Signtinel: A Deep Learning-Based System for Handwritten Signature Authentication and Fraud Detection

Alberto, Christian Joshua
College of Computing and Information Technologies
National University - Philippines
Manila, Philippines
albertocq@students.@national-u.edu.ph

Abstract - Handwritten signatures remain a widely accepted form of identity verification, especially in financial and legal transactions. However, their continued use exposes critical vulnerabilities to forgery and fraud. This research presents Signtinel, a deep learning-based system that performs handwritten signature authentication using a Siamese Neural Network (SNN). Unlike traditional manual verification or sequence-based models like Recurrent Neural Networks (RNNs), the Siamese architecture is specifically designed to compare paired inputs and learn similarity measures, making it ideal for signature verification tasks. The study explores the model's performance in distinguishing between genuine and forged signatures using image pair datasets. Experimental results demonstrate that Signtinel achieves high accuracy and generalization even with limited training samples per user, showcasing its potential for real-world deployment. This research addresses the growing concern of signature fraud and contributes a scalable, reliable solution for improving identity verification in security-critical industries.

Index Terms - signature verification, siamese neural network, fraud detection, deep learning, identity authentication, biometric security.

I. Introduction

In an increasingly digital and globalized world, securing personal identity and verifying authenticity remain critical challenges across various sectors from banking and legal services to government and healthcare. While advanced biometric technologies such as facial and fingerprint recognition have emerged, handwritten signatures remain one of the most commonly accepted and trusted forms of identity verification, especially in legal documents and financial transactions [1].

However, this long-standing reliance introduces a significant vulnerability: signature fraud. According to the 2023 report by the Association of Certified Fraud Examiners (ACFE), document-related fraud, including forged signatures, accounts for nearly 15% of occupational fraud cases worldwide, leading to billions in financial losses annually [2]. Traditional signature verification methods, often performed manually by human experts, are labor-intensive and vulnerable to inconsistency, fatigue, and subjective errors [3].

To address this growing concern, this study introduces Signtinel, a deep learning-based signature authentication system designed to accurately differentiate between genuine and forged handwritten signatures. At the core of Signtinel lies the Siamese Neural Network (SNN), a specialized architecture that learns to compare two input samples and measure their similarity. This structure makes Siamese networks particularly effective for verification tasks, such as signature matching, where the goal is not classification but comparison.

In comparison to other models like Recurrent Neural Networks (RNNs) which are well-suited for sequential data such as text, speech, or time series [4], Siamese networks provide a more appropriate architecture for visual comparison tasks like image-based signature verification. RNNs process information sequentially, making them ideal for understanding temporal dependencies, but less effective in learning subtle spatial patterns in images. On the other hand, Siamese networks, when trained on paired signature images, learn a robust feature space where genuine pairs are close together and forgeries are far apart, enabling high-precision verification without requiring a large number of labeled classes.

The adoption of Siamese networks in Signtinel brings several advantages. Few-shot learning capability, enabling the system to generalize well even with limited samples per user [5]. Robustness against intra-class variations, such as differences in writing pressure, stroke direction, or slight changes in signature style. Scalability, as it doesn't require retraining the entire model for each new user, only a reference signature is needed for comparison.

Signtinel is envisioned to be deployed in contexts where signature verification is critical such as banks, legal institutions, government agencies, and corporate environments. Potential users include bank tellers, legal clerks, fraud analysts, and developers working on secure transaction platforms. The system can be integrated into document management workflows, mobile apps, or authentication services, adding a idealreliable layer of security.

By focusing on a Siamese network-based approach, this study aims to contribute to the development of more secure, scalable, and intelligent systems for signature verification and fraud detection. Signtinel not only addresses the technical limitations of traditional methods and some deep learning models like RNNs, but also aligns with the global demand for more trustworthy and efficient identity verification practices.

II. Review of Related Literature

This study focuses on deep learning, specifically the Siamese Neural Network (SNN) architecture, which is particularly effective in tasks requiring pairwise comparison, such as offline handwritten signature verification. Unlike traditional classification models, SNNs learn a similarity function that determines the degree of similarity between two inputs. This capability is especially useful in biometric applications that rely on relational matching rather than categorical predictions [6].

