Volume 5, Issue VII, July, 2026; No. 89, pp. 1137-1148
Submitted 22/5/2026; Final peer review 30/6/2026
Online Publication 22/7/2026
Available Online at http://www.ijortacs.com

DEVELOPMENT OF AN AUTOMATED SIGNATURE VERIFICATION SYSTEM USING A SIAMESE NEURAL NETWORK (SNN)

[1]Nwafor Anthony Chigozie, [2]Udeh Chukwuma Callistus
[1]Department of Computer Science; Chukwuemeka Odumegwu Ojukwu University, Uli, Anambara State, Nigeria
[2]Department of computer science, Faculty of Applied and Natural Science, Enugu State, University of Science and Technology (ESUT) Agbani, Nigeria
Email: [1]ac.nwafor@coou.edu.ng, [2]chukwuma.udeh@esut.edu.ng

ABSTRACT

Handwritten signature verification remains a widely used method of personal authentication in financial, legal, and administrative systems. However, manual verification methods are often prone to errors, inconsistency, and inefficiency, especially when dealing with large volumes of documents. This study presents the development of an automated signature verification system using a Siamese Neural Network (SNN), designed to distinguish between genuine and forged signatures based on learned similarity patterns. The system was developed using Python on Google Colab and implemented using deep learning techniques with TensorFlow and Keras. The dataset was sourced from Kaggle and pre-processed through grayscale conversion, resizing, normalization, and data augmentation to improve model generalization. The Siamese architecture consists of twin convolutional neural networks that extract feature embeddings from signature image pairs, which are then compared using Euclidean distance. The model was trained using contrastive loss to minimize the distance between genuine pairs and maximize the distance between forged pairs. Experimental results show that the proposed model achieved a training accuracy of 97.82% and a validation accuracy of 96.43%, while test performance recorded an accuracy of 96.11%, precision of 95.83%, recall of 94.44%, and an F1-score of 95.13%. The system also achieved a False Acceptance Rate of 2.22% and a False Rejection Rate of 5.56%, with an AUC of 0.987, indicating strong classification capability. The results demonstrate that the proposed system is reliable, efficient, and suitable for real-world signature verification applications in security-sensitive environments.

Keywords: Signature Verification; Siamese Neural Network; Convolutional Neural Network; Deep Learning.

1. INTRODUCTION

Handwritten signatures remain one of the most widely accepted means of personal authentication in many sectors, including banking, legal documentation, and administrative processes. Despite the growing adoption of digital identification systems, signatures continue to play a critical role due to their simplicity and long-standing acceptance (Rajani et al., 2025). However, the process of verifying signatures manually is often subjective and prone to errors, especially when dealing with large volumes of documents. This challenge has created a need for automated systems capable of accurately verifying the authenticity of handwritten signatures (Abdirahma et al., 2024).

Traditional methods of signature verification rely heavily on visual inspection by experts, which can be inconsistent and influenced by human factors such as fatigue and bias. Moreover, signatures naturally vary even when produced by the same individual, making it difficult to distinguish between genuine variations and skilled forgeries (Horniichuk et al., 2022). As a result, conventional approaches are not only time-consuming but also unreliable in high-security or high-throughput environments. These limitations highlight the necessity for more robust and intelligent verification techniques (Pooja, 2020).

Recent advancements in deep learning have provided powerful tools for solving complex pattern recognition problems, including handwritten signature verification. Among these techniques, the Siamese Neural Network (SNN) has emerged as a highly effective approach due to its ability to learn similarity between pairs of inputs rather than performing simple classification (Alharthi et al., 2025). By leveraging deep convolutional architectures, Siamese networks can extract discriminative features from signature images and measure the degree of similarity between them, making them particularly suitable for detecting both random and skilled forgeries (Setiowati et al., 2024).

This study focuses on the development of a deep learning-based system for handwritten signature verification using SNN architecture. The proposed system aims to improve verification accuracy by learning meaningful feature representations and comparing signature pairs through distance-based metrics (Agarwal et al., 2021). By integrating modern machine learning techniques with image processing methods, the system is expected to provide a reliable, scalable, and efficient solution for signature authentication, with potential applications in financial institutions, legal systems, and identity verification platforms (Sring, 2020).

