# Enhancement of Siamese Neural Network for Improved Signature Fraud Detection

**Marevil E. Catugas[1], Christelle Joyce M. Cerezo[2], Raymund M. Dioses[3], Khatalyn E. Mata[4]**

[1,2]Student, College of Information System and Technology Management, Pamantasan ng Lungsod ng Maynila, Philippines
[3,4]Professor, College of Information System and Technology Management, Pamantasan ng Lungsod ng Maynila, Philippines

**Abstract**

This study enhances the Siamese Neural Network (SNN) in detecting signature fraud detection by addressing its critical challenges in feature extraction, difficulty handling class imbalances, and computational inefficiency. SMOTE was employed to balance the dataset, optimized training methodologies were applied, and the network architecture was redesigned to improve performance and scalability. Experimental evaluations were conducted using publicly available datasets, CEDAR and BHSig260, under a writer-independent setup, where the model was trained on one group of individuals and tested on unseen writers. The enhanced model demonstrated substantial improvements in performance metrics. The enhanced model achieved significant performance improvements, with accuracy rising from 67.61% to 99.65%, and F1-score from 0.0000 to 0.9944, with ROC-AUC from 0.5000 to 0.9989, The findings highlight the enhanced model's effectiveness in real-world applications, reinforcing public document security associated with signature forgery. This research contributes to the growing field of biometric verification, offering a scalable and adaptable solution tailored for the evolving demands of signature authentication.

**Keywords:** Siamese Neural Network, Fraud Signature Detection, SNN, SMOTE, CEDAR, Biometric Authentication, Machine learning, Writer-Independent Model

## 1. Introduction

Siamese Neural Networks (SNNs) have emerged as a powerful deep learning algorithm designed for tasks requiring pairwise comparisons, such as signature verification. By mapping similar inputs closer together and dissimilar inputs farther apart in a high-dimensional vector space, SNNs enable highly accurate similarity-based analysis. This architecture relies on two identical subnetworks with shared weights that extract features from input pairs, followed by a distance metric to measure their similarity. Its capability to capture intricate patterns and distinguish subtle geometric variations makes SNNs particularly well-suited for detecting forgeries.

One notable implementation of SNNs in signature verification is the SigNet architecture, a convolutional Siamese network proposed by Dey et al. (2017). SigNet has demonstrated exceptional performance in writer-independent offline signature verification by leveraging convolutional layers for effective feature extraction and employing a distance-based comparison for decision-making. Despite its success, challenges such as the imbalance of training data and the sensitivity of key parameters in the model reveal the need for further refinement. Imbalanced datasets, often characterized by a disproportionate representation of genuine and forged signatures, can bias the model towards the majority class, reducing its ability to generalize effectively. Furthermore, the performance of Siamese Neural Networks heavily depends on the careful tuning of hyperparameters, including learning rates and margin thresholds in the distance metric, which can significantly impact the network's robustness and accuracy.

This research builds upon the foundational principles of Siamese Neural Networks and proposes a series of enhancements to improve their performance in real-world signature verification scenarios. The refined model introduces advanced feature extraction techniques to better capture the unique characteristics of signatures, optimized training strategies to reduce computational overhead, and structural modifications to improve the network's robustness against diverse forgery styles. These improvements aim to address the increasing risk of digital forgery, particularly in high-stakes environments such as public document security in Metro Manila, where the integrity of electronic records is crucial for maintaining trust and transparency.

## 2. Related Literature

In the field of cybersecurity, fraud detection is essential in ensuring the security of transactions especially when it comes to verifying signatures. Although digital signatures are frequently used to verify documents and stop illegal access, they are susceptible to fraud, particularly when papers are altered using fake signatures[1]. The goal of fraud detection in signature verification is to differentiate between authentic and fake signatures, which is essential for protecting financial and legal transactions.SNNs, compared with traditional neural networks, are made to compare two inputs by calculating how similar they are [3][17]. The Siamese network is a perfect model for differentiating between authentic and fraudulent signatures because of its structure, which consists of two similar sub-networks that share weights [16]. Offline verification is just as important in situations where signatures are captured in scanned documents or photos, even though online signature verification records real-time attributes like velocity and pressure. In these situations, looking at visual characteristics such as shape, size, and trajectory is necessary to confirm the legitimacy of a signature where it's easy to fraud because it can be easily imitated [4][10][12]. Dealing with unbalanced datasets is one of the most challenging aspects of detecting signature forgery. The dataset typically has a class imbalance that may affect the neural network model performance because there are a lot more authentic signatures than fake ones [9][20] that can lead to overfitting because of the unbalanced datasets [15]. The model might not be able to correctly identify forged signatures if the dataset is skewed toward the majority class [8]. Researchers have been exploring a number of strategies to balance the dataset as a result of this difficulty, including data augmentation techniques [19] like the Synthetic Minority Over-sampling Technique (SMOTE) [2]. To guarantee that the model is trained on a more balanced dataset and increase the precision of forged signature detection, SMOTE creates synthetic samples for the minority class [7]. Siamese neural networks have shown significant performance in offline verification of signatures. SNNs can accurately detect minute distinctions between authentic and fake signatures by learning the similarity between signature pairs. According to recent research, SNNs can beat conventional machine learning models in terms of classification accuracy when paired with pre-processing methods to improve the quality of signature photos [16]. SMOTE, which solves the issue of imbalanced datasets and contributes to the model's increased robustness, has been added to these techniques in order to improve them [18].

## 3. Research Method

### 3.1 Siamese Neural Network Baseline

The SigNet framework, a convolutional Siamese Neural Network (SNN), serves as the baseline architecture for this research. It is designed explicitly for writer-independent offline signature verification, a task known for its high complexity due to subtle variations in handwriting styles and potential forgeries. SigNet operates by embedding signature pairs into a learned feature space, minimizing distances between embeddings of genuine signature pairs while maximizing distances for forged pairs.

The architecture of SigNet is composed of twin convolutional subnetworks with shared weights, ensuring consistent feature extraction across input pairs. Each subnetwork processes an input signature image, producing embeddings that are compared using the Euclidean distance. The shared weights ensure that similar inputs produce similar embeddings, effectively capturing the nuanced details of handwriting. Its architecture leverages a deep convolutional neural network (CNN) for feature extraction. Its configuration is inspired by Krizhevsky et al. (2012) and adapted for signature verification.

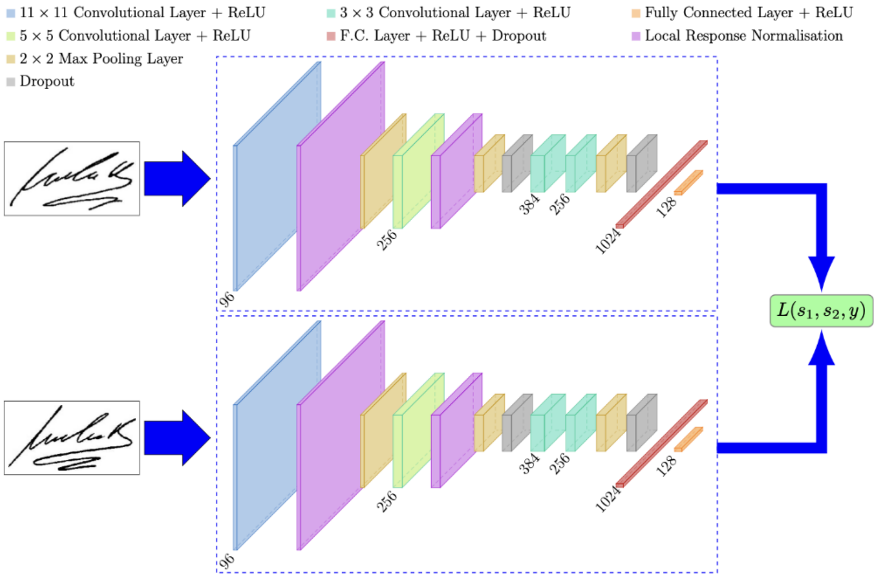

11 × 11 Convolutional Layer + ReLU
5 × 5 Convolutional Layer + ReLU
2 × 2 Max Pooling Layer
Dropout
3 × 3 Convolutional Layer + ReLU
F.C. Layer + ReLU + Dropout
Fully Connected Layer + ReLU
Local Response Normalisation
96
256
384
256
1024
128
96
256
384
256
1024
128
$L(s_1, s_2, y)$

**Figure 1: Architecture of Signet**

The SigNet architecture begins with an input and preprocessing stage, where the input consists of a pair of signature images. These images are resized to 155×220 pixels to ensure uniform dimensions across the dataset. Pixel values are normalized by dividing by the standard deviation of pixel intensities, stabilizing gradient flow and improving convergence during training. This preprocessing stage ensures consistency in input dimensions and scales, preparing the data for feature extraction in the convolutional layers.

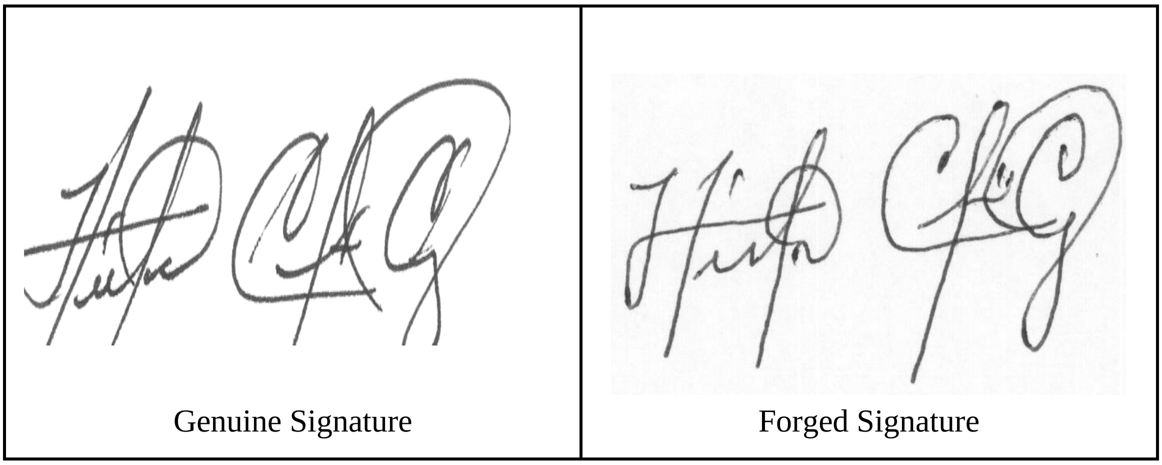
The convolutional layers represent the backbone of the SigNet architecture. These layers employ a twin convolutional network with shared weights to extract hierarchical features from the input signature pair. The first convolutional layer uses 96 filters of size 11×11 with a stride of 4, capturing low-level features such as edges and corners. The second layer consists of 256 filters with 5×5 kernels and padding, focusing on mid-level features. Deeper convolutional layers, including Layers 3 and 4, refine the feature representation using smaller 3×3 kernels, extracting high-level abstractions critical for distinguishing genuine and forged signatures. Pooling layers are applied after specific convolutional layers, reducing the spatial dimensions of feature maps and preventing overfitting by limiting the model's complexity.

Following feature extraction, the network transitions to the fully connected layers. The flattened feature maps are passed to a fully connected layer with 1024 neurons and a dropout rate of 0.5, which reduces the risk of overfitting. The final fully connected layer compresses the representation into a 128-dimensional embedding, which serves as the feature vector for the signature. These embeddings are compared using the Euclidean distance, which quantifies the similarity between the input signature pair. The distance metric is used in conjunction with the contrastive loss function to train the network.

The final step involves optimization, where the network parameters are updated using the RMSprop optimizer with a learning rate of 1e-4 and a batch size of 128. This iterative process enables the model to accurately capture nuanced differences in signature features while remaining robust to variations in handwriting styles and forgery techniques. The output of the network is a similarity score based on the computed distance, with a threshold applied to determine whether the pair belongs to the "genuine" or "forged" category.

### 3.2 Dataset

The experimental evaluation was conducted using CEDAR (Figure 1), a publicly available offline signature dataset. In this study, a 70:30 train-test split was adapted for each dataset, diverging from the dataset-specific splits utilized in the original SigNet paper.



Genuine Signature

Forged Signature

**Figure 2: Sample of CEDAR Dataset**

The CEDAR dataset consists of 55 writers, each contributing 24 genuine and 24 forged signatures. Following the 70:30 split, the training set includes signatures from 39 writers, while the remaining 16 writers are reserved for testing. The dataset is widely used in signature verification research and includes both skilled and random forgeries.

The datasets used in the study were organized to facilitate easier integration into the training pipeline. No additional manual preprocessing, such as cropping or resizing, was applied to the images to maintain consistency with the original dataset characteristics. Instead, the focus was on structuring and organizing the datasets for improved usability and readability.

### 3.3 Pseudocode of Enhanced Siamese Neural Network

**Load Images**

3.3.1. Load signature images img1 and img2 using a custom image loader.

3.3.2. Ensure images have the same dimensions through resizing

**Preprocess Images**

3.3.3. Convert images to grayscale for consistency.

3.3.4. Normalize pixel values by dividing by the dataset's pixel standard deviation.

3.3.5. Invert pixel values so the background has a value of 0.

**Apply SMOTE for Class Imbalance**

3.3.6. Flatten image features

3.3.7. Separate the dataset into features (X) and labels (y) for genuine and forged classes.

3.3.8. Use SMOTE to generate synthetic samples for the minority class, resulting in a balanced dataset.

3.3.9 Reconstruct the balanced dataset by pairing genuine and forged signatures to form (img1, img2) pairs.

**Define CNN Architecture**

3.3.10. Construct a convolutional network:

Layer 1: Apply 96 filters of size 11x11 with ReLU activation.

Layer 2: Use 256 filters of size 5x5 with max pooling and dropout.

Layer 3: Add 384 filters of size 3x3.

Layer 4: Add 256 filters of size 3×3.

**Output Layer: Fully connected layers reduce features to 128-dimensional embeddings.**

**Build Siamese Network**

3.3.11. Define two identical CNN branches sharing weights.

3.3.12. Input img1 and img2 to the twin networks.

3.3.13. Compute feature embeddings from both branches.

**Define Loss Function**

3.3.14. Calculate the Euclidean distance between feature vectors.

3.3.15. Apply Contrastive Loss

**Train the Network**

3.3.16. Use the balanced pairs (from Step 3.4) for training.

3.3.17. Initialize parameters using Xavier Initialization.

3.3.18. Use the RMSprop optimizer with a learning rate of 1e-4 and momentum 0.9.

3.3.19. Train with a batch size of 32 over 20 epochs.

**Verify Signatures**

3.3.20. Input new signature pairs img1,img2.

3.3.21. Compute distance D between their embeddings.

3.3.22. Compare D with a threshold to classify pairs

**Evaluate Performance**

3.3.23. Compute metrics: True Positive Rate (TPR), True Negative Rate (TNR), Accuracy, Precision, Recall (Genuine Acceptance Rate or GAR), F1-Score, ROC-AUC, and False Rejection Rate (FRR).

