Signature Forgery Detection Using Generative Adversarial Networks (GANs)

Saja Moussa

October 16, 2025

Abstract

This report presents the ongoing development of a signature forgery detection system using Generative Adversarial Networks (GANs). The project aims to generate synthetic (fake) signature samples using a GAN trained on genuine signatures, and then leverage these generated samples to train a binary classifier capable of distinguishing real from forged signatures. The work to date includes dataset preparation, preprocessing, GAN training, and performance analysis. Future work will focus on designing more robust forgery detection strategies and implementing a user-facing demonstration system.

1 Introduction

Signature verification is a key task in biometric authentication and fraud prevention. However, collecting large and balanced datasets of genuine and forged signatures is often difficult. To address this issue, this project uses a **Generative Adversarial Network (GAN)** to generate realistic synthetic forgeries that supplement limited human-forged data and improve classifier training.

The system consists of three main phases:

1. Training a GAN on genuine signatures to learn their visual distribution.
2. Generating synthetic signatures to act as forged samples.
3. Training a convolutional neural network (CNN) to classify genuine and forged signatures.

This combination of generative and discriminative models aims to enhance the accuracy and robustness of signature forgery detection.

2 Dataset Preparation

The dataset consists of 2,500 images for both genuine and forged signatures. These were split into training and validation subsets as follows:

The data organization ensures balanced subsets for both classes, facilitating consistent GAN and classifier training later on.

Table 1: Dataset Split for Genuine and Forged Signatures

Class	Train	Validation
Genuine	1920	480
Forged	1920	480

3 Data Preprocessing

To ensure consistency and improve model performance, all signature images underwent a standardized preprocessing pipeline. Each image was first converted to grayscale to reduce input dimensionality while preserving essential texture information. Images were then resized to 128×128 pixels for the GAN and 64×64 pixels for the classification model.

Subsequently, pixel intensity values were normalized to the ranges $[-1, 1]$ for GAN training and $[0, 1]$ for the classifier, ensuring stable gradient behavior during optimization. The dataset was organized into distinct training (80%) and validation (20%) subsets for both genuine and forged signatures, maintaining class balance throughout all experiments.

4 GAN Training

The Generative Adversarial Network (GAN) was trained to synthesize realistic-looking handwritten signatures by learning the distribution of genuine samples. The implementation is based on the **Wasserstein GAN with Gradient Penalty (WGAN-GP)** framework, which ensures more stable convergence compared to conventional GANs by enforcing a smooth Lipschitz constraint on the critic's gradients.

The architecture consists of two components: **(1) Generator** – a convolutional network that transforms a 100-dimensional latent vector into a 128×128 grayscale image using successive upsampling and convolutional layers; and **(2) Critic** – a convolutional discriminator that assigns scalar authenticity scores to real and fake samples without using a sigmoid output.

Both networks were optimized using the **Adam** optimizer with a learning rate of 1×10^{-4} and momentum parameters $\beta_1 = 0.0$ and $\beta_2 = 0.9$. The gradient penalty coefficient λ_{GP} was set to 10. The critic was updated five times per generator iteration to maintain training stability. Training was performed on GPU when available, with a batch size of 64 and a total of 1000 epochs.

The generator and critic losses evolved progressively as shown in Figure 1, which presents the loss curves across epochs. As training advanced, the generator began to produce visually structured strokes and realistic signature shapes, though some fluctuations persisted in later epochs due to dataset size limitations and computational constraints.

To visualize the qualitative improvement, Figures 2 illustrate generated samples at different stages of training. Each figure shows 20 synthesized signatures (normalized and resized to 128×128). Due to the normalization to $[-1, 1]$ and grayscale conversion, slight blurriness or loss of fine detail is expected.

The overall training progression is summarized in Table 2. Results show that the discriminator initially dominates, but the generator gradually improves, producing visually coherent and diverse signatures before partial instability appears in the final epochs.

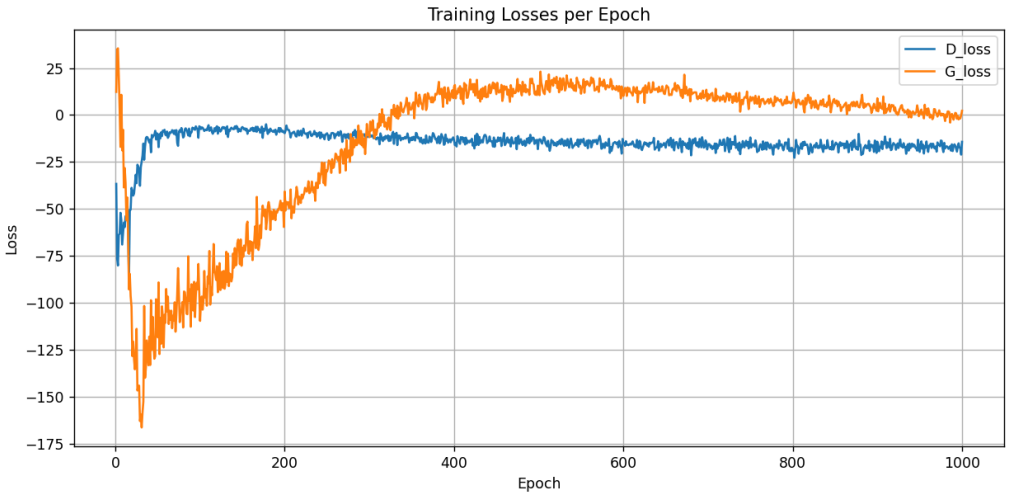
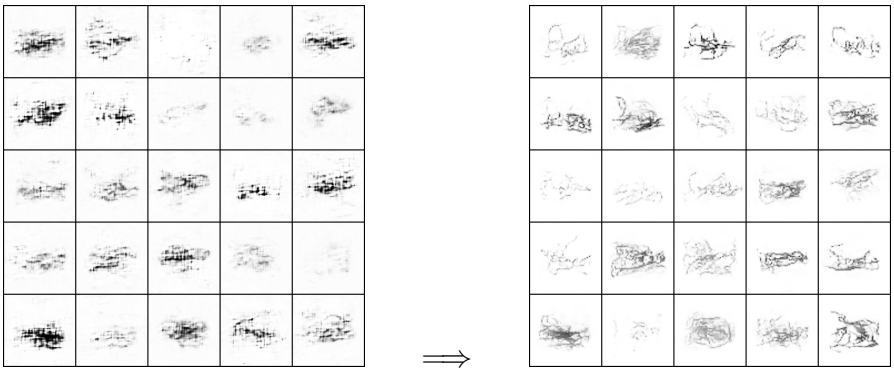


