Handwritten Signature Verification System with Siamese Neural Networks.

Project Midterm Report

Team Members: Lucas Sorge, Kenny Jia Hui Leong

Abstract

This project presents a handwritten signature verification system based on a Siamese Neural Network. Traditional signature verification models typically require many labeled samples per user, which is impractical in real-world applications. We frame the task as a one-shot learning problem to address this, enabling verification with only a single reference signature.

Our Siamese architecture consists of twin convolutional networks that learn a shared embedding space and a similarity function to compare signature pairs. The model was trained and evaluated using a publicly available genuine and forged signatures dataset. Experimental results show that the system effectively distinguishes between authentic and forged signatures, achieving high accuracy with minimal data per user.

The ability to generalize to unseen individuals without retraining makes this approach well-suited for scalable and secure identity verification in financial and administrative contexts.

1. Introduction

Given the significance of signatures, a well-forged one can be detrimental to individuals when used against them in banking, legal, or financial activities. To address this, we propose the development of a system that verifies handwritten signatures using Siamese neural networks (SNN). We plan to train the model by providing tasks such as identifying real and forged signatures. Our dataset is obtained through Kaggle and contains over 5000 handwritten signatures of both real and forged. The distribution of real and forged signatures is roughly equal, approximately 2,900 real and 2,700 forged signatures.

We have chosen SNN as our learning algorithm due to its ability to compare a new signature against a stored reference with high accuracy, even if only one sample is available (Koch et al., 2015). We expect SNN to provide the best possible accuracy compared to traditional methods such as Linear Perceptron, Linear Regression, SVM, and Random Forest in signature verification.

2. Related Work

Our project builds on foundational research in one-shot learning and Siamese Neural Networks. A particularly influential work in this area is *Siamese Neural Networks for One-shot Image Recognition* by Koch et al. (2015), which introduced a novel deep learning architecture designed to generalize across classes with minimal labeled data. Rather than approaching classification as a traditional supervised task, their model learns a similarity function determining whether two input images belong to the same class. This is achieved through a Siamese network composed of two identical convolutional neural networks with shared weights, trained jointly using a contrastive loss function.

A key contribution of their work is demonstrating that such a network can be effectively reused for unseen classes without retraining, enabling powerful generalization in low-data scenarios. Their experiments on the Omniglot dataset showed that Siamese networks could achieve competitive one-shot classification performance, approaching human-level accuracy.

Inspired by this architecture, our project applies a similar approach to the task of handwritten signature verification. As in Koch et al., we formulate the problem as a pairwise verification task, where the model learns to predict whether two signatures are from the same individual. This formulation allows the system to scale efficiently to new users, since only a single reference signature is required, and no retraining is needed when enrolling new identities—a major advantage in real-world authentication applications.

3. Method: Siamese Neural Network for Signature Verification

3.1 Dataset and Task Formulation

The objective of our project is to determine whether two handwritten signatures belong to the same individual. We approach this as a pairwise binary classification task, in which the model receives two signature images and predicts whether they originate from the same person (label 0) or from different individuals (label 1). For this purpose, we use a publicly available signature verification dataset containing 2913 genuine and 2713 forged signatures collected from multiple writers. Each signature is labeled as real or forged and linked to a specific writer ID, which enables the construction of meaningful training and evaluation pairs.

To train the Siamese Neural Network, we generate two types of input pairs: positive pairs, consisting of two genuine signatures from the same individual, and negative pairs, composed of one genuine and one forged signature, typically from different individuals. This setup allows the model to learn a similarity function rather than memorizing individual identities, making it particularly suitable for one-shot learning scenarios where limited data per user is available. The

dataset is divided into 60% for training, 20% for validation, and 20% for testing, ensuring that the model is evaluated on unseen signature pairs for a realistic assessment of generalization.

3.2 Preprocessing

To ensure consistency and enhance model performance, we applied several preprocessing steps to the signature images before training. First, all images were converted to grayscale, as color information is irrelevant for handwritten signatures. We then resized the images to 128×128 pixels to standardize input dimensions and reduce computational cost.

Pixel values were normalized to the range [−1, 1], which improves training stability and convergence when using activation functions. Although not yet implemented in this phase, we plan to incorporate data augmentation techniques—such as rotation, translation, and scaling—in future iterations to simulate natural variations in writing and improve generalization.

To maintain robustness, the preprocessing pipeline also includes error handling for corrupted or invalid image files. Any unreadable images are automatically skipped during training and evaluation, ensuring that only valid data contributes to the learning process.

3.3 Model Architecture

We implemented a Siamese Neural Network, an architecture specifically designed to learn a similarity function between pairs of inputs rather than performing direct classification. The core idea is to map input images into a shared feature space, where similar inputs (i.e., signatures from the same individual) are closer together, and dissimilar ones are farther apart.

Our model consists of two identical convolutional neural networks (CNNs) that share weights. This shared architecture ensures that both inputs are processed in the same way, promoting symmetry and consistency in feature extraction. Each CNN is followed by fully connected layers that produce a fixed-length embedding vector of 128 dimensions, representing the core features of the input signature.

The final output of the Siamese network is the Euclidean distance between the two embeddings. During training, this distance is passed to a binary classification loss function—such as a sigmoid with binary cross-entropy or a contrastive loss—to encourage smaller distances for genuine signature pairs and larger distances for forged pairs.

