SIