Al Furat Journal of Innovations in Engineering Applications (AJIEA)

# Fingerprint Verification and Classification Using a Siamese Neural Network with Real-Time Interface Integration

Hussein A.H Rokan [1*], Muhammed S.S Al-Kafaji [2], Ahmed G. Wadday[3]

[1] Electrical Engineering Techniques Department, Technical College /Al-Mussaib, Al-Furat Al-Awsat Technical University,Najaf, Iraq. hussein.abbas.tcm66@student.atu.edu.iq
[2] Al-Furat Al-Awsat Technical University,Najaf, Iraq. com.muh7@atu.edu
[3] Al-Furat Al-Awsat Technical University,Najaf, Iraq. ahmadghw@atu.edu.iq
[*]Corresponding author E-mail: hussein.abbas.tcm66@student.atu.edu.iq

https://doi.org/10.46649/ajiea.2026.02102

***Abstract.*** *Fingerprints can be interpreted as maps that show local features like minutiae. Fingerprint images also feature ridge flows that look just like contour lines, which are typically employed to show the latitude of region on a two-dimensional map. The cores and deltas are one of the sites that ridge flows focus on. The area around a core point is more like a mountain, while the area around a delta point is more like where two rivers meet. Fingerprint recognition has long been a basic biometric method for identifying people since it is distinctive, stable, and reliable. But standard fingerprint verification systems generally have trouble with changes in image quality, aberrations, and prints that aren't complete. This study suggests a deep learning approach that uses a Siamese Neural Network architecture made for verifying fingerprints. The model is trained on a dataset that include both synthetic and real fingerprint images of different levels of difficulty. A unique data generator creates matching and non-matching fingerprint pairs on the fly using real-time augmentation algorithms. The network learns a similarity function that can tell if two fingerprint images belong to the same person. The model is quite accurate, precise, and robust when tested on many different sets of data. Gradio is also used to develop a real-time user interface that lets users upload fingerprint pairs and get verification results right away.*
*This research demonstrates the advantages of deep metric learning in biometrics and lays a strong basis for future progress in secure, scalable, and interactive fingerprint recognition systems.*

***Keywords:*** *Fingerprint Verification;Identity Authentication;Siamese Neural Network(SNN);Deep Learning Artificial Intelligence (AI) tools.*

## 1. INTRODUCTION

Biometric systems are more used in security and identity frameworks. Fingerprints are a crucial biometric characteristic due to their permanence, uniqueness, and ease of use. Due to their reliability and widespread acceptance, fingerprints have been employed for decades in both civilian and forensic applications [1].

Traditional fingerprint matching methods heavily depend on manual feature extraction, primarily concentrating on minutiae points such as ridge endings and bifurcations. These systems require clear, high-resolution images and perform poorly with noise, incomplete prints, rotations, or distortion [2].

Consequently, they lack accuracy and reliability in uncontrolled environments, such as when users utilize mobile devices, public equipment, or evidence from forensic investigations [3]. Recent advancements in deep learning have transformed computer vision by enabling models to directly learn hierarchical features from raw input. Convolutional Neural Networks (CNNs) are currently the predominant method for image classification, object identification, and image comparison tasks. Nevertheless, conventional classification CNNs were inadequate for verification tasks intended to determine if two photos represent the same thing, rather than individually categorizing each image [4].

Siamese Neural Networks (SNNs) are an effective choice for verification tasks [5]. An SNN acquires a similarity score by evaluating pairs of inputs rather than identifying individual samples. This architecture is particularly effective for fingerprint verification, the process for estimating whether two images of fingerprints belong to the same individual [6].

This project develops a fingerprint verification system utilizing a Siamese Neural Network constructed in Keras. The system acquires knowledge through synthetic fingerprint datasets categorized as easy, medium, or hard, and subsequently undergoes evaluation with real fingerprint images. To enhance the model's robustness and generalization capabilities, extensive data augmentation is performed using modifications such as blurring, scaling, rotation, and translation.

The data generator produces matched and unmatched fingerprint pairs during model training. The model is trained using binary cross-entropy loss, and its performance is evaluated based on accuracy, precision, recall, F1-score, and area under the ROC curve (AUC).

Gradio is a Python library-based framework that let test machine learning models in a browser. It is used to make the system easy to use and interactive.

In the end, the application enables users to upload two fingerprint images or pairs and provides a real-time indication of whether the prints matched or not. SNN are applicable in various domains, including speech recognition, computer vision tasks, and the processing of graph-structured data. They possess strength due to their ability to discern similarities of inputs, enabling the construction of an embedding space that precisely represents patterns and relationships within the data. Exploring and modelling Siamese embedding spaces is crucial for enhancing their efficiency within various domains.