We also experimented with variations in the encoder’s depth and structure to explore the effect of network complexity on performance.

3.4 Loss Function and Optimization

To train the Siamese Neural Network, we employed the Binary Cross-Entropy with Logits Loss (BCEWithLogitsLoss). This loss function was applied to the Euclidean distance between the two signature embeddings, effectively treating the distance as a proxy for similarity. The goal is to minimize the loss for genuine signature pairs (label 0) by reducing their distance and to increase the loss for forged pairs (label 1) by encouraging greater separation in the embedding space.

The network was trained using the Adam optimizer with a learning rate 0.0001. Training was performed over 10 epochs with a batch size of 32, which provided a good balance between memory usage and convergence speed. No learning rate scheduling or regularization techniques were applied at this stage.

The training loop iterates over pairs of signature images, computing the embedding distance for each pair, calculating the binary cross-entropy loss, and updating model weights through backpropagation. The model's performance was monitored through the average loss per epoch, which consistently decreased throughout training, indicating effective learning of the similarity function.

Although more advanced optimization techniques (e.g., learning rate scheduling, contrastive loss, or hard negative mining) were not yet explored, they are planned for future iterations to further improve verification performance.

3.5 Evaluation Metrics

To evaluate the performance of our Siamese Neural Network, we relied on a combination of quantitative metrics that collectively reflect the model's ability to differentiate between genuine and forged signatures. The primary metric used was accuracy, which represents the overall proportion of correctly classified signature pairs. While useful as a general performance indicator, accuracy alone may be insufficient when class distributions are imbalanced.

To gain a deeper understanding of the model's behavior, we also examined the confusion matrix, which details the number of true positives, true negatives, false positives, and false negatives. This allowed us to assess whether the model was favoring one class over the other. Additionally, we plotted the Receiver Operating Characteristic (ROC) curve and computed the Area Under the Curve (AUC) to evaluate how well the model distinguishes between classes across varying similarity thresholds. A higher AUC indicates stronger discriminative capability.

We further generated a classification report, which includes precision, recall, and F1-score for both genuine and forged classes. These metrics provide a more nuanced view of the model's sensitivity to misclassifications. Overall, the evaluation results confirmed that our model reliably

distinguishes between signature pairs and generalizes well, even when only a single reference signature is available per individual.

3.6 Summary

The Siamese Neural Network architecture proved to be a strong fit for the one-shot learning nature of the signature verification task. By learning a distance-based similarity function rather than relying on traditional classification, the model is capable of verifying identities with only a single reference signature. Its ability to generalize to unseen users without requiring retraining makes it particularly attractive for real-world authentication systems that demand scalability and low data requirements.

4. Preliminary Results

Train-Validation-Test: 60-20-20

Algorithm	Accuracy
Linear Perceptron	0.53
Logistic Regression	0.57
SVM (Linear)	0.59
SVM (Poly)	0.63
SVM (RBF)	0.69
Random Forest	0.66
Siamese Neural Network	**0.68**

More detailed result for top two algorithm accuracies:

Algorithm	Class	Precision	Recall	F1-Score
Siamese Neural Network (SNN)	Real (0)	0.68	0.67	0.68
	Forged (1)	0.68	0.69	0.68
	Macro avg.	0.68	0.68	0.68
	Weighted avg.	0.68	0.68	0.68

SVM (RBF)	Real (0)	0.68	0.75	0.71
	Forged (1)	0.70	0.63	0.66
	Macro avg.	0.69	0.69	0.69
	Weighted avg.	0.69	0.69	0.69

Based on our experimental results, we observed that SVM with RBF kernel achieved the highest accuracy (69%), just slightly higher than Siamese Neural Network (SNN) (68%). Despite our initial predictions, both algorithms perform similarly at classifying real and forged signatures (around 68% of the time). SNN performs better at detecting forged signatures, achieving a higher recall (69%) for class “Forged (1)”, meaning it identifies fraudulent signatures more effectively. On the other hand, SVM is better at detecting real signatures, achieving a higher recall (75%) for class “Real (0)”, meaning it identifies genuine signatures more effectively. A probable reason for this could be due to the usage of Histogram of Oriented Gradients (HOG) for feature extraction in SVM, whereas no optimization was applied for SNN, potentially limiting its ability to extract features effectively.

5. Future plan

For future work, we aim to improve the accuracy and generalization of our Siamese Neural Network through a series of optimizations. Planned enhancements include incorporating learning rate scheduling to improve convergence, replacing the current loss function with contrastive loss for more effective distance learning, and implementing hard negative mining to challenge the model during training. We also intend to explore better weight initialization strategies, perform systematic hyperparameter optimization, and apply affine distortions as data augmentation to simulate natural variations in handwriting. These improvements are expected to significantly enhance the model's robustness and overall verification performance.

6. References

Koch, G., Zemel, R., Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. ICML deep learning workshop, vol. 2., https://paperswithcode.com/paper/siamese-neural networks-for-one-shot-image

Pashkin, Zack. Image dataset for handwritten signature verification. Kaggle. https://www.kaggle.com/datasets/tienen/handwritten-signature-verification/data