3.3.24. Maximize accuracy

**Display Results**

3.3.25. Visualize detected matches and mismatches using embedding distances.

3.3.26. Display performance metrics for evaluation.

### 3.4 Framework of Enhanced Siamese Neural Network

This study adopts an experimental and developmental approach to enhance the Siamese Neural Network (SNN) for signature verification. The framework is designed to address key challenges in SNN applications, such as class imbalance, computational inefficiency, and scalability, which are critical obstacles to deploying SNNs in real-world tasks of signature fraud detection. The enhanced Siamese Neural Network is structured to tackle these challenges through a multi-step approach.

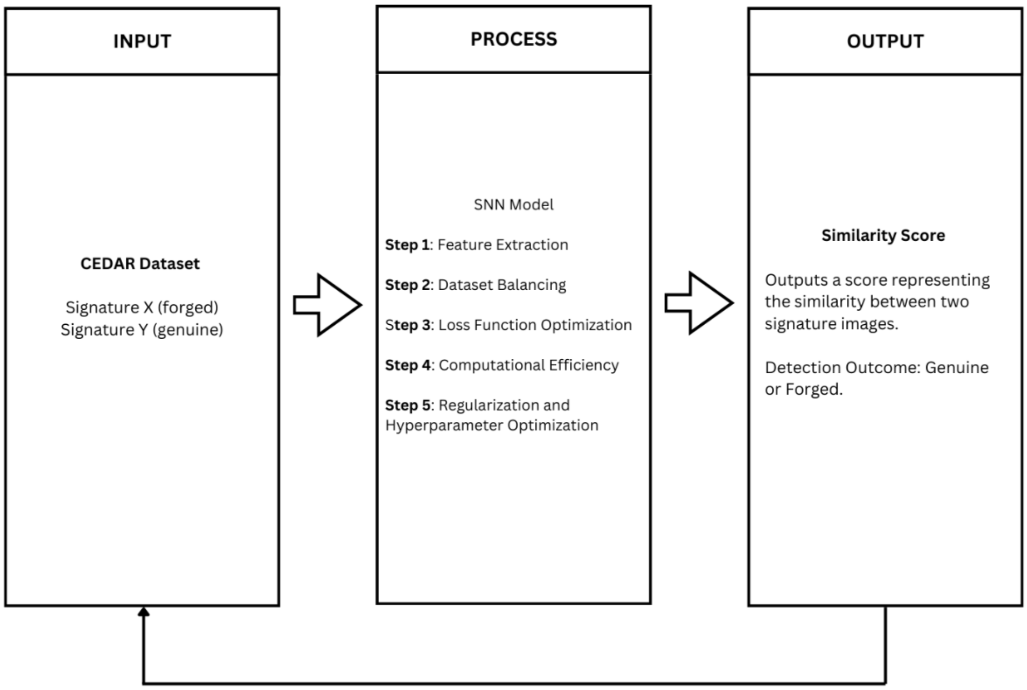

INPUT
CEDAR Dataset
Signature X (forged)
Signature Y (genuine)
PROCESS
SNN Model
Step 1: Feature Extraction
Step 2: Dataset Balancing
Step 3: Loss Function Optimization
Step 4: Computational Efficiency
Step 5: Regularization and Hyperparameter Optimization
OUTPUT
Similarity Score
Outputs a score representing the similarity between two signature images.
Detection Outcome: Genuine or Forged.

**Figure 3: Siamese Neural Network Operational Framework**

First, the network extracts embeddings from input signature pairs, generating robust feature representations. To address the class imbalance between genuine and forged samples, the Synthetic Minority Oversampling Technique (SMOTE) is applied, ensuring balanced training data and equitable learning across classes. Next, a contrastive loss function is employed to optimize the separation between embeddings of genuine and forged signatures, thereby improving classification accuracy. Computational efficiency is enhanced through batch-wise sampling strategies and optimized distance calculations, making the network scalable for large datasets. Finally, hyperparameter optimization for learning rates and batch sizes, prevent overfitting and improve the model’s generalization.