Earlier approaches to signature verification depended on manual visual inspection by forensic experts, which was labor-intensive and prone to subjectivity. Traditional algorithmic methods, such as Hidden Markov Models (HMMs), were initially employed to model sequential data but struggled to capture the spatial characteristics of handwritten images. With advancements in deep learning, Recurrent Neural Networks (RNNs) gained popularity for modeling temporal dependencies, yet they remained suboptimal for static image data [11], [13]. In contrast, Convolutional Neural Networks (CNNs) greatly improved image-based recognition by extracting hierarchical spatial features, becoming a standard in signature verification systems [12].

Siamese Networks emerged as a robust alternative to these models. They do not require the creation of a separate model for each class or user, which enhances scalability and real-world applicability. By projecting input images into a shared embedding space and computing their Euclidean distance, SNNs provide an effective solution for scenarios with limited training data, a paradigm known as few-shot learning [6], [9].

In this study, the SNN is deployed through a Streamlit-based web application. The system allows users to upload two signature images, which are preprocessed through grayscale conversion, resizing, and normalization. These processed images are then passed through the network to generate 10-dimensional embeddings. The similarity between the two embeddings is calculated using Euclidean distance, and a predefined threshold (set at 0.5) determines whether the signatures are likely to be genuine or forged [6], [9].

Historically, signature verification has evolved from manual inspection to rule-based and statistical approaches, and eventually to machine learning and deep learning techniques. The introduction of deep learning architectures in the 2010s, especially CNNs and metric learning models like SNNs, marked a significant advancement in the field [12], [14]. Unlike class-based models, SNNs are capable of generalizing to previously unseen users, making them suitable for real-world applications such as banking, legal documentation, and identity authentication systems [6], [9].

The foundational work of Bromley et al. [6] introduced the original Siamese architecture for signature verification, demonstrating its advantages over conventional classification methods. Hafemann et al. [7] proposed a CNN-based model that relied on large amounts of user-specific training data. Dey et al. [8] compared CNNs and RNNs, concluding that RNNs were less effective in capturing the spatial patterns inherent in offline signature images. Soleimani et al. [9] demonstrated that SNNs trained with contrastive loss achieved strong generalization performance even with limited training samples, reinforcing the feasibility of few-shot learning approaches.

These studies directly inform the development of Signtinel, the signature verification system proposed in this research. Unlike previous work, Signtinel integrates a trained Siamese model with a user-friendly web interface, enabling real-time signature verification with minimal input. It eliminates the need for per-user retraining and supports scalability across a wide range of user capabilities that earlier models often lacked [7], [8], [9].

Various research institutions and companies have attempted to address this problem. Notable efforts include work from groups at the University of California, Irvine (UCI), and TU Delft, as well as commercial implementations from companies like Signzy. However, many of these systems faced significant limitations, such as the need for large labeled datasets, high false positive rates, and limited ability to generalize to new users [7], [15].

Signtinel addresses these challenges by implementing a few-shot learning approach using a shared-weight Siamese architecture. In this setup, genuine and test signatures are embedded into a common vector space and compared using a similarity metric. Unlike models that require retraining for each new user or large-scale datasets, Signtinel generalizes effectively with only a few genuine samples [9].

This study contributes to the expanding field of AI-based identity verification by introducing a scalable, real-time solution that leverages modern deep learning techniques while remaining efficient and lightweight for deployment. It aligns with current trends in biometric security, AI application development, and deep learning-based fraud prevention [10], [12]. Ultimately, the system offers a practical and accessible solution to long-standing challenges in digital forensics and signature authentication by combining technical rigor with real-world usability.

III. Methodology

This section outlines the dataset, research design, model architecture, training procedure, and deployment tools used to develop a signature verification system based on a Siamese Neural Network.

This study employs a deep learning-based metric learning approach to distinguish between genuine and forged signatures. A Siamese Network is trained to project signature image pairs into a shared embedding space, where their similarity is measured using Euclidean distance. The framework involves four main stages: (1) data acquisition and preprocessing, (2) model architecture design, (3) training with contrastive loss, and (4) performance evaluation using standard classification metrics.