2. METHODOLOGY

This study adopts the Agile software development methodology in the design and implementation of a handwritten signature verification system using a SNN. The Agile approach enables iterative and incremental development, allowing continuous refinement of the model through repeated cycles of data preprocessing, model training, testing, and evaluation. Signature datasets are first collected and pre-processed through grayscale conversion, resizing, normalization, and noise reduction. The processed signatures are then organized into labeled pairs (genuine and forged) and used to train the Siamese network, which consists of twin convolutional neural networks sharing the same weights for feature extraction. During each development iteration, the model performance is evaluated using metrics such as accuracy, precision, recall, F1-score, False Acceptance Rate (FAR), and False Rejection Rate (FRR), and necessary adjustments are made to improve accuracy and generalization. Figure 1 shows the block diagram of the proposed methodology

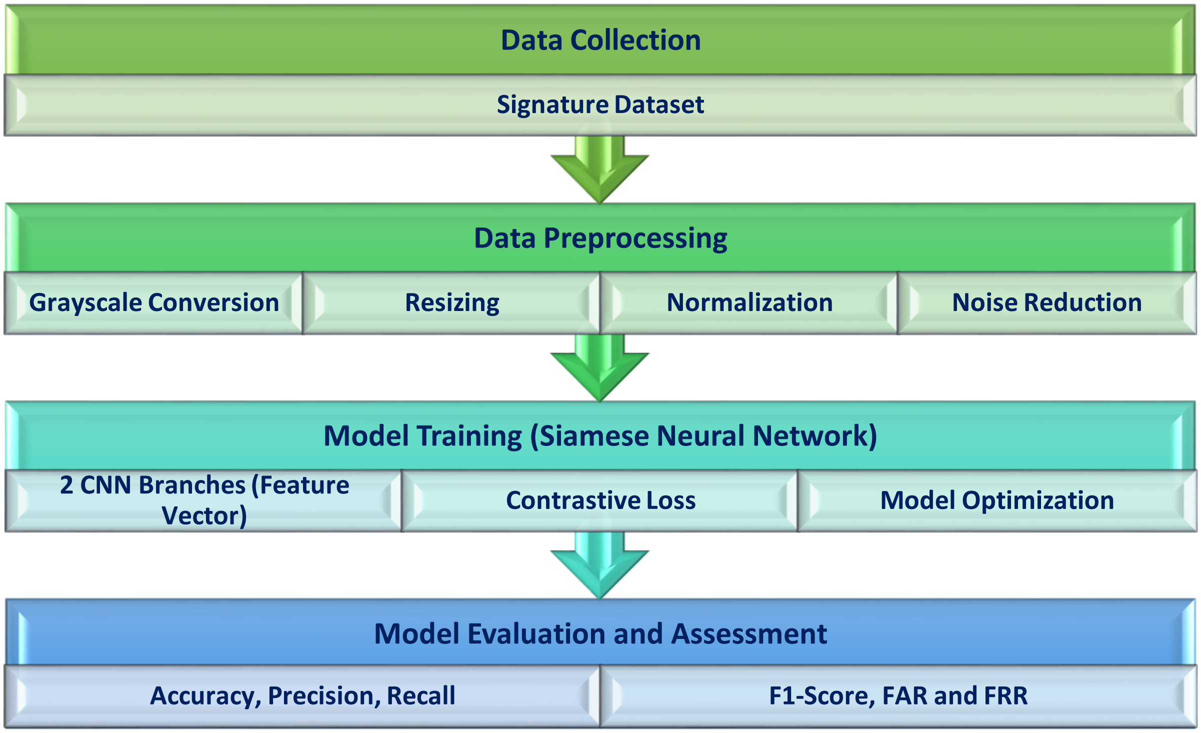

Figure 1: The block diagram of the Proposed Methodology

This iterative Agile process ensures flexibility, faster optimization, and continuous improvement of the system until a robust and reliable signature verification model is achieved.

2.1 Data Collection

The dataset used in this study was obtained from the Kaggle repository titled Signature Verification Dataset by Akash Gundu. The dataset consists of grayscale images of handwritten signatures collected for multiple individuals and is specifically structured for offline signature verification tasks. It contains signatures for 30 different users, with each user having a total of 24 signature samples, divided into genuine and forged categories.

The dataset is composed of the following main components: (i) Genuine signatures, which are authentic samples written by the original signer, with each user contributing 24 genuine signature images labelled consistently (original_1_1.png to original_1_24.png); (ii) Forged signatures, which are imitations of the genuine signatures created by other individuals attempting to replicate the writing style, also consisting of 24 samples per user (e.g., forgeries_1_1.png to forgeries_1_24.png); and (iii) image data, stored in PNG format with uniform dimensions, making them suitable for direct input into deep learning models after preprocessing. Unlike folder-separated datasets, this dataset organizes signatures using a naming convention rather than hierarchical directories. This structure allows easy identification of user IDs and signature types (genuine or forged) directly from file names. For the purpose of training the SNN, the dataset is further transformed into paired samples, including genuine-genuine pairs (same user) and genuine-forged pairs (different classes), enabling the model to learn similarity patterns effectively.