## 2. MATERIALS AND METHOD

### *2.1. TRADITIONAL FINGERPRINT MATCHING*

Historically, fingerprint identification systems utilized minutiae-based techniques that extract and compare unique characteristics, like ridge ends and bifurcations. These approaches align tiny points and figure out how similar they are [7]. They work well when everything is perfect, but they are quite sensitive to partial or distorted prints, poor scan quality, rotation or translation of fingers and changes due to pressure or skin condition. To deal with these problems, new methods that use global texture analysis, orientation fields, and frequency-based features were created. But because they had to follow strict criteria, they were less flexible and less able to deal with complex or noisy data [8].

### 2.1.1. MACHINE LEARNING AND DEEP LEARNING IN FINGERPRINT RECOGNITION

Machine Learning (ML) is a branch of AI that lets computers understand patterns and make decisions based on data. Machine learning algorithms derive rules from labelled examples rather than manually encoding them. This explains why machine learning is highly effective for challenging tasks such as fingerprint recognition, when establishing patterns by simplistic rules is difficult [9]. usually machine learning challenges categorized into three distinct categories:

• Calcification: Classifying an input into one of several predefined groups (for instance, identifying the individual associated with a fingerprint).

• Regression: Making a trained guess about something that continuously changing, (like the pressure or quality of a fingerprint.)

• Clustering: Gathering unlabelled data elements which are similar, such as various types of fingerprints or individuals with unknown identity.

In fingerprint systems, the primary objectives are classification and verification. Classification provides a unique identifier for each individual. Verification, conversely, analyses two fingerprints to determine if they belong to the same individual [10].

### 2.1.2. MODEL SELECTION: SVM, CNN, SNN

Three distinct models had been used and compared to examine fingerprint recognition performance across different learning paradigms: Support Vector Machine (SVM), Convolutional Neural Network (CNN), and Siamese Neural Network. Each model was chosen for a specific rationale within the framework of biometric identification or verification

### 2.1.3. SUPPORT VECTOR MACHINE – AS A BIAS LINE CLASSIFIER:

The SVM model were chosen due to its rapidity and clarity as a foundational method for fingerprint classification. It operates with flattened image vectors and requires little processing resources. Despite its simplicity, the Support Vector Machine (SVM) serves as a robust benchmark due to its reliance on margin-based categorization, which is most effective when data can be linearly separated in a high-dimensional space[11, 12].

### 2.1.4. CONVOLUTIONAL NEURAL NETWORK (CNN) – FEATURE LEARNING CLASSIFIER:

The CNN model was presented as a modern deep learning benchmark to evaluate the effects of spatial feature extraction in fingerprint images. The CNN doesn't use raw pixel vectors like SVM does. Instead, it learns hierarchical patterns directly from image data, like ridges and bifurcations. It was trained to classify fingerprints into more than one class[13, 14]

### 2.1.5. SIAMESE NEURAL NETWORK – VERIFICATION ARCHITECTURE

The Siamese network was designed for verifying fingerprints in pairs. It doesn't give class labels as SVM or CNN does. Instead, it learns a similarity metric between two fingerprints. This design is perfect for real-world verification situations, especially when encountering identities not seen during training (open-set recognition) [15, 16].

### 2.1.6. FINAL MODEL CHOICE: SIAMESE NETWORK

While SVM and CNN provide useful classification baselines, the goal of this study is not to classify fingerprints, but to compare them in order to determine whether two given inputs belong to the same identity. Therefore, as shown in Fig.1 the Siamese Neural Network was chosen as the final model architecture, as it is specifically designed for similarity-based verification tasks [17].

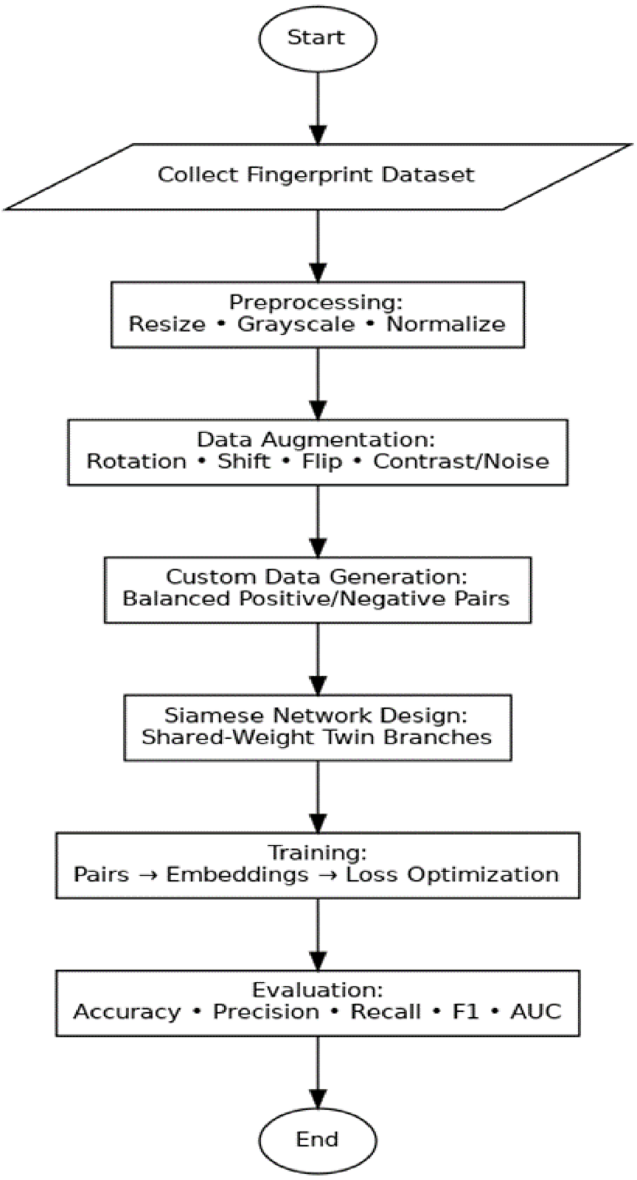



**Fig. 1. Flowchart Of The System Structure**

## 2.2. DATASET USED IN RESEARCH

This study utilizes the SOCOFing dataset. The SOCOFing provides an empty canvas for researchers to build up theories and reach conclusions. The SOCOFing is an essential resource for academics focused on fingerprint identification and categorization. The system serves as a benchmark to compare and evaluate various fingerprint recognition systems. It offers an abundance of fingerprint samples, facilitating advancement. It also enables the evaluation and refinement of novel models. The dataset's superior quality and accessibility render it an invaluable resource for academics developing and evaluating novel fingerprint recognition algorithms, thereby enhancing biometric systems and [18].

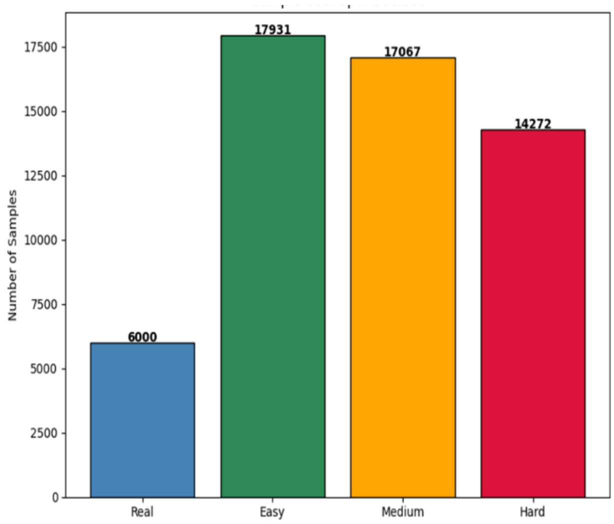



**Fig. 2. Types of Data Set Used In This Study.**

Synthetic datasets allow for controlled changes, including distortion, blur, and noise, but it may not have all the variety of real-world data. So, a typical way to make training more diverse is to combine synthetic and real fingerprint examples. The fingerprint scanners might pick up random noise because of issues with the sensors, the environment, or differences in how the skin touches the scanner. This effect is mimicked by adding a random noise block to the system architecture. Figure 2 represents the types of data which use in this study.

### *2.2.1. DATA PREPARATION*

The fingerprint verification model was trained using (hybrid data) which was a combination of synthetic and real fingerprint datasets. The synthetic datasets were categorized into three difficulty levels: easy, medium, and hard. These were generated with controlled distortions using the SFinGe tool. The real dataset served as a benchmark for generalization. All images were resized to 90×90 pixels and converted to grayscale. Datasets were stored in. npz and. npy formats and loaded using NumPy. Each fingerprint image was associated with a string-based label representing the subject's identity. To avoid data leakage, the dataset was separated using unique identities instead of random image sampling. This guaranteed that identities in the training set were not consistent with those in the validation set, facilitating a reliable evaluation of a generalization.

**Table 1. Dataset Categories Used In Model**

| Category | No. of images | Training set 90% | | Testing set 10% |
|---|---|---|---|---|
| | | Training data 80% | Validation data 20% | |
| Real | 6000 | 4320 | 1080 | 600 |
| Easy | 17931 | 12910 | 3228 | 1793 |
| Medium | 17067 | 12289 | 3072 | 1706 |
| Hard | 14272 | 10276 | 2569 | 1427 |
| Total | 55270 | 39795 | 9949 | 5526 |

### *2.2.2. DATA AUGMENTATION*

The image library was utilized for data augmentation to replicate variances seen in real fingerprint captures as shown in Fig3. The methodologies encompassed [19, 20]:

• Gaussian blur: use to simulate a smudging in image or pressure induced distortion.

• Affine transformations: use to perform a geometric changing like rotation, scaling, and translation.

• Additive noise: to simulates a real environmental interfaces.

• Random augmentation order: to prevent the risk of overfitting and applying different, random sequence for each sample during training. The transformations were specifically applied to synthetic fingerprint during training, enhancing dataset variety without augmentation memory use.

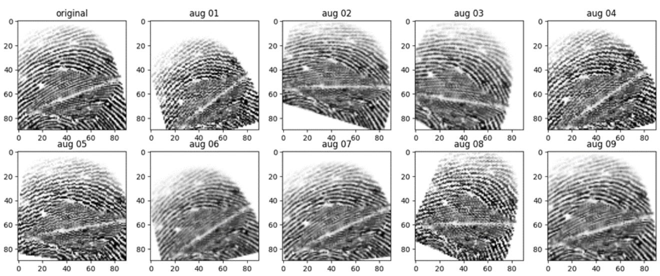



**Fig. 3. Fingerprint Augmentation**

visualizing the distribution of pixel intensities over nine enhanced variations of a single fingerprint image. Each augmented image was generated by a combination of Gaussian blur, affine transformations (scaling, translation, rotation), and constant padding.

In figure 4, Each rectangle signifies a singular enhanced sample. The central line in every rectangle represents the median pixel value, while the top and lower margins denote the 75th and 25th percentiles, respectively. The whiskers reach the furthest non-outlier values, while individual points, if present, signify outliers. This graphic explains the impact of augmentation on the distribution of grayscale intensity. A uniform distribution of pixel values signifies that the augmentations preserve the fundamental contrast and configuration of the fingerprint ridges. Conversely, any substantial deviation—such as numerous outliers or a highly skewed distribution—may indicate excessive transformation or noise, potentially detrimentally affecting model training. The pixel distributions across the augmentations stayed within an acceptable range, confirming the robustness and realism of the implemented transformations.

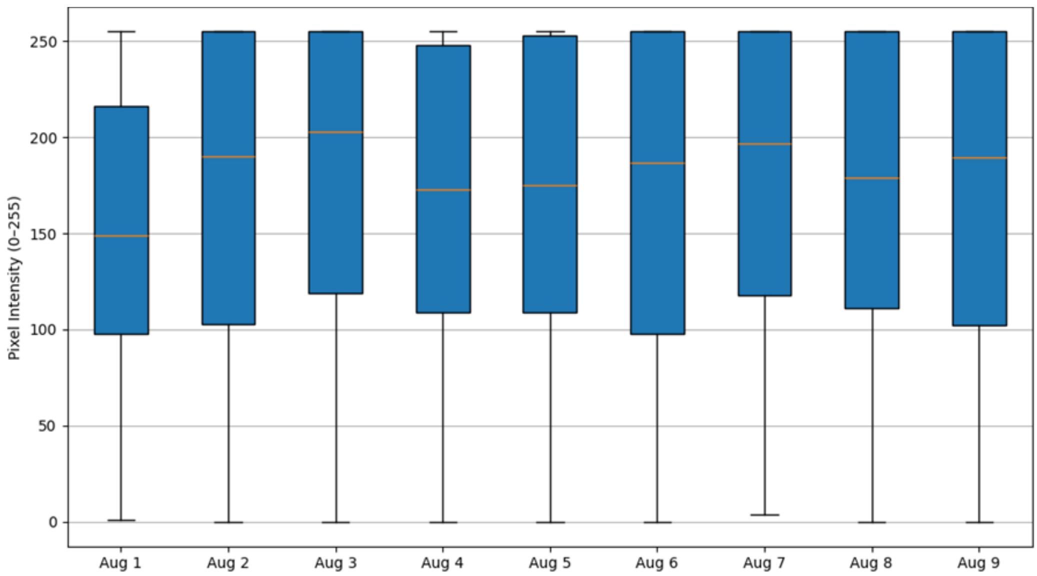



**Fig. 4. Distribution Of Pixel Intensity In Images Augmentation**

### *2.2.3. DATA GENERATION*

A custom data generator class was implemented by sub-classing keras. utils. Sequence Its basic responsibilities involved:

• Generate a dynamic batches of fingerprint pairs.

• Sampling of balance matched and unmatched pairs.

• Apply a real time augmentation for a synthetic sample.

• Normalize a pixel values to the range [ 0,1].

The data generation step is fundamental to this study, significantly enhancing the performance of the Siamese Neural Network by augmenting both the quantity and variety of the dataset. Advanced data augmentation techniques, including as rotation, translation, flipping, and modifications to brightness and contrast, were employed to replicate diverse fingerprint capture situations and enhance the model's generalization power. A particular data generator was built to consistently produce balanced image pairs, comprising: positive pair for fingerprint image belong to the same person and Negative pair for fingerprint images belong to distinct person. Pairs were matched according to shared identity labels, whilst non-matching pairs were selected to ensure no label mismatch. This method generated realistic training data without the necessity of precomputing every possibility combination [21].

Al Furat Journal of Innovations in Engineering Applications (AJIEA)

### 2.2.4. DATA CLASSIFICATION

The fingerprint classification approach is essential for structuring the dataset and preparing it for training and evaluation. This study utilized the Henry Classification System, a well-established standard in forensic and biometric applications, including national identity systems in Iraq. This approach classifies fingerprints into five principal categories according to ridge flow patterns (plain arch, tented arch, whorl, left loop, and right loop) as shown in Fig.5 [22].

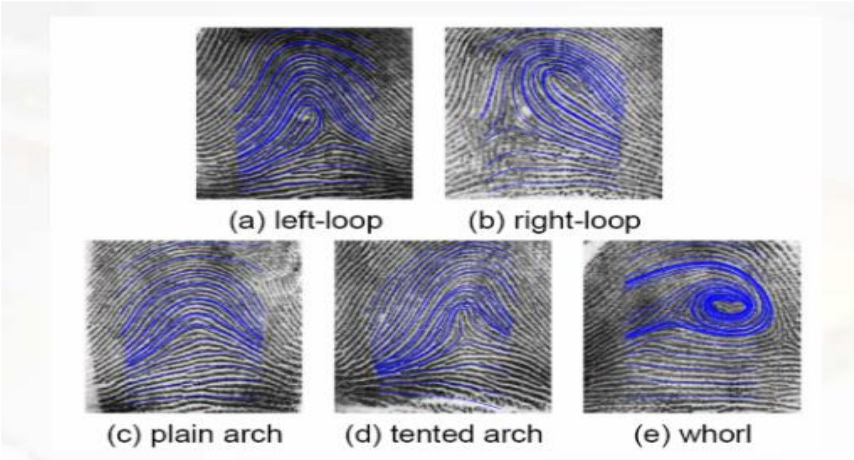



**Fig. 5. Henry Classification System**

### 2.2.5. MODEL TRAINING

Siamese neural network was constructed using a Keras functional API. Its design included:

-Conv2D → ReLU → MaxPooling → Dropout (repeated twice)

- Feature Comparison Layer:

-subtracts the output embedding of both branches

-Classification Head:

-Conv2D → MaxPooling → Flatten → Dense (64, ReLU) → Dense (1, Sigmoid).

This architecture outputs a probability score between 0 and 1

Indicating the likelihood that two input fingerprint match.

Model training was conducted using the following hyper parameters: Epochs: 15, Batch size: 32, Loss function: Binary Cross entropy, Optimizer: Adam and Validation split: 10% (by identity).

Figure 6 represents the steps of model training from the first step (loading data) to the last step which display the result if the two fingerprint belong to the same person or not.