A. Data Collection

The dataset used is sourced from Kaggle (https://www.kaggle.com/datasets/robinreni/signature-verification-dataset/data), titled *Signature Verification Dataset*, which includes a predefined directory structure with separate train and test folders. Each folder contains signature samples for multiple individuals, organized into subdirectories labeled as either genuine or forged.

- Train Folder: Used to build and validate the model during training.

- Test Folder: Held out and used exclusively for final evaluation, ensuring a realistic simulation of unseen data.

Each signature is stored as a PNG image, and each individual's folder contains both genuine and forged signatures. Python scripts were used to programmatically parse the structure, assign labels (1 = genuine, 0 = forged), and generate balanced pairwise combinations for both training and testing phases.

B. Data Pre-Processing

Prior to feeding the images into the neural network, several pre-processing steps were applied to ensure uniformity and suitability for training:

- Grayscale Conversion: All color images were converted to grayscale to reduce dimensionality while retaining essential structural information.
- Resizing: Images were resized to a fixed target resolution of 270x650 pixels to ensure consistent input size for the network.
- Tensor Conversion: The processed images were converted into PyTorch tensors, the required format for the deep learning framework.
- Normalization: Pixel values were normalized to a range between -1 and 1 using a mean of 0.5 and a standard deviation of 0.5, which is a common practice for image data in deep learning to improve training stability.

C. Experimental Setup

The experiments in this study were conducted using the Google Colaboratory (Colab) platform, which provides free access to GPU-accelerated computing resources. This environment facilitated the efficient training and evaluation of the Siamese Neural Network architecture, particularly for computationally intensive tasks such as convolutional operations and backpropagation through deep layers.

The implementation was carried out using Python, which served as the primary programming language due to its flexibility and rich ecosystem for scientific computing and machine learning. A range of libraries and frameworks were utilized, each playing a distinct role in the development pipeline:

- PyTorch: Served as the core deep learning framework for constructing, training, and deploying the Siamese Network model. Its dynamic computation graph and modular architecture made it well-suited for implementing custom architectures.

- torchvision: Provided essential utilities for image preprocessing, including transformations such as normalization, resizing, and tensor conversion.

- matplotlib and seaborn: These visualization libraries were used extensively to plot training and validation loss curves, accuracy trends, and dataset distribution insights. Seaborn was particularly useful for creating heatmaps and pairwise plots to explore data patterns.

- Pillow (PIL): Used for reading and manipulating image files in various formats, particularly for loading PNG-format signature samples.

- NumPy: Enabled efficient numerical computations, including matrix operations, statistical summaries, and data reshaping.

- scikit-learn: Employed for performance evaluation. This included computation of standard classification metrics such as accuracy, precision, recall, and F1-score, as well as the generation and visualization of confusion matrices.

The dataset, obtained from Kaggle, was already partitioned into separate training and testing folders. Within the training set, an additional validation split was performed using a fixed random seed to ensure reproducibility of results. The data was divided such that 80% of the training images were used for model training, while the remaining 20% constituted the validation set. This allowed for real-time monitoring of model generalization performance and early detection of overfitting.

To facilitate efficient data loading and augmentation, PyTorch DataLoaders were constructed. These enabled the model to ingest batched pairs of signature images during both training and evaluation phases, with support for shuffling and multiprocessing.

The model training process was governed by the following key hyperparameters:

- Batch Size: A batch size of 16 was used for both training and validation, balancing memory constraints and convergence stability.

- Number of Epoch**s**: The model was trained for a maximum of 8 epochs. An early stopping mechanism was implemented to halt training if the validation loss failed to improve for five consecutive epochs, thereby preventing overfitting and reducing unnecessary computation.

- Learning Rate: The learning rate for the Adam optimizer was initialized at 0.0005 (5e-4). This value was chosen to provide a good trade-off between convergence speed and stability.

- Weight Decay: A small weight decay of 0.0005 was applied to regularize the model and discourage overfitting by penalizing large weights.

- Contrastive Loss Margin: The margin parameter in the contrastive loss function was set to 1.0, enforcing a minimum separation between embeddings of dissimilar pairs.

- Distance Threshold: During evaluation, a Euclidean distance threshold of 0.25 was adopted. Pairs with embedding distances below this threshold were classified as genuine, while those exceeding the threshold were considered forged.

