcFace further improved discriminative face representations through an additive angular margin loss [4]. These approaches demonstrate the usefulness of embedding spaces for identity comparison, although they require substantial training data and careful consideration of privacy and deployment constraints.

Traditional similarity measures remain useful when the feature representation is already informative. Euclidean distance measures the geometric separation between two vectors, whereas cosine similarity measures their angular relationship. For normalized feature vectors, cosine similarity is particularly attractive because it reduces sensitivity to vector magnitude. The choice of metric should nevertheless be validated against the characteristics of the selected feature representation.

The proposed system differs from a model-specific biometric recognizer by emphasizing a modular verification architecture. A feature extractor may be replaced without changing the comparison and decision layers. This makes the framework suitable for academic experimentation with handcrafted features, conventional machine-learning features, or pretrained embedding models. The same architecture can also be adapted to multimodal verification when multiple independent feature sources are available.

PROPOSED SYSTEM

The proposed system contains six principal stages: identity enrollment, input acquisition, preprocessing, feature extraction, similarity computation, and decision making. During enrollment, one or more reference samples are associated with a claimed identity and converted into feature vectors. During verification, a new sample is processed using the same pipeline. The resulting query vector is compared with the stored reference vector or identity template.



The logical flow is: Input Sample → Quality Check → Preprocessing → Feature Extraction → Feature Normalization → Similarity Calculation → Threshold Comparison → Accept/Reject. A quality-control stage is included before feature extraction because severe blur, occlusion, illumination problems, or incomplete samples can create unreliable representations and consequently unstable verification decisions.

The enrollment database should store identity identifiers separately from feature representations whenever possible. A reference identity can contain multiple templates rather than a single sample, allowing the system to model normal within-person variation. During verification, the query can be compared against the templates of the claimed identity and aggregated using an appropriate rule, such as maximum similarity, mean similarity, or a learned score-fusion strategy.

The decision module receives a similarity score S and a threshold τ . For cosine similarity, the claimed identity is accepted when $S \geq \tau$. For Euclidean distance D , the identity is accepted when $D \leq \tau$. The threshold should not be selected arbitrarily; it must be calibrated using development data so that the operational trade-off between false acceptance and false rejection is appropriate for the application.

The quality-control stage is included before feature extraction because severe blur, occlusion, illumination variations, noise, or incomplete samples may produce unreliable representations and unstable verification decisions. The enrollment database should store identity identifiers separately from feature representations whenever possible, and a reference identity may contain multiple templates rather than a single sample to capture normal within-person variation. During verification, the query can be compared with multiple templates belonging to the claimed identity and the resulting scores can be aggregated using maximum similarity, mean similarity, or an appropriate score-fusion strategy. The decision module receives a similarity score (S) and a threshold (τ); for cosine similarity, the claimed identity is accepted when ($S \geq \tau$), whereas for Euclidean distance, acceptance occurs when ($D \leq \tau$). The threshold should be calibrated using representative development data rather than selected arbitrarily, ensuring an appropriate balance between false acceptance and false rejection. In addition, the proposed system can incorporate security and reliability mechanisms such as liveness detection, encrypted template storage, access control, failed-attempt monitoring, and audit logging to improve protection against spoofing and unauthorized access. These mechanisms make the framework suitable for applications such as laboratory and classroom access, examination authentication, office entry, and other digital services where reliable identity verification is required. The modular design also allows individual components, including feature extractors, similarity measures, and decision strategies, to be replaced or improved without redesigning the complete verification architecture.

METHODOLOGY

A. Data Acquisition and Enrollment: The system can operate on biometric images or other structured identity samples. During enrollment, samples are captured under controlled procedures and associated with a verified identity label. Multiple samples per identity are recommended because a single reference may not represent the natural variation encountered during later verification.

B. Preprocessing: Preprocessing aims to reduce irrelevant variation before feature extraction. Depending on the modality, operations may include resizing, alignment, normalization, background handling, noise reduction, and quality assessment. The same preprocessing policy must be applied to both enrollment and query samples. A mismatch between training/enrollment preprocessing and verification preprocessing can reduce the reliability of similarity scores.



C. Feature Extraction: The feature extractor converts an input sample into a numerical vector. A conventional implementation may use statistical, texture, shape, or local descriptors, while a modern implementation may use a pretrained neural embedding model. Let x represent an input sample and $f(\cdot)$ represent the feature extractor. The resulting representation is $z = f(x)$. Normalization can then transform z into a representation suitable for comparison.

D. Similarity Measures: For two vectors x and y , Euclidean distance is $D(x,y) = \sqrt{\sum_i (x_i - y_i)^2}$. Smaller values indicate greater similarity. Cosine similarity is $S(x,y) = (x \cdot y) / (\|x\| \|y\|)$. Larger values indicate greater angular similarity. The measures should be evaluated empirically because the score distributions depend on the feature space.