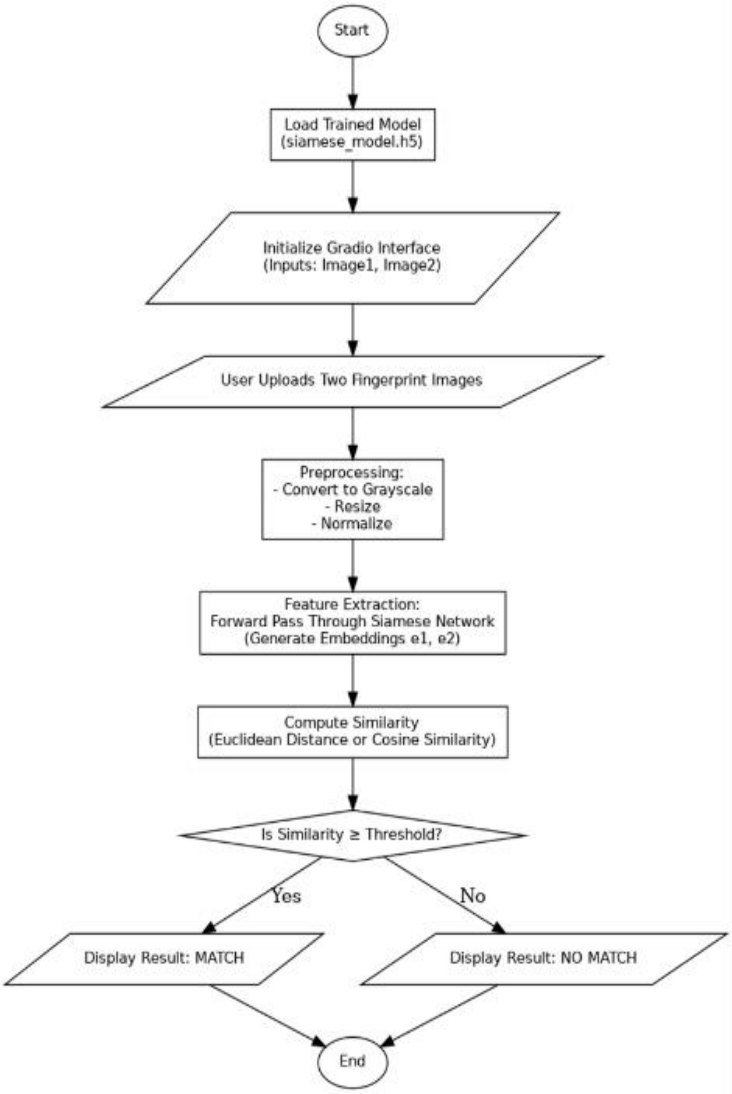



**Fig. 6. Gradio Flowchart**

### *2.2.6. ENVIRONMENT SETUP*

The system was implemented by using Python and executed in a GPU-enabled environment using Google Colab and Jupyter Notebook. The following libraries were used:

• Open cv : for load and preprocessing data

• TensorFlow/keras : to architect and training the model.

•Numpy: for load a data and numerical operations.

•Immaug : for images augmentation in real time.

•Matplotlib: to visualize an input sample and training metrics.

•Scikit-learn :to evaluate the performance of the model with metric.

•Gradio : create and building a browser-based user interface.

## 3. RESULT AND EVALUATION

This section illustrates the empirical results acquired during the training and validation of the Siamese Neural Network for fingerprint verification. The assessment involves quantitative performance measurements, visual diagnostic tools and qualitative testing with real-time input.

The model performed training for 15 epochs utilizing a batch size of 32. The training procedure employed dynamic pair formation and real-time data augmentation, which ensures a balanced combination of matched and unmatched fingerprint pairs. The validation set, obtained from unfamiliar identities, provided a realistic baseline for generalization. During training, both loss and accuracy metrics demonstrated consistent convergence. The training and validation curves exhibited close alignment, signifying negligible overfitting and homogeneous learning behaviour.

### *3.1. QUANTITATIVE EVALUATION*

The following metrics were calculated on the validation set using scikit-learn for SNN:

• Accuracy: The metric which evaluate the percentage of correctly identified samples relative to the total number of models [23].

$$ACC = \frac{100(TP + TN)}{TP + TN + FN + FP} \quad 1$$

• Precision: it is representing a percentage of correctly identified samples in the positive rate which determine using an equation [24]:

$$precision = \frac{TP}{TP + FP} \quad 2$$

•Recall: is a proportion of an adequately categorized samples to all correctly classified instances [25]:

$$Recall = \frac{TP}{TP + FN} \quad 3$$

• F1 score: is a popular measure that accounts for both accuracy and recall.

$$F1\,score = 2 * \frac{precision * Recall}{precision + Recall} \quad 4$$

ATU- AJIEA, Volume: 2, Issue: 01, 06, 30, 2026, 

Al Furat Journal of Innovations in Engineering Applications (AJIEA)

Where: TP, TN, FP, and FN are the amounts of true positive, true negative, false positive and false negative. These metrics confirm the model's ability to distinguish between genuine and impostor fingerprint pairs with high reliability. Table 2 represents the percentage of all these metrics.