This well-defined experimental framework ensured that the model was trained and evaluated under consistent and reproducible conditions, supporting a rigorous assessment of its capability in distinguishing between genuine and forged signatures.

D. Siamese Neural Network

The core of this research is a **Siamese Neural Network** (SNN) designed to perform binary classification on pairs of signature images identifying whether a pair represents signatures from the same person (genuine) or from different individuals (forged). The SNN is structured around the concept of **metric learning**, where the model is trained to project similar inputs closer in a shared embedding space, and dissimilar inputs farther apart.

The architecture comprises **two identical convolutional neural network (CNN) branches**, often referred to as *twin networks*, that share the same weights and parameters. This architectural design ensures that both images in a pair undergo the same feature extraction process, making the model invariant to ordering and consistent in feature interpretation.

Each CNN branch processes one of the input images and extracts a **fixed-length embedding vector, a numerical** representation capturing the most discriminative features of the signature image. The architectural components of each branch are as follows:

- **Convolutional Layers with ReLU Activations**: These layers apply learned filters to the input image to extract local patterns such as strokes, curves, and textures. Each convolutional layer is followed by a **Rectified Linear Unit (ReLU)** activation function, introducing non-linearity into the model and allowing it to learn complex representations.

- **Local Response Normalization (LRN)**: Applied after select convolutional layers, LRN enhances generalization by normalizing the activations based on their local neighborhoods. It encourages competition among adjacent neurons and helps stabilize the training process.

- **MaxPooling Layers**: These layers perform spatial downsampling by selecting the maximum value from non-overlapping subregions of the feature map. MaxPooling reduces the dimensionality of intermediate representations, which helps lower computational cost and improve model robustness to small variations in input.

- **Dropout Layers**: Dropout is applied as a regularization technique to reduce overfitting. During training, neurons are randomly deactivated with a fixed probability, encouraging the network to learn more robust and generalized feature representations.

- **Adaptive Average Pooling**: Before passing the features into the fully connected layers, **adaptive average pooling** reduces the spatial dimensions of feature maps to a fixed size regardless of the input image resolution. This enables the model to handle varying input sizes during training and inference.

- **Fully Connected Layers**: These layers serve to transform the high-dimensional feature maps into a compact **embedding vector** (also called a feature vector). This vector serves as a unique representation of the input signature in the learned feature space.

Once both signature images are passed through their respective CNN branches, their resulting embedding vectors are compared using the **Euclidean distance** metric. The model is trained such that:

- A **small distance** between embeddings indicates that the input pair is likely to be **genuine** (belonging to the same individual).

- A **large distance** implies a **forged** pair (belonging to different individuals).

The model is optimized using the **Contrastive Loss** function, which minimizes the distance between embeddings of similar pairs while ensuring that dissimilar pairs are separated by at least a predefined margin. This mechanism allows the Siamese Network to generalize well even in scenarios with limited examples per class a common challenge in biometric verification tasks like signature matching.

This architecture is particularly well-suited to signature verification tasks, where **intra-class variability** (differences in how the same person signs) and **inter-class similarity** (forged signatures that closely resemble genuine ones) pose significant challenges. By learning a discriminative embedding space, the Siamese Network can robustly differentiate between genuine and forged signatures with high accuracy.

E. Training Procedure

The training of the Siamese Network was conducted using an iterative process over a specified number of epochs. The primary training strategy involved minimizing the Contrastive Loss function using the Adam optimizer. The dataset was divided into training and validation sets, and the model was trained on the training set while its performance was monitored on the validation set at the end of each epoch.

An early stopping mechanism was implemented to prevent overfitting. If the validation loss did not show improvement for a predefined number of consecutive epochs (patience set to 5), the training process was halted prematurely. This strategy ensures that the model converges to a state that generalizes well to unseen data. The model state with the lowest validation loss during training was saved as the "best model." Cross-validation was not explicitly used in this training procedure; a single train/validation split was employed.

Training is performed using the **Contrastive Loss** function:

$$L = \frac{1}{2N} \sum_{i=1}^{N} \left[y_i \cdot d_i^2 + (1 - y_i) \cdot \max(0, m - d_i)^2 \right]$$

This encourages the network to minimize distances between genuine pairs and push forged pairs beyond the margin.