Figure 1: Generator and Critic Losses per Epoch over 1000 Epochs.



After 50 Epochs ⟹ After 1000 Epochs

Figure 2: Evolution of Generated Signature Samples from 50 to 1000 Epochs.

Despite these constraints, the GAN successfully learned meaningful visual patterns from the dataset, producing progressively more realistic signature-like outputs over time.

5 Signature Classification Model

5.1 Model Architecture and Training

A convolutional neural network (CNN) was trained to classify genuine and forged signatures. The network architecture consists of four convolutional layers with ReLU activation and max pooling, followed by two fully connected layers and a sigmoid output unit for binary classification. The model was trained using the **Binary Cross-Entropy Loss** function and the **Adam optimizer** with a learning rate of 1×10^{-3} over 30 epochs.

Data augmentation (random horizontal flipping) and normalization were applied to improve generalization. The model was implemented and trained using **PyTorch** and executed on GPU when available.

Table 2: Summary of WGAN-GP Training Progression (1–1000 Epochs)

Phase	**Epoch Range**	**Avg D Loss**	**Avg G Loss**	**Observations**
Initialization	1–100	≈ -45	≈ -80	Critic dominates; generator outputs mostly noise.
Early Learning	101–300	≈ -15	≈ -40	Structured stroke patterns begin to form.
Mid Training	301–600	≈ -13	≈ -20	Balanced learning; signatures become clearer.
Convergence	601–800	≈ -14	$\approx +5$	Realistic textures appear; minor oscillations.
Late Instability	801–1000	≈ -17	≈ 0	Slight collapse due to limited data and compute power.

5.2 Evaluation Metrics

After training, the classifier was evaluated on the test dataset. Table 3 summarizes the model's performance metrics based on the confusion matrix and classification report.

Table 3: CNN Classifier Evaluation Results on Test Dataset

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Forge	0.77	0.88	0.82	2100
Genuine	0.71	0.54	0.61	1200
Accuracy		0.75 (3300 samples)		
Macro Avg	0.74	0.71	0.72	3300
Weighted Avg	0.75	0.75	0.74	3300

The confusion matrix in Equation 1 illustrates that most forged signatures were correctly identified, while genuine ones were more challenging to classify.

$$\begin{bmatrix} 1842 & 258 \\ 558 & 642 \end{bmatrix} \tag{1}$$

Overall, the model achieved a classification accuracy of approximately **75%**, indicating that the CNN successfully learned discriminative visual features related to handwriting style and stroke characteristics.

5.3 Evaluation on GAN-Generated Images

The trained classifier was then used to assess synthetic signatures generated by the GAN. A total of 500 GAN-generated images were evaluated, with results shown in Table 4.

These results show that the classifier predicted approximately half of the generated samples as forgeries, suggesting that the GAN still produces partially unrealistic features distinguishable by the CNN.

Table 4: CNN Evaluation on GAN-Generated Signatures

Metric	Count	Percentage
Predicted as Genuine	223	44.60%
Predicted as Forged	277	55.40%

6 Discussion and Limitations

While the developed system demonstrates promising results, several limitations must be noted:

- **Dataset Size:** The number of available signature samples was relatively limited. This constraint affects both the diversity of genuine samples and the generalization ability of the model.
- **GAN Quality:** The generated signatures, although visually realistic, still lack the finer texture and stroke irregularities observed in real handwriting, which affects classifier evaluation.
- **Hardware Constraints:** Training was conducted using a student workstation with limited computational power (CPU and low-memory GPU). This restricted model complexity and prolonged training time.
- **Cloud Resources:** Virtual Private Cloud (VPC) usage was limited, reducing the possibility of scaling experiments or testing larger architectures.

Despite these limitations, the combined GAN–CNN pipeline successfully demonstrates an integrated generative and discriminative framework for signature forgery detection. With more extensive datasets and improved computing resources, higher accuracy and stronger generalization could be achieved.

7 Conclusion

This work implemented a complete end-to-end system for signature forgery detection, combining GAN-based signature synthesis with CNN-based classification. The classifier achieved a 75% test accuracy and was able to reasonably distinguish between real and synthetic signatures. The approach illustrates the potential of combining generative and discriminative models for biometric authentication applications.