**Table 2. Evaluation Metrics**

| Metric / Model | Accuracy | Precision | Recall (sensitivity) | F1 score | AUC Area under the ROC curve |
|---|---|---|---|---|---|
| SNN | 98.00 | 92.00 | 94.75 | 93.00 | 97.35 |
| CNN | 95.00 | 92.00 | 94,00 | 93.00 | 90.00 |
| SVM | 97.00 | 88.00 | 89.00 | 88.00 | 90.00 |

### *3.2. ROC CURVE AND CONFUSION MATRIX*

The ROC curve illustrated a high true positive rate across varying thresholds, confirming the robustness of the learned similarity metric [26]. A AUC of 0.9735 refers to a strong discriminative power. The confusion matrix demonstrated an equitable ratio of true positives to true negatives, with little misclassifications.

### *3.3. MODEL PERFORMANCE UNDER DIFFICULTY LEVELS*

The model's performance remained consistent and stable among synthetic fingerprints classified by difficulty as shown:

• Easy: accuracy still near perfect match.

• Medium: slightly drop in performance but handled well with augmentation.

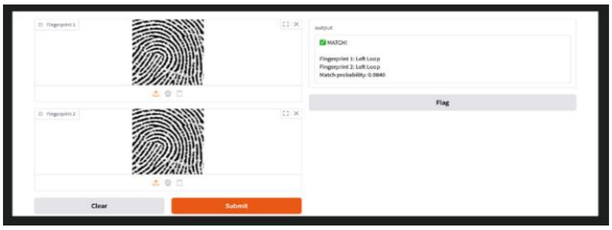
•Hard: the model achieved high recall which confirm a model robustness.

Testing on real fingerprints (not seen during training) gave a comparable outcome, which showed that the Siamese architecture can generalize.

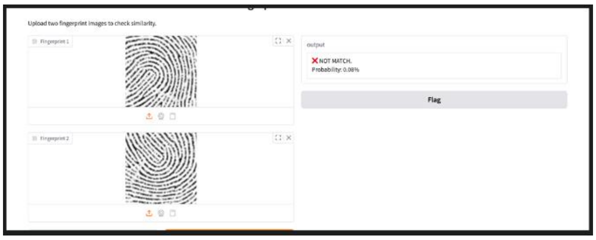
### *3.4. QUALITATIVE TESTING USING GRADIO*

The user interface based on Gradio made it possible to check fingerprint pairs manually. Users may upload images and get instant feedback on how likely they were to match. This helped prove that the model worked well with shifting or rotated fingerprints, partial or smudged prints and compares a synthetic with real images.each cases, including significant blur or distortion, occasionally resulted in false negatives or diminished confidence scores. These are anticipated under real-world settings and underscore candidates for more profound architectural improvements as shown in Fig.7 A, B.

Al Furat Journal of Innovations in Engineering Applications (AJIEA)




A




B

**Fig. 7. Qualitative Testing Results**

### *3.5. INFERENCE SPEED AND SYSTEM RESPONSIVENESS*

On standard GPU hardware, the model's inference time was under 100 milliseconds per pair. This makes the solution suitable for real-time applications such as mobile verification or biometric terminals.

## 4. CONCLUSION

The Siamese-based fingerprint verification technology exhibits significant real-world assurance. It provides a scalable, precise, and interactive solution for biometric identification. Despite several limitations, especially in harsh or deteriorated situations, the system offers a robust foundation for future research and implementation. the advantages of SNN can de summarize into optimized for binary verification tasks, require fewer samples per class, generalize to unseen identities (open-set verification) and eliminate the need to retrain for new users. This research introduced the creation of a fingerprint authentication system utilizing a Siamese Neural Network (SNN). The aim was to develop an architecture that can ascertain whether two fingerprint images belong to the same individual, ensuring high accuracy, resilience, and real-time performance.

To enhance this project, the subsequent recommendations are proposed:

• Modifying the architecture: can use a triple networks or introduce attention mechanisms.

• Expansion of dataset: use more divers of real dataset.

• Mobile and edge deployment: using Tensorflow lite for mobile or ONNX for integration into low power devices.

•Explainable AI: Apply technologies like Grad-CAM to elucidate network decisions and develop user confidence in essential applications

**Refrences**


[1] K. N. Win, K. Li, J. Chen, P. F. Viger, and K. Li, "Fingerprint classification and identification algorithms for criminal investigation: A survey," Future Generation Computer Systems, vol. 110, pp. 758-771, 2020.

[2] J. J. Shea, "Handbook of fingerprint recognition [Book Review]," IEEE Electrical Insulation Magazine, vol. 20, no. 5, pp. 45-45, 2004.