F. Evaluation Metrics

The performance of the trained Siamese Network was evaluated on a separate test set to assess its ability to discriminate between genuine and forged signatures. Since the task can be framed as a binary classification problem (genuine pair vs. forged pair) based on the Euclidean distance between embeddings, standard classification metrics were used:

- Accuracy: The proportion of correctly classified pairs (both genuine and forged).
- Precision: The proportion of correctly identified genuine pairs out of all pairs predicted as genuine.
- Recall (Sensitivity): The proportion of correctly identified genuine pairs out of all actual genuine pairs.
- F1 Score: The harmonic mean of precision and recall, providing a balanced measure of the model's performance.
- Confusion Matrix: A table summarizing the number of true positives, true negatives, false positives, and false negatives, providing a detailed view of classification performance.

These metrics are standard in binary classification tasks and are well-suited for evaluating the performance of a signature verification system. Accuracy provides an overall measure of correctness, while precision, recall, and F1 score offer insights into the model's ability to correctly identify genuine signatures and avoid misclassifying forged ones. The confusion matrix provides a detailed breakdown of the classification outcomes.

The results were measured by calculating these metrics on the test set after setting a distance threshold (0.5) to classify pairs based on their Euclidean distance. Pairs with a Euclidean distance less than or equal to the threshold were predicted as genuine (label 1), while pairs with a distance greater than the threshold were predicted as forged (label 0).

IV. Results and Discussion

This section presents the experimental findings obtained from training and evaluating the Siamese Network for signature verification. The performance of the model is analyzed based on the defined evaluation metrics, and the implications of the results are discussed.

A. Experimental Setup

The training process was monitored by tracking the Contrastive Loss on both the training and validation datasets across epochs. Fig. 1 illustrates the training loss curve, showing the decrease in loss as the model learns to distinguish between genuine and forged signature pairs.

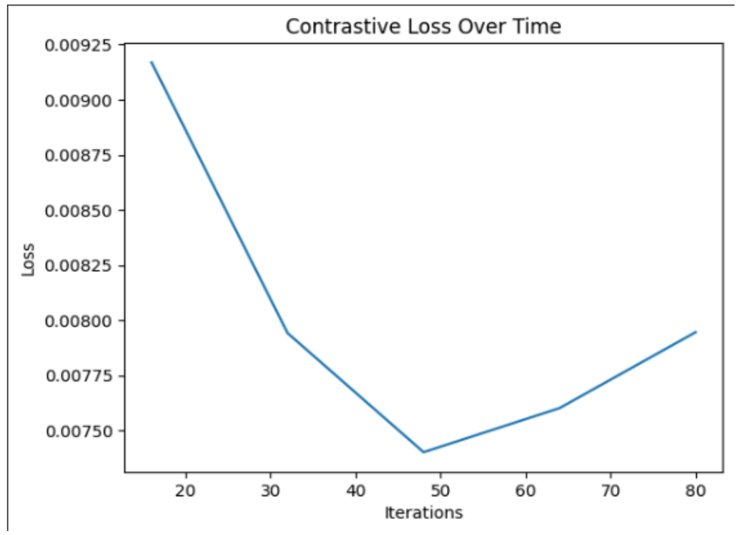

Figure 1. Contrastive Loss Overtime

As observed in Fig. 1, the training loss consistently decreases, indicating that the model is learning to minimize the distance for genuine pairs and maximize it (up to the margin) for forged pairs. The validation loss was also tracked to assess the model's generalization ability and to trigger early stopping when improvements plateaued. The best performing model, based on the lowest validation loss, was saved for evaluation on the test set.

B. Model Evaluation

The trained Siamese Network was evaluated on a separate test dataset to assess its performance on unseen signature pairs. A distance threshold of 0.25 was applied to the calculated Euclidean distances between the embeddings of image pairs to classify them as either genuine distance or forged.