E. Threshold Calibration: Verification produces two classes of comparisons: genuine comparisons, where the query and reference belong to the same identity, and impostor comparisons, where they belong to different identities. The genuine and impostor score distributions are used to select an operating threshold. Important metrics include false acceptance rate (FAR), false rejection rate (FRR), equal error rate (EER), verification accuracy, and receiver operating characteristic (ROC) behavior.

F. Verification Algorithm: The query is first validated for quality, transformed into a feature vector, normalized, and compared against the claimed identity template. The resulting score is compared with the calibrated threshold. If the score satisfies the acceptance rule, verification succeeds; otherwise, the system rejects the claim. Logging should record the decision and relevant non-sensitive metadata while avoiding unnecessary storage of raw biometric samples.

$$D(x,y) = \sqrt{\sum_i (x_i - y_i)^2}$$

$$S(x,y) = (x \cdot y) / (\|x\| \|y\|)$$

SYSTEM ARCHITECTURE

The complete architecture is organized into an enrollment path and a verification path. The enrollment path receives verified samples, applies the same preprocessing and feature extraction pipeline used at verification time, and stores protected feature templates. The verification path receives a claimed identity and a query sample, produces a query embedding, retrieves the claimed template, computes a similarity score, and applies the calibrated decision threshold.

The conceptual processing pipeline is: Input Sample $\rightarrow$ Quality Check $\rightarrow$ Preprocessing $\rightarrow$ Feature Extraction $\rightarrow$ Feature Normalization $\rightarrow$ Similarity Calculation $\rightarrow$ Threshold Comparison $\rightarrow$ Accept / Reject. In a practical implementation, audit logging, access control, and secure template storage surround the core pipeline rather than forming part of the similarity computation itself.

A modular architecture allows the feature extractor to be replaced without redesigning the comparison layer. This is useful when experimenting with conventional descriptors, pretrained embeddings, or domain-specific representations. The same architecture can support single-modal and multimodal verification by combining similarity scores from multiple sources.

EXPERIMENTAL DESIGN

The evaluation should be designed around verification rather than simple classification. For every query, the system is given a claimed identity and must determine whether the query belongs to that identity. Genuine trials compare two samples from the same identity, whereas impostor trials compare a sample with a reference from a different identity. The final test set should remain isolated from threshold tuning.



A confusion matrix can be constructed from accepted and rejected genuine and impostor trials. ROC and precision-recall analyses can be used when a range of operating thresholds is important. When security requirements are strict, the evaluation should emphasize low-FAR operating points rather than overall accuracy alone. The dataset should contain enough within-identity variation to test robustness. For visual data, different illumination conditions, poses, expressions, image qualities, and acquisition sessions should be represented where possible. For structured data, controlled perturbations such as missing values and measurement noise can be introduced to test stability. To ensure reproducibility, the experimental report should specify the dataset, number of identities, number of samples per identity, preprocessing operations, feature extractor, feature dimension, similarity measure, threshold-selection procedure, and hardware/software environment. Numerical performance should only be reported after these choices have been fixed and the system has been evaluated on held-out data.

TABLE I. VERIFICATION PERFORMANCE MEASURES

Metric	Definition	Interpretation
FAR	False acceptances / impostor trials	Lower is safer
FRR	False rejections / genuine trials	Lower improves usability
EER	Point where $FAR \approx FRR$	Compact operating indicator
Accuracy	Correct decisions / all trials	Overall correctness
ROC	True-positive vs false-positive trade-off	Threshold analysis

RESULTS AND DISCUSSION

Because this work defines a general verification framework rather than a single fixed dataset, numerical performance should be reported only after the selected feature extractor, dataset, and experimental protocol are fixed. The most important result is the separation between genuine and impostor score distributions. A reliable representation should place genuine comparisons toward higher cosine similarity or lower Euclidean distance, while impostor comparisons should occupy a clearly separated region.

A threshold sweep can be used to generate the operating characteristic of the system. At each threshold, the numbers of genuine accepts, genuine rejects, impostor accepts, and impostor rejects are recorded. This makes the trade-off between convenience and security explicit. For access-control applications, an operating point may prioritize a very low FAR, whereas a consumer-facing verification service may choose a more balanced operating point. The two similarity measures considered in this framework have different geometric interpretations. Euclidean distance directly measures separation in the feature space, while cosine similarity emphasizes orientation. If feature vectors are L2-normalized, cosine similarity becomes especially convenient for comparing learned embeddings. Nevertheless, empirical validation is required before selecting one metric as the deployment standard. Robustness should be assessed under acquisition variations. In visual verification, illumination, pose, facial expression, blur, occlusion, camera quality, and background changes can alter the extracted representation. A strong system should maintain useful score separation under such conditions. If performance degrades significantly, improvements should target preprocessing, feature learning, data diversity, or quality control rather than simply changing the decision threshold.

