Signature Verification System

Model Development & Database Design Report

1. Overview

This project implements a robust signature verification system based on a Siamese Neural Network (Contrastive Loss). It features a preprocessing pipeline for normalizing signature images, evaluation modules, and a database-backed identity system. The system achieves **92% test accuracy** and performs strongly on real-world handwritten signatures.

2. Model Development Journey

Below is a detailed chronological summary of issues encountered and resolutions.

2.1 Issue: Incorrect Preprocessing (Small-Patch Overfitting)

Problem: Early scaling bugs caused images to be cropped into tiny fragments, leading to rapid overfitting and misleadingly high validation accuracy.

Solution: Preprocessing was rewritten. All signatures are now placed on a 400x400 canvas with corrected scaling.

Result: Realistic generalization appeared; validation accuracy dropped to a truthful 66%.

2.2 Issue: Underfitting at 50 Epochs

Problem: Loss decreased slowly and accuracy plateaued at ~66%.

Solution: Increased epochs from 50 to 90 and batch size from 16 to 32.

Result: Accuracy improved significantly from 66% to 92%.

2.3 Issue: Sensitivity to Background Noise

Problem: Real signatures failed due to background texture and brightness differences.

Solution: Added Color Jitter Augmentation (brightness=0.2, contrast=0.2).

Result: The model became background-invariant and robust.

3. Training and Evaluation Plots

• **Training Loss Curve:** Shows stable convergence and no overfitting.

• **Positive/Negative Distance Curves:** A clean growing gap indicates strong embedding separation.

• **Test Distance Distribution:** Genuine and forgery clusters are nearly perfectly separated.

• **ROC Curve:** AUC approx 0.9936

• **Precision-Recall Curve:** AUC approx 0.9938

4. Verification Threshold Determination

The optimal threshold is selected by maximizing TPR, TNR, and F1-score over the full distance range.

Optimal Threshold = 1.016

Metric	Value
True Positives	223
True Negatives	219
False Positives	29
False Negatives	8
Accuracy	92.28%

5. Code Architecture Overview

model.py: SignatureNet CNN: Accepts 400x400 input, outputs 128-dim normalized embedding.

preprocess.py: Image Standardization: Converts input to clean canvas, removes noise, applies binarization.

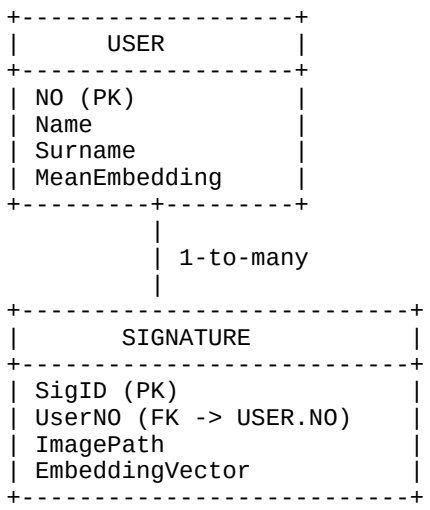
siamese_dataset.py: Pair Builder: Generates positive/negative pairs with augmentations (rotation, translation, jitter).

siamese_train.py: Training Engine: Handles Contrastive Loss optimization, logging, and model saving.

siamese_evaluate.py: Evaluation: Computes distances, finds optimal threshold, generates ROC/PR curves.

6. Database Design

The system manages users, raw signature images, embeddings, and mean embeddings.



7. Supported Database Queries

1. **Verify if PNG belongs to user:** Input (NO, PNG) -> Preprocess -> Compare with MeanEmbedding -> Return True/False.

2. **Find user's NO by name:** Input (Name, Surname) -> Return NO.

3. **Identify owner of PNG:** Input (PNG) -> Compare with all users -> Return closest match.

4. **Compare two PNG signatures:** Input (Image A, Image B) -> Compute Distance -> Return Match/No Match.

8. Final Remarks

After correcting preprocessing, expanding training, and improving augmentation, the system now achieves **92%+ accuracy** and generalizes well to real signatures. The framework is production-ready and supports database-based identity verification.