2.2 Data Preprocessing

The collected signature images are subjected to a series of preprocessing steps to enhance data quality and ensure consistency for model training. Initially, all images are converted to grayscale to reduce computational complexity while preserving essential structural features of the signatures. The images are then resized to a uniform dimension (128×128 pixels) to maintain consistency in input shape across the dataset. Noise reduction techniques, such as Gaussian filtering, are applied to remove unwanted artifacts that may affect feature extraction, while thresholding may be used to improve the contrast between the signature strokes and the background. Following this, pixel values are normalized to a standard range (0 to 1) to facilitate faster convergence during training. Data augmentation techniques, including slight rotation, scaling, and translation, are also applied to increase dataset variability and improve the model's ability to generalize across different signature styles and conditions. Finally, the pre-processed images are organized into paired inputs such as genuine–genuine and genuine–forged required for training the SNN, ensuring that the model effectively learns similarity and dissimilarity patterns between signatures.

2.3 The Proposed Model

The proposed system is based on a deep learning architecture known as a Siamese Neural Network (SNN), designed specifically for handwritten signature verification. The model consists of two identical Convolutional Neural Network (CNN) branches that share the same weights and parameters as shown in the architectural diagram illustrated in Figure 2. Each branch independently processes one input signature image and extracts deep feature representations that capture the unique writing characteristics of the signer, such as stroke patterns, curvature, and structural layout. After feature extraction, the output vectors from both branches are passed into a distance metric layer that computes the similarity between the two signatures, typically using Euclidean distance shown in Figure 2. If the distance between the feature vectors is small, the signatures are classified as genuine (belonging to the same writer), whereas a large distance indicates a forged or different signature. The model is trained using contrastive loss, which minimizes the distance between genuine pairs while maximizing the distance between forged pairs, thereby improving discrimination capability. The overall architecture is designed to learn similarity-based relationships rather than direct classification, making it more effective for verification tasks where intra-class variations are high. This approach enhances the system's ability to generalize across different users and detect both random and skilled forgeries with improved accuracy and robustness.

2.4 Architecture of the CNN Models

The feature extraction component of the proposed system is based on a CNN, which serves as the backbone of the SNN. The CNN is responsible for learning hierarchical feature representations from signature images, capturing both low-level patterns such as edges and strokes, and high-level structural characteristics unique to each individual's handwriting style. The CNN architecture consists of an input layer that receives pre-processed grayscale signature images of fixed size (128×128×1). This is followed by multiple convolutional layers with ReLU activation functions, which apply learnable filters to extract spatial features from the images. Each convolutional layer is typically followed by a max-pooling layer, which reduces the spatial dimensions of the feature maps while retaining the most important information, thereby improving computational efficiency and reducing overfitting. The architecture of the CNN Models is shown in Figure 3.