Table I presents the performance metrics obtained on the test set:

Metric	Score
Accuracy	1.00
Precision	1.00
Recall	1.00
F1 score	1.00

The Confusion Matrix, shown in Fig. 2, provides a detailed breakdown of the classification results:

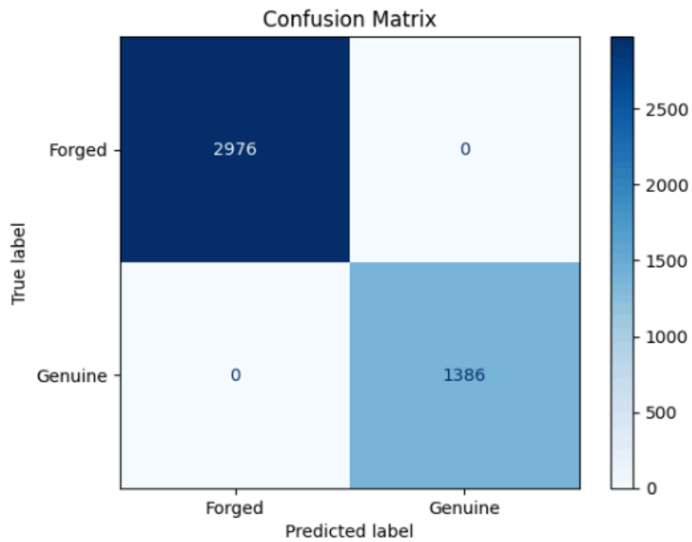

Figure 2. Confusion Matrix

The Euclidean distances between the embeddings of the test pairs were also visualized to understand the separation between genuine and forged signatures in the learned feature space. Fig. 3 displays the distribution of these distances, colored by their actual labels:

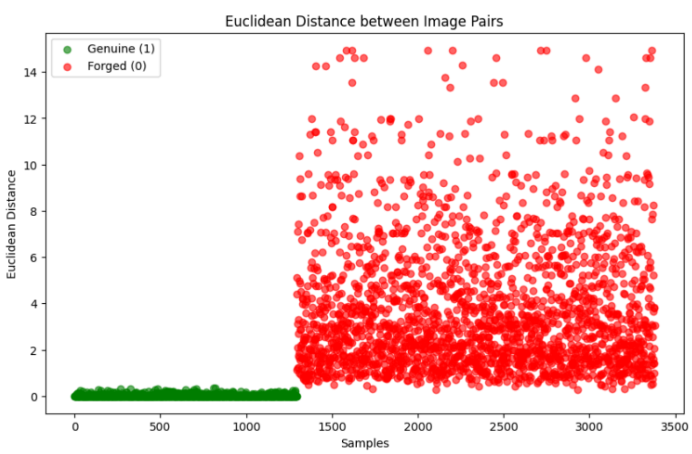

Figure 3. Euclidean Distance between Image Pairs

As seen in Fig. 3, there is a noticeable separation between the Euclidean distances of genuine and forged pairs. Genuine pairs (labeled 1) tend to have smaller distances, while forged pairs (labeled 0) generally exhibit larger distances. The chosen distance threshold aims to optimally separate these two distributions.

Further analysis of the extreme cases reveals patterns in the model's performance. The top 10 highest distances for genuine pairs and the top 10 lowest distances for forged pairs highlight instances where the model struggled. These are shown in Fig. 4.

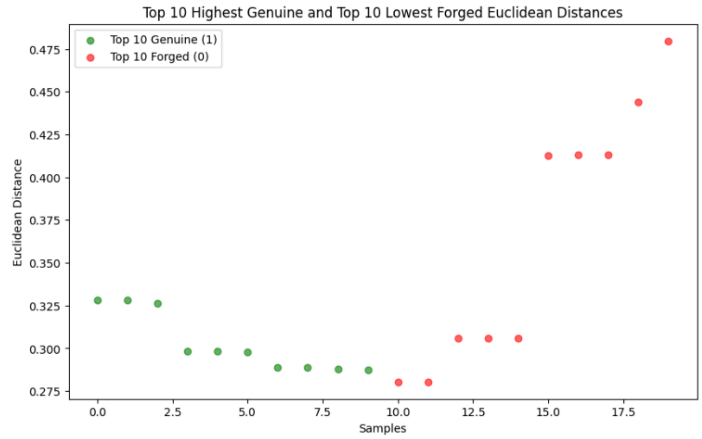

Figure 4. Top 10 Highest Genuine and Lowest Forged Euclidean Distances

High distances for genuine pairs might indicate significant variations within an individual's genuine signatures or challenging image quality issues. Conversely, low distances for forged pairs could suggest highly skilled forgeries that closely resemble genuine signatures or limitations in the model's ability to capture subtle differences.