APPLICATION SCENARIOS

The framework can support controlled access to laboratories, classrooms, offices, examination systems, digital services, and other environments where a claimed identity must be verified. The modular



architecture also allows the same comparison engine to be used with different feature sources, provided that the representation is numeric and comparable.

For high-risk applications, similarity-based verification should be treated as one component of a larger authentication architecture. A second factor, device binding, transaction monitoring, or human review can reduce the consequences of a false acceptance. For lower-risk applications, a well-calibrated threshold combined with quality checks may provide an appropriate balance between usability and protection.

Table II to strengthen the system architecture:

Component	Function	Possible Implementation
Preprocessing	Standardize input	Resize, alignment, normalization
Feature extractor	Create representation	Descriptor or neural embedding
Matcher	Compute closeness	Euclidean / cosine
Decision layer	Apply threshold	Calibrated τ
Template store	Protect references	Encrypted, access-controlled storage
Audit layer	Record decisions	Minimal non-sensitive logs
Security layer	Prevent unauthorized or spoofed access	Liveness detection, multi-factor authentication, device binding
Quality control	Check input reliability before verification	Image/feature quality score, rejection of low-quality samples
Monitoring layer	Detect unusual verification activity	Failed-attempt tracking, anomaly detection, rate limiting

SECURITY AND PRIVACY ANALYSIS

Identity verification systems are security-sensitive because compromise of an identity template can have long-lasting consequences. Unlike a password, a biometric characteristic cannot simply be replaced after disclosure. Consequently, templates should be protected using access controls, encryption, secure storage, and carefully designed retention policies. Raw biometric images should not be retained when they are unnecessary for the intended purpose. Presentation attacks are another important concern. An attacker may attempt to present a photograph, replay a recorded signal, or manipulate an input to obtain an unauthorized acceptance. Similarity alone does not guarantee that the input originates from a live subject. Liveness detection or presentation-attack detection should therefore be added where the threat model requires it. Threshold selection also affects security. A permissive threshold may reduce false rejections but increase false acceptance, while a strict threshold may improve resistance to impostor attempts at the cost of rejecting legitimate users. Thresholds should therefore be selected from representative development data and periodically reviewed when acquisition conditions, sensors, populations, or operating environments change. The framework has several limitations. Its performance depends strongly on the quality and discriminative power of the selected feature representation. Similarity metrics cannot compensate for weak features or severe domain mismatch. In addition, a threshold calibrated on one



population or device may not transfer directly to another. Bias and unequal error rates across demographic or operational groups must be measured where applicable.

DEPLOYMENT CONSIDERATIONS

Deployment introduces factors that are often absent from laboratory evaluation. Camera or sensor changes can modify feature distributions, network delays can affect user experience, and storage failures can interrupt verification. A production system should therefore monitor data quality, score distributions, failure rates, and threshold performance without retaining more personal data than necessary. Template protection is particularly important. If embeddings are treated as sensitive identity data, they should be protected at rest and in transit. Access to enrollment and template-update operations should be restricted. Administrative actions should be logged, and old templates should be removed according to a defined retention policy. Where possible, verification can be performed locally on a trusted device so that raw biometric information does not need to leave the device. Bias and fairness should be considered whenever the system uses human biometric data. Evaluation should examine whether error rates vary across relevant groups or acquisition conditions. A single aggregate accuracy number can hide substantial differences. If unequal performance is observed, data collection, preprocessing, model selection, and threshold policy should be reviewed before deployment. Presentation-attack resistance should be evaluated separately from ordinary verification accuracy. A system may perform well on genuine and impostor comparisons while remaining vulnerable to replay or presentation attacks. Liveness detection, challenge-response mechanisms, trusted sensors, or additional authentication factors can be incorporated according to the threat model.

LIMITATIONS AND FUTURE WORK

Future work can extend the system with modern deep embeddings, multimodal fusion, adaptive template updating, liveness detection, uncertainty estimation, and privacy-preserving matching. Secure template transformation, federated learning, and on-device inference can also reduce exposure of sensitive data. Future experiments should validate the approach on a clearly specified dataset, compare alternative feature extractors and similarity measures, and test robustness under realistic environmental and adversarial conditions. Independent test sets should be used so that reported performance represents generalization rather than threshold or model tuning. Additional research can investigate score-level fusion for multimodal verification, dynamic thresholds based on quality estimates, continual monitoring for distribution shift, and explainable verification reports. These improvements should be evaluated against both security and usability requirements. Another important direction for future development is the integration of privacy, security, and fairness considerations into the verification framework. Future studies can evaluate the system across different demographic groups and operating conditions to identify potential performance disparities and reduce biased verification outcomes. Privacy-preserving techniques such as encrypted feature representations, secure template storage, and decentralized processing can further protect sensitive identity information. In addition, real-time monitoring and anomaly detection can be incorporated to identify repeated failed attempts, spoofing activities, or unusual access patterns. A comprehensive evaluation combining verification accuracy, computational efficiency, privacy protection, security, fairness, and user convenience would help establish the proposed framework as a reliable solution for practical identity verification applications.