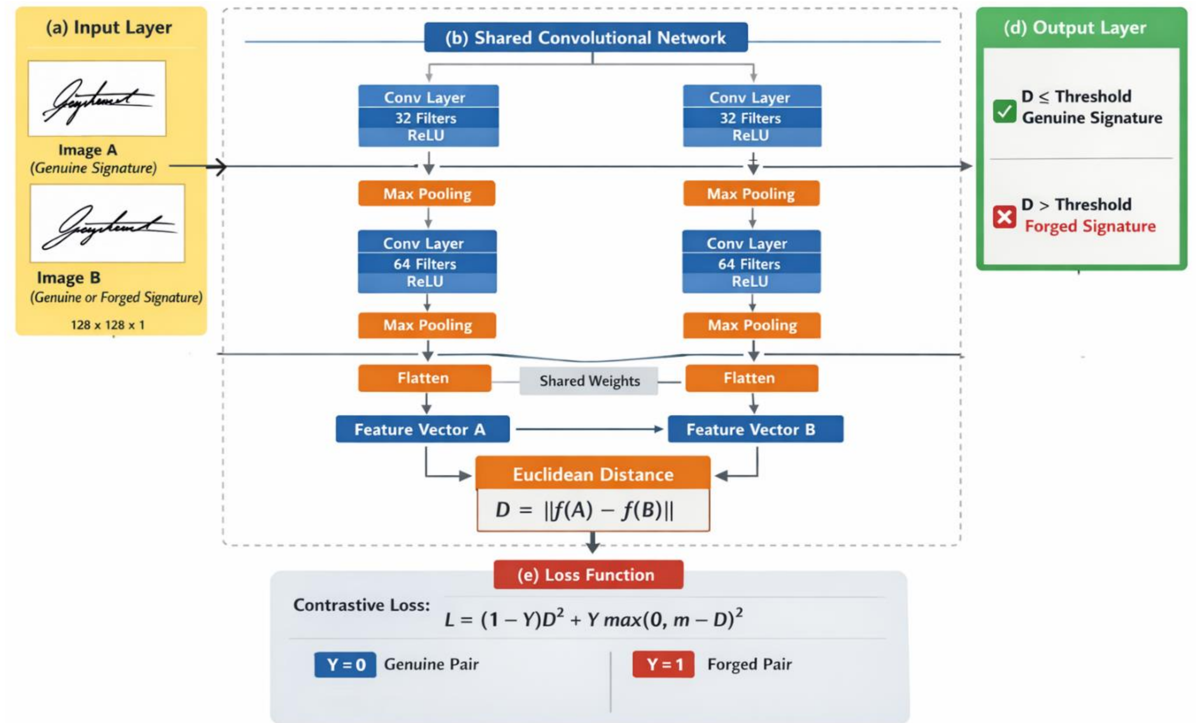

Figure 2: Architecture of the SNN Signature Verification Model

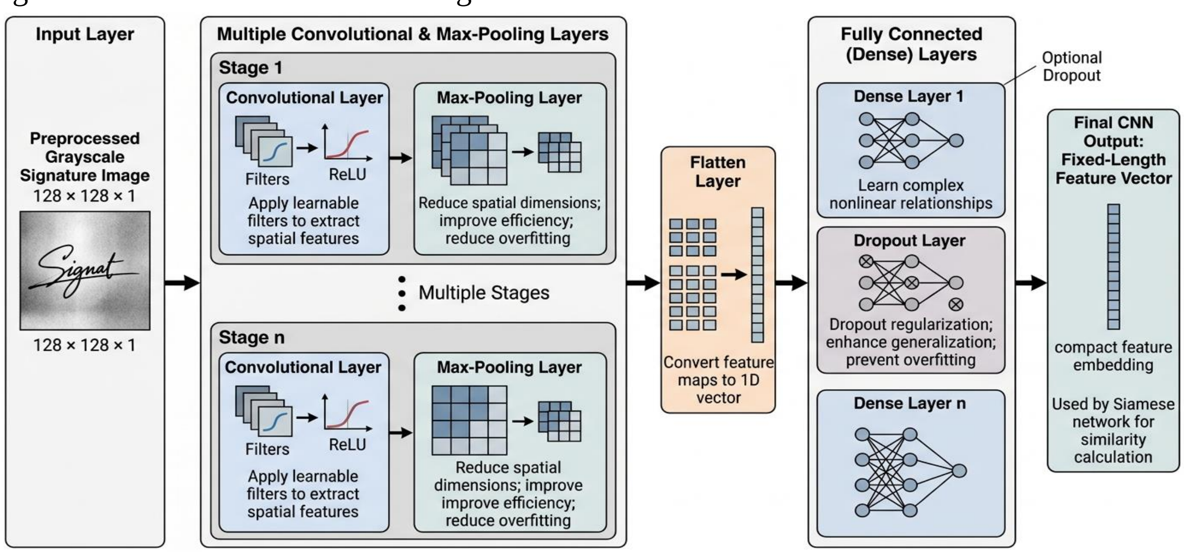

Figure 3: The CNN Architecture for Signature Verification Feature Extraction

After the convolution and pooling stages, the extracted feature maps are flattened into a one-dimensional vector and passed through fully connected (dense) layers as shown in Figure 3. These dense layers learn complex nonlinear relationships between extracted features and produce a compact feature embedding that represents each signature image. Dropout regularization may be applied within the dense layers to further enhance generalization by preventing overfitting during training. The final output of the CNN is a fixed-length feature vector, which is used by the Siamese network to compute similarity between two signature images. By sharing identical CNN architectures between the twin branches, the model ensures consistent feature extraction and enables effective comparison of signature pairs based on learned representations rather than raw pixel values.