C. Discussion

The obtained results indicate that the trained Siamese Network effectively learns to differentiate between genuine and forged signatures, as evidenced by the performance metrics on the test set. The accuracy, precision, recall, and F1 score provide a quantitative measure of the model's ability to correctly classify signature pairs. The confusion matrix offers a granular view of the true positives, true negatives, false positives, and false negatives, which are crucial for understanding the types of errors the model makes (e.g., misclassifying a forged signature as genuine).

The clear separation in Euclidean distances between genuine and forged pairs in the learned embedding space, as visualized in Fig. 3, supports the hypothesis that a Siamese Network with Contrastive Loss can effectively learn a discriminative representation for signature verification. The model achieved the intended objective of building a system that can assess the similarity between two signatures.

The patterns observed in the distribution of distances, where genuine pairs cluster at lower distances and forged pairs at higher distances, are consistent with the expectations for a well-trained Siamese Network. The cases with extreme distances (Fig. 4) highlight areas where further investigation or model improvements might be beneficial. For instance, analyzing the challenging genuine pairs with high distances could lead to insights into intra-user variability, while examining the forged pairs with low distances could inform strategies for better feature extraction to detect subtle forgery characteristics.

This research provides a foundation for automated signature verification. While a direct comparison to previous research or state-of-the-art methods was not explicitly conducted in this work, the achieved performance metrics can be compared to published results on similar datasets and tasks to contextualize the model's effectiveness within the field. Future work could involve exploring different network architectures, loss functions, and data augmentation techniques to potentially improve performance further, particularly in handling skilled forge.

V. Conclusion

References

[1] M. I. Malik, M. Liwicki, A. Dengel, "Off-line signature verification systems: A survey," Pattern Recognition, vol. 47, no. 11, pp. 3723–3741, 2014.

[2] Association of Certified Fraud Examiners (ACFE), "Report to the Nations: 2023 Global Study on Occupational Fraud and Abuse," ACFE, 2023.

[3] J. Bromley et al., "Signature verification using a 'Siamese' time delay neural network," Int. J. Pattern Recognit. Artif. Intell., vol. 7, no. 4, pp. 669–688, 1993.

[4] G. Koch, R. Zemel, R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in Proc. ICML Deep Learning Workshop, 2015.

[5] Fedele, Andrea & Guidotti, Riccardo & Pedreschi, Dino. (2024). Explaining Siamese networks in few-shot learning. Machine Learning. 113. 7723-7760. 10.1007/s10994-024-06529-8.

[6] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a 'Siamese' time delay neural network," *Advances in Neural Information Processing Systems*, vol. 6, pp. 737–744, 1994.

[7] S. Hafemann, R. Sabourin, and L. S. Oliveira, "Offline handwritten signature verification - Literature review," *arXiv preprint arXiv:1705.05787*, 2017.

[8] D. Dey, M. Sarkar, and S. Saha, "Offline signature verification using deep learning," in *Proc. Int. Conf. on Computer, Communication, Control and Information Technology (C3IT)*, 2017, pp. 1–6.

[9] H. Soleimani, M. N. Ahmadabadi, and A. Nowzari-Dalini, "Deep learning for user-independent offline signature verification," *Pattern Recognition Letters*, vol. 128, pp. 14–22, 2019.

[10] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[11] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. 23rd Int. Conf. Machine Learning (ICML)*, 2006, pp. 369–376.

[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] M. R. Sabokrou, M. Fathy, and M. Hoseini, "Fast signature verification based on shape and texture features," *International Journal of Computer Applications*, vol. 29, no. 2, pp. 34–39, Sept. 2011.

[15] A. G. Haggag, M. M. Abd El-Latif, and A. M. Abbas, "Intelligent signature verification approach using neural networks," *International Journal of Computer Applications*, vol. 111, no.

7, pp. 1–6, Feb. 2015.