### 3.4 Metrics for Evaluation

To evaluate the performance of the enhanced Siamese Neural Network, the following metrics were used

1. Accuracy - Represents the overall correctness of the model, calculated as the percentage of correctly classified pairs, including both genuine and forged signatures.
2. Precision - Measures the model’s ability to correctly identify forged signatures, defined as the ratio of true positives (correctly identified forged signatures) to the total number of predicted positives (true

and false positives). This metric evaluates the model's reliability in detecting forged signatures while minimizing false alarms.

3. Recall (Genuine Acceptance Rate - GAR) - Assesses the model's effectiveness in correctly identifying genuine signatures. It is computed as the ratio of true positives (correctly identified genuine signatures) to the total number of actual positives (all genuine signatures). GAR reflects the model's ability to accept authentic signatures accurately.
4. F1-Score - Provides a balanced measure of the model's performance by calculating the harmonic mean of precision and recall. This metric is particularly useful when dealing with imbalanced datasets, as it accounts for both false positives and false negatives.
5. ROC-AUC - Evaluates the model's discriminatory power by measuring the area under the ROC curve. This metric assesses the model's ability to distinguish between genuine and forged signatures across various threshold values, with higher values indicating better performance.
6. False Rejection Rate (FRR) - Represents the proportion of genuine signatures incorrectly classified as forged. It is computed as the complement of GAR (FRR = 1 - GAR). A lower FRR indicates the model's reliability in accepting authentic signatures without unnecessary rejection.

## Results

**Table 1: Comparison of the Performance of the Enhanced Siamese Neural Network**

| | **Original** | **Enhanced** |
|---|---|---|
| **Accuracy** | 0.6761 | 0.9964 |
| **Precision** | 0.0000 | 1.0000 |
| **Recall (GAR)** | 0.0000 | 0.9889 |
| **F1-Score** | 0.0000 | 0.9944 |
| **ROC-AUC** | 0.5000 | 0.9989 |
| **False Rejection Rate (FRR)** | 1.0000 | 0.0111 |

The enhanced Siamese Neural Network (SNN) demonstrates a significant improvement in performance metrics compared to the original model (Table 1), particularly in addressing class imbalance in the dataset. Without Synthetic Minority Oversampling Technique (SMOTE), the model struggled to identify forged signatures due to an inherent imbalance, with genuine samples outnumbering forged ones by approximately 2:1. The inclusion of SMOTE successfully balanced the dataset, increasing the number of forged samples and mitigating bias. As a result, the enhanced model achieved a remarkable accuracy of 99.64%, compared to 67.61% in the original model. Precision and recall also improved drastically, with the enhanced model reaching perfect scores of 1.00 and 0.9889, respectively, and an F1-score of 0.9944. The ROC-AUC metric rose from 0.5000 in the original model to 0.9989, demonstrating a near-perfect ability to distinguish between genuine and forged signatures. Furthermore, the False Rejection Rate (FRR) dropped significantly from 1.0000 to 0.0111, showcasing enhanced reliability. However, these improvements came at the cost of increased training time, which rose from 2988.22 seconds to 3100.87 seconds, highlighting a trade-off between computational efficiency and performance.

## Conclusion

This study addressed the growing concern of signature fraud by enhancing Siamese Neural Networks (SNNs) to improve accuracy and reliability in offline signature verification. Recognizing the critical need for good systems in safeguarding public records and official documents in Metro Manila, this research tackled the existing class that hinders the algorithm's performance. The proposed enhancements, balanced training using Synthetic Minority Oversampling Technique (SMOTE) and architectural refinements, led to substantial improvements across all performance metrics. The enhanced model achieved a near-perfect accuracy of 99.64%, with a precision of 1.0000 and a recall (Genuine Acceptance Rate) of 98.89%, demonstrating exceptional capability in distinguishing genuine signatures from forgeries. And with this, the False Rejection Rate dropped to 0.0111, ensuring the model reliably accepted genuine signatures while minimizing errors. These results give sight to the potential of the enhanced model to provide reliable and secure signature verification for critical applications.