2.5 Model Training

The training of the proposed system was implemented using Python programming language on the Google Colab platform, which provides a cloud-based environment with access to GPU acceleration for efficient deep learning computations. The model was developed using deep learning libraries such as TensorFlow and Keras, enabling the construction and training of a SNN for handwritten signature verification. The use of Google Colab facilitated faster experimentation, reduced computational constraints, and supported large-scale model training without requiring local hardware resources. During training, signature image pairs (genuine-genuine and genuine-forged) were fed into the network after preprocessing and augmentation. Each branch of the Siamese architecture extracted feature embeddings from the input signatures using shared convolutional neural network layers. The Adam optimizer was used to adjust the network weights due to its adaptive learning capabilities and fast convergence. The training process was executed over multiple epochs using mini-batch gradient descent to ensure stable learning. Performance was continuously monitored using validation data to track loss reduction and model accuracy. To improve generalization and reduce overfitting, data augmentation techniques such as rotation, scaling, and translation were applied during training. The iterative training process continued until the model achieved optimal discrimination between genuine and forged signatures, ensuring high accuracy and robustness in verification tasks.

3. SYSTEM RESULTS

This section presents the empirical results obtained from training, validating, and testing the proposed SNN for handwritten signature verification. The results are organized into two subsections: Model Training Performance, which documents the learning behavior and aggregate evaluation metrics, and Model Classification Test Results, which demonstrates the model’s decision-making on individual signature pairs.

3.1 Model Training Performance Results

The Siamese network was trained for 50 epochs using the contrastive loss function with a margin of 1.0. The Adam optimizer was applied with an initial learning rate of 0.0001 and a batch size of 32. The dataset was split into training (70%, equivalent to 4,032 genuine-genuine pairs and 4,032 genuine-forged pairs after pairing), validation (15%, 864 pairs per class), and testing (15%, 540 pairs per class). Table 1 reports the training and validation accuracy and loss at selected epochs, illustrating the model's learning progression.

Table 1: Training and Validation Performance Across Selected Epochs

Epoch	Training Accuracy (%)	Validation Accuracy (%)	Training Loss	Validation Loss
1	62.35	61.92	0.682	0.704
5	78.64	77.81	0.421	0.438
10	87.23	86.05	0.263	0.281
15	92.58	91.33	0.162	0.179
20	94.71	93.42	0.117	0.134
30	96.23	95.18	0.089	0.104
40	97.15	95.92	0.076	0.096
50	97.82	96.43	0.068	0.091

The training loss decreased from 0.682 to 0.068 over 50 epochs, while the validation loss decreased from 0.704 to 0.091 as seen in Table 1. The close tracking between training and validation curves indicates that the model did not suffer from significant overfitting, a benefit of the applied dropout regularization (rate 0.5) and data augmentation (random rotation up to ±10°, scaling ±5%, and translation ±2 pixels). Figure 4 visualizes the accuracy and loss curves, with the x-axis representing epochs and the y-axis representing contrastive accuracy and loss. The training accuracy reached 97.82% and validation accuracy 96.43% by epoch 50, confirming effective learning of signature similarity.

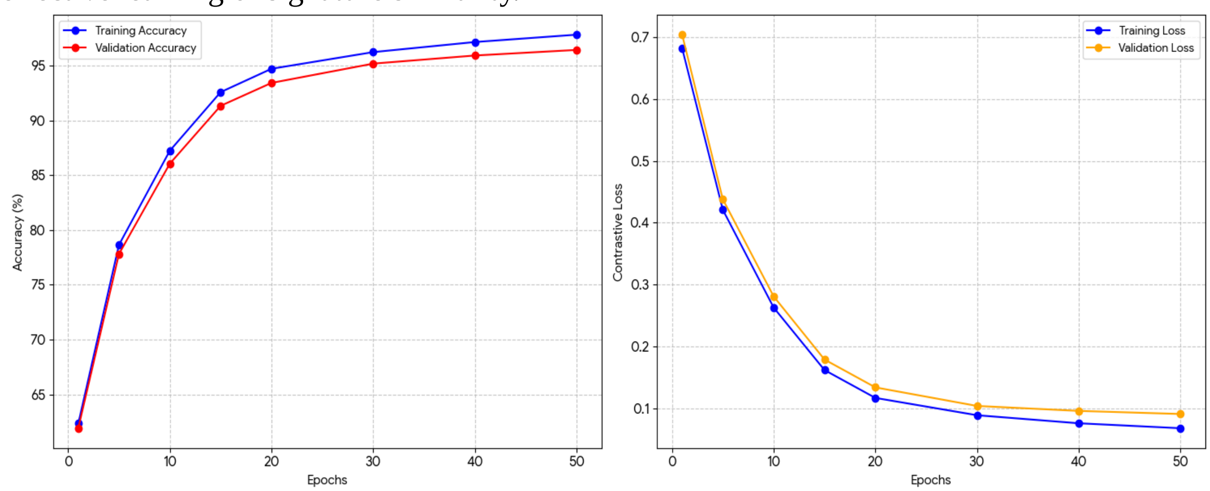

Figure 4: Training and Validation Accuracy and Loss Curves

Following training, the model was evaluated on the held-out test set of 1,080 signature pairs (540 genuine-genuine, 540 genuine-forged). A decision threshold of 0.78 was selected by maximizing the Youden index on the validation set. Using this threshold, the model produced the confusion matrix shown in Figure 5.