[3] J. Kalunga and S. Tembo, "Development of fingerprint biometrics verification and vetting management system," American Journal of Bioinformatics Research, vol. 6, no. 3, pp. 99-112, 2016.

[4] N. Amadu, F. W. Douamba, and M. I. Umar, "Fingerprint Recognition Based on Convolutional Neural Network," pp. 303-309: IEEE.

[5] Z. Li et al., "A novel fingerprint recognition method based on a Siamese neural network," Journal of Intelligent Systems, vol. 31, no. 1, pp. 690-705, 2022.

[6] J. Ma and X. Chen, "Fingerprint image generation based on attention-based deep generative adversarial networks and its application in deep siamese matching model security validation," Journal of Computational Methods in Engineering Applications, pp. 1-13, 2024.

[7] A. Jain, A. Ross, and S. Prabhakar, "Fingerprint matching using minutiae and texture features," vol. 3, pp. 282-285: IEEE.

[8] R. Singh, R. Singh, R. K. Tripathi, and P. Agarwal, "Fingerprint Recognition Using Artificial Neural Networks," Proceedings of the National Academy of Sciences, India Section A: Physical Sciences, pp. 1-9, 2025.

[9] C. Johnes, "Machine Learning in Fingerprint Recognition: Technological Advancements and Applications in Real World," Available at SSRN 5353281, 2025.

[10] N. Serrano and A. Bellogín, "Siamese neural networks in recommendation," Neural Computing and Applications, vol. 35, no. 19, pp. 13941-13953, 2023.

[11] P. Pravallika and K. S. Prasad, "SVM classification for fake biometric detection using image quality assessment: Application to iris, face and palm print," vol. 1, pp. 1-6: IEEE.

[12] H. G. Muhammad and Z. A. Khalaf, "A Survey Of Fingerprint Identification System Using Deep Learning," International Journal of Computing, vol. 17, no. 1, pp. 1-17, 2025.

[13] N. Sasuga, K. Ito, and T. Aoki, "Fingerprint feature extraction using cnn with multiple attention mechanisms," pp. 1-8: IEEE.

[14] M. Ali, C. Wang, and M. O. Ahmad, "A Deep CNN-Based Feature Extraction and Matching of Pores for Fingerprint Recognitio," IEEE Transactions on Biometrics, Behavior, and Identity Science, 2024.

[15] I. Jarraya, T. M. Hamdani, H. Chabchoub, and A. M. Alimi, "Tinysiamese network for biometric analysis," International Journal of Machine Learning and Cybernetics, pp. 1-15, 2025.

[16] S. M. Zaman, M. A. Hasan, M. R. Sadat, and M. A. Haque, "Comprehensive fingerprint recognition utilizing one shot learning with Siamese Network," 2023.

[17] D. Chicco, "Siamese neural networks: An overview," Artificial neural networks, pp. 73-94, 2021.

[18] Y. I. Shehu, A. Ruiz-Garcia, V. Palade, and A. James, "Sokoto coventry fingerprint dataset," arXiv preprint arXiv:1807.10609, 2018.

[19] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," Global Transitions Proceedings, vol. 3, no. 1, pp. 91-99, 2022.

[20] D. S. Ametefe, S. S. Sarnin, D. M. Ali, and Z. Z. Muhammad, "Fingerprint pattern classification using deep transfer learning and data augmentation," The Visual Computer, vol. 39, no. 4, pp. 1703-1716, 2023.
[21] R. Cappelli, D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, "Synthetic fingerprint generation," Handbook of fingerprint recognition, pp. 203-232, 2003.
[22] S. C. Dass, "Classification of fingerprints," Retrieved January, vol. 15, p. 2017, 2016.
[23] Z. Yao, J. M. Le Bars, C. Charrier, and C. Rosenberger, "Literature review of fingerprint quality assessment and its evaluation," Iet Biometrics, vol. 5, no. 3, pp. 243-251, 2016.
[24] A. Akram, A. Tariq, M. S. Ali, M. U. Tariq, and A. Altaf, "Recognizing Facial Expressions Across Cultures Using Gradient Features," Journal of Innovative Computing and Emerging Technologies, vol. 3, no. 1, 2023.
[25] A. Akram, J. Rashid, A. Jaffar, F. Hajjej, W. Iqbal, and N. Sarwar, "Weber Law Based Approach forMulti-Class Image Forgery Detection," Computers, Materials & Continua, vol. 78, no. 1, 2024.
[26] S. Yaqoob, A. Akram, T. Mansoor, H. Mustafa, and A. Altaf, "COVID-19 Detection using Curvelet Transformation and Support Vector Machine."