## References

1. Anghel, L. P., Radulescu D., Marinescu I. A. (2023). Effective solutions to prevent digital fraud by introducing electronic signature of PDF files | IEEE Conference Publication | IEEE Xplore.
2. Chawla N. V., Bowyer K. W., Hall, L. O., Kegelmeyer W. P. (2020). View of SMOTE: Synthetic Minority Over-sampling Technique.
3. De Rosa, G. J., & Papa, J. P. (2022). Learning to weight similarity measures with Siamese networks: a case study on optimum-path forest. In Elsevier eBooks (pp. 155–173).
4. Dey, S., Dutta, A., Toledo, J. I., & Pal, U. (2019). SIGNET: Convolutional Siamese Network for writer Independent Offline Signature Verification.
5. Ferro, M. V., Mosquera, Y. D., Ribadas-Pena, F. J., & Bilbao, V. M. D. (2023). Early stopping by correlating online indicators in neural networks. Neural Networks, 159, 109–124.
6. Heroza, R. I., Gan, J. Q., & Raza, H. (2023). SIA-SMOTE: A SMOTE-Based Oversampling Method with Better Interpolation on High-Dimensional Data by Using a Siamese Network. In Lecture notes in computer science (pp. 448–460).
7. Kadlag A., Ingole, A. B., Patil, K. P. (2022). Novel Approach to Offline Signature Classification and Verification System, 3(6), 736-740.
8. Kalaivani N., Beena R. (2021). Improved SMOTE and Optimized Siamese Neural Networks for Class Imbalanced Heterogeneous Cross Project Defect Prediction. | International Journal of Intelligent Engineering & System.
9. Khaneja, A. (2021, December 11). Using Siamese Networks with Unbalanced Data - Ayush Khaneja - Medium.
10. Li, M., Chang, K., Bearce, B., Chang, C. Y., Huang, A. J., Campbell, J. P., Brown, J. M., Singh, P., Hoebel, K., Erdoğmuş, D., Ioannidis, S., Palmer, W. E., Chiang, M. F., & Kalpathy–Cramer, J. (2020). Siamese neural networks for continuous disease severity evaluation and change detection in medical imaging. Npj Digital Medicine, 3(1).
11. Li Y., Chen P. C. L., Zhang T. (2022). A Survey on Siamese Network: Methodologies, Applications and Opportunities (pp. (99): 1-21).
12. Melekhov, I., Kannala, J., & Rahtu, E. (2019). Siamese network features for image matching. In 2016 23rd International Conference on Pattern Recognition (ICPR) (pp. 378-383).
13. Prechelt, L. (2019). Automatic early stopping using cross validation: quantifying the criteria. Neural Networks, 11(4), 761–767.
14. Prechelt, L. (2020). Early Stopping-But when? | Semantic Scholar. Neural Networks
15. Salman, S., & Liu, X. (2019, January 19). Overfitting mechanism and avoidance in deep neural networks. arXiv.org.

16. Sharma, N., Gupta, S., Mohamed, H. G., Anand, D., Mazón, J. L. V., Gupta, D., & Goyal, N. (2022). Siamese Convolutional Neural Network-Based twin Structure model for independent offline signature verification. Sustainability, 14(18), 11484.
17. Vijayakumar, T. (2022). Verification-System-for-Handwritten-Signatures-with-Modular-Neural-Netowrks.pdf
18. Xiao W., And Di W. (2021). An Improved Siamese Network Model for Handwritten Signature Verification | IEEE Conference Publication | IEEE Xplore.
19. Ying, X. (2019). An Overview of Overfitting and its Solutions. Journal of Physics. Conference Series, 1168, 022022.
20. Zhao, L., Shang, Z., Tan, J., Zhou, M., Zhang, M., Gu, D., Zhang, T., & Tang, Y. Y. (2022). Siamese networks with an online reweighted example for imbalanced data learning. Pattern Recognition, 132, 108947.