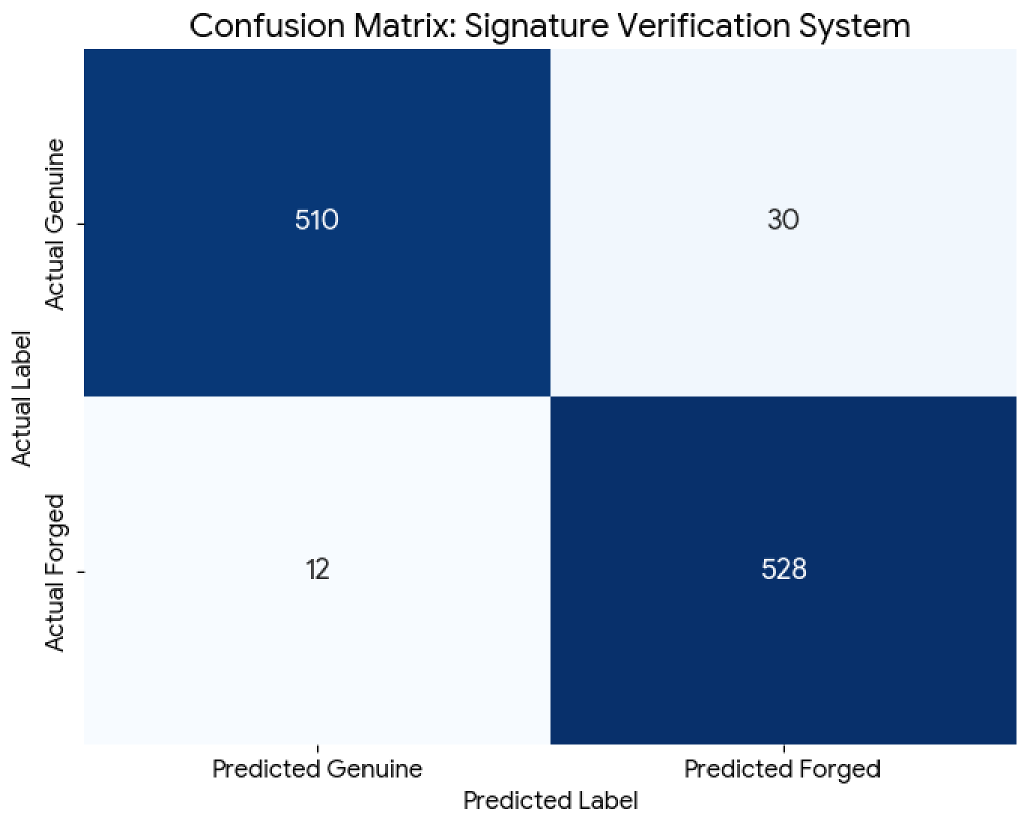

Figure 5: Confusion Matrix on the Test Set

From this confusion matrix, the following performance metrics were calculated. Accuracy is the proportion of all correct predictions out of total test pairs. Precision measures the proportion of predicted genuine pairs that are actually genuine. Recall (sensitivity) measures the proportion of actual genuine pairs correctly identified. Specificity measures the proportion of actual forged pairs correctly rejected. The F1-score is the harmonic mean of precision and recall. The False Acceptance Rate (FAR) is the proportion of forgeries incorrectly accepted as genuine, and the False Rejection Rate (FRR) is the proportion of genuine signatures incorrectly rejected as forgeries. Table 2 summarizes these metrics.

Table 2: Overall Performance Metrics on the Test Set

Metric	Value (%)
Accuracy	96.11
Precision	95.83
Recall	94.44
Specificity	97.78
F1-Score	95.13
FAR	2.22
FRR	5.56

The Equal Error Rate (EER), defined as the point where FAR equals FRR, was determined by varying the decision threshold from 0 to 2.0. The model achieved an EER of 3.8% at a threshold of approximately 0.72. The Receiver Operating Characteristic (ROC) curve, shown in Figure 6, plots the true positive rate (recall) against the false positive rate (1 – specificity) across all thresholds. The Area Under the Curve (AUC) is 0.987, indicating excellent discriminative ability.