CONCLUSION

This paper presented a modular similarity-based identity verification system that converts an input identity sample into a feature representation and compares it with a stored reference using measurable similarity or distance. The architecture separates acquisition, preprocessing, feature extraction, comparison, and decision making, allowing individual components to be improved without redesigning the complete verification pipeline. Euclidean distance and cosine similarity provide simple and interpretable mechanisms for comparing feature vectors, while threshold calibration transforms continuous scores into verification decisions. A rigorous evaluation should use genuine and impostor comparisons and report FAR, FRR, EER, accuracy, and operating characteristics. Security measures such as protected templates, liveness detection, and controlled retention are necessary when the system is deployed with sensitive identity information. The proposed framework provides a practical foundation for academic and prototype identity verification applications. Future implementations should validate the approach on a clearly specified dataset, compare alternative feature extractors and similarity measures, and test robustness under realistic environmental and adversarial conditions. The architecture separates acquisition, preprocessing, feature extraction, comparison, and decision making, allowing individual components to be improved without redesigning the complete verification pipeline. Euclidean distance and cosine similarity provide simple and interpretable mechanisms for comparing feature vectors, while threshold calibration transforms continuous scores into verification decisions. A rigorous evaluation should use genuine and impostor comparisons and report FAR, FRR, EER, accuracy, and operating characteristics. Security measures such as protected templates, liveness detection, and controlled retention are necessary when the system is deployed with sensitive identity information. The proposed framework provides a practical foundation for academic and prototype identity verification applications. Future implementations should validate the approach on a clearly specified dataset, compare alternative feature extractors and similarity measures, and test robustness under realistic environmental and adversarial conditions.

REFERENCES

- [1] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, Jan. 2004.
- [2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [3] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823.
- [4] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4690–4699.
- [5] P. Grother, M. Ngan, and K. Hanaoka, "Face recognition vendor test (FRVT) part 3: Demographic effects," *NISTIR 8280*, National Institute of Standards and Technology, 2019.
- [6] M. Wang and W. Deng, "Deep face recognition: A survey," *Neurocomputing*, vol. 429, pp. 215–244, Mar. 2021.
- [7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [8] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York, NY, USA: Wiley, 2001.



-
- [9] A. Ross, K. Nandakumar, and A. K. Jain, *Handbook of Multibiometrics*. Boston, MA, USA: Springer, 2006.
- [10] P. J. Phillips et al., "Face recognition vendor test 2002," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1173–1181, 2005.
- [11] A. K. Jain, K. Nandakumar, and A. Ross, "50 years of biometric research: Accomplishments, challenges, and opportunities," *Pattern Recognition Letters*, vol. 79, pp. 80–105, Aug. 2016.
- [12] A. K. Jain, A. Ross, and K. Nandakumar, *Introduction to Biometrics*. New York, NY, USA: Springer, 2011.
- [13] W. J. Scheirer, A. Rocha, R. Micheals, and T. E. Boult, "Meta-recognition: The theory and practice of recognition score analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1689–1695, Aug. 2011.
- [14] N. Poh, R. Wong, J. Kittler, and F. Roli, "Challenges and research directions in biometric score-level fusion," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 4, pp. 753–764, Dec. 2007.
- [15] K. Nandakumar, Y. Chen, S. C. Dass, and A. K. Jain, "Likelihood ratio-based biometric score fusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 342–347, Feb. 2008.
- [16] A. Ross and A. Jain, "Information fusion in biometrics," *Pattern Recognition Letters*, vol. 24, no. 13, pp. 2115–2125, Sept. 2003.
- [17] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp. 1042–1052, Oct. 1993.
- [18] P. Jonathon Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, Oct. 2000.
- [19] P. Grother, M. Ngan, and K. Hanaoka, "Face recognition vendor test (FRVT) part 2: Identification," NISTIR 8271, National Institute of Standards and Technology, 2019.
- [20] P. Grother, M. Ngan, and K. Hanaoka, "Face recognition vendor test (FRVT) part 1: Verification," NISTIR 8202, National Institute of Standards and Technology, 2018.
- [21] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1701–1708.
- [22] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1891–1898.
- [23] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 2892–2900.
- [24] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. British Machine Vision Conference (BMVC)*, 2015, pp. 1–12.
- [25] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823.
-