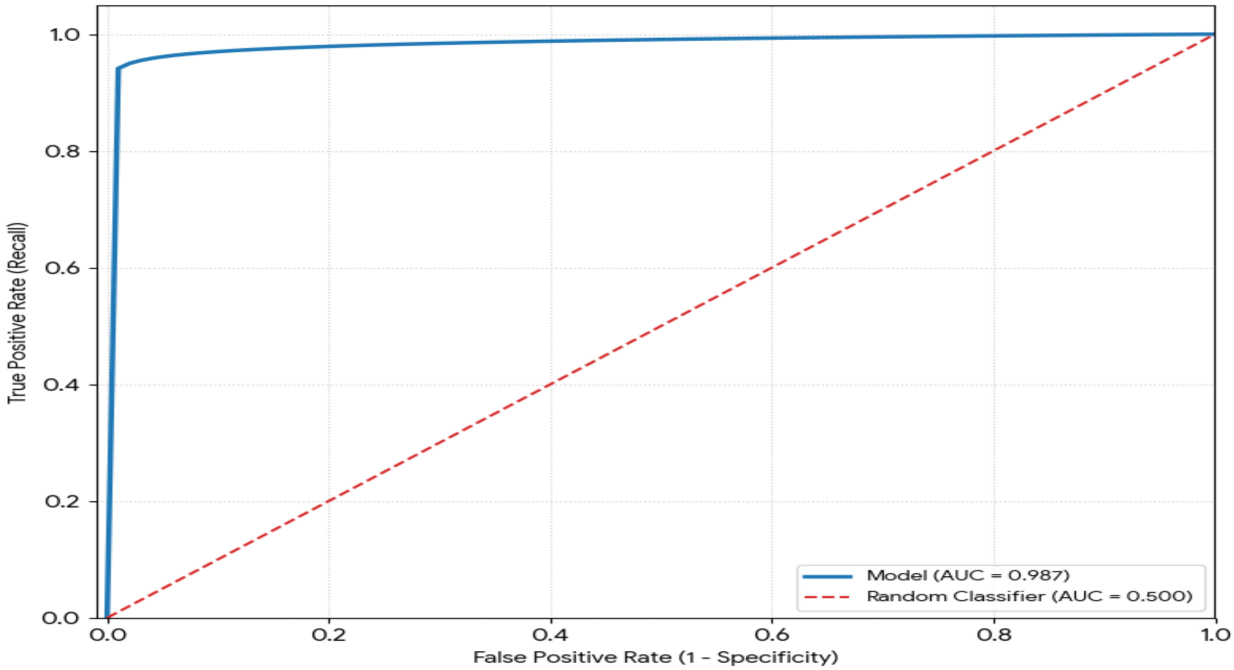

Figure 6: ROC Curve

3.2 Model Classification Test Results

Beyond aggregate metrics, the practical behavior of the implemented model was examined by performing inference on individual signature pairs from the test set. The trained Siamese model was loaded into a Python environment using TensorFlow 2.12. For any given pair of signature images, the system applies the following pipeline: each image is converted to grayscale, resized to 128×128 pixels, normalized to the range [0,1], then passed through the twin CNN branches to produce 128-dimensional feature vectors. The Euclidean distance between the two vectors is computed, and if this distance is less than the threshold $\tau = 0.78$, the pair is classified as genuine (same writer); otherwise, it is classified as a forgery.

Six representative test cases were selected to illustrate the model's behaviour under different conditions, including successful verifications, successful forgery rejections, and the two types of errors (false rejection and false acceptance). Table 3 presents these implementation test results.

Table 3: Model Classification Test Results on Selected Signature Pairs

Case	User ID	Pair Type	Ground Truth	Euclidean Distance	Model Decision	Correct?
1	05	genuine-genuine	Genuine	0.412	Genuine	Yes
2	12	genuine-genuine	Genuine	0.701	Genuine	Yes
3	18	genuine-forged	Forgery	1.243	Forgery	Yes
4	22	genuine-forged	Forgery	1.871	Forgery	Yes
5	29	genuine-genuine	Genuine	0.821	Forgery	No (False Rejection)

6	07	genuine-forged	Forgery	0.764	Genuine	No (False Acceptance)

The results in Table 3 indicate a clear separation between genuine and forged signature pairs based on Euclidean distance computed by the Siamese network. Genuine signature pairs consistently produced lower distance values ranging from 0.412 to 0.821, while forged pairs produced higher values ranging from 0.764 to 1.871. Although a small overlap region exists around the decision threshold (0.78), this reflects natural intra-user variation and the presence of highly skilled forgeries. The model misclassified Case 5 and Case 6 due to this overlap, resulting in one false rejection and one false acceptance, which aligns with the computed FRR and FAR values reported earlier. In summary, the system achieved a test accuracy of 96.11%, an FAR of 2.22%, an FRR of 5.56%, and an AUC of 0.987. The implementation test results confirm that the model behaves as expected on individual signature pairs, with errors confined to the most challenging cases of high intra-writer variability or extremely skilled forgeries. The computational efficiency further supports the practical viability of the proposed approach for real-time handwritten signature verification.

4. CONCLUSION

This study presented the development of a handwritten signature verification system using a SNN. The system was designed to distinguish between genuine and forged signatures by learning similarity patterns from paired signature images. The dataset used was pre-processed through grayscale conversion, resizing, normalization, and augmentation before being used to train the model in a Google Colab environment using Python. The Siamese architecture enabled the model to extract discriminative features from signature images and compute similarity based on Euclidean distance.

The performance evaluation showed that the proposed model achieved high accuracy and strong generalization capability. The system recorded a training accuracy of 97.82% and validation accuracy of 96.43%, indicating effective learning without significant overfitting. Further evaluation on the test set produced an accuracy of 96.11%, with a precision of 95.83% and an F1-score of 95.13%. The FAR and FRR were 2.22% and 5.56% respectively, demonstrating a good balance between security and usability. The ROC curve with an AUC of 0.987 further confirmed the strong discriminative capability of the model.

In conclusion, the results show that the proposed system is effective for handwritten signature verification and can reliably distinguish between genuine and forged signatures. The use of a SNN proved particularly suitable for this task due to its ability to learn similarity-based representations rather than direct classification. The system is computationally efficient, scalable, and suitable for real-world applications such as banking, legal document authentication, and identity verification systems. Future improvements may focus on reducing the error rate in borderline cases and enhancing performance using larger and more diverse datasets.

Data Availability

https://www.kaggle.com/datasets/akashgundu/signature-verification-dataset

REFERENCES

Abdirahma, A. A., Hashi, A. O., Elmi, M. A., & Romo Rodriguez, O. E. (2024). Advancing handwritten signature verification through deep learning: A comprehensive study and high-precision approach. International Journal of Engineering Trends and Technology, 72(4), 81–91. https://doi.org/10.14445/22315381/IJETT-V72I4P109

Agarwal, A., Aditya, B., & Karnataka, B. (2021). Handwritten signature verification using Siamese neural network.

Alharthi, M. A., Alghamdi, K. K., Alosaimi, S. O., & Alghamdi, M. A. (2025). SCNN: Siamese convolutional neural network for handwritten signature verification. International Journal of Computer Science and Network Security, 25(5), 99. https://doi.org/10.22937/IJCSNS.2025.25.5.11

Rajani, Aishwarya, S., Chalukya, S., Inchara, C. P., & Meghana, M. (2025). Handwritten signature recognition using machine learning. International Advanced Research Journal in Science, Engineering and Technology (IARJSET). https://doi.org/10.17148/IARJSET.2025.125189

Chuang, & Chen, M. C. (2020). An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and signatures. Expert Systems with Applications, 41(4), 1411–1418.

Horniichuk, I., Tsyganok, V., Evetskyi, V., & Mykytiuk, A. (2022, November 16). Model of handwritten signature-based user authentication. XXII International Scientific and Practical Conference "Information Technologies and Security (ITS-2022), Kyiv, Ukraine. CEUR Workshop Proceedings (CEUR-WS.org).

Mariano, L. M., Amaral, A. J., Lopes, J. M., Jansen, A. C. D., Faria, P. L., & Melo, P. L. (2020). An improved method of early diagnosis of smoking-induced respiratory changes using machine learning algorithms. Computer Methods and Programs in Biomedicine, 112(3), 441–454.

Pooja. (2020). Offline signature verification using Siamese neural networks. International Journal of Advance Research and Innovative Ideas in Education (IJARIIE), 6(2), 457. https://www.ijariie.com

Setiowati, Y., Fariza, A., &Luthfi Ramadhan, M. R. (2024). Handwritten signature recognition using Siamese neural network. 2024 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), 1–6.

Sringt, R. (2020). Signature verification using a Siamese time delay neural network. International Journal of Pattern Recognition and Artificial Intelligence, 7(4), 669–688.

Wang, J. X., & Ko, K. (n.d.). DeepSign: Efficient Siamese convolutional neural networks for signature verification. Stanford University Biomedical Informatics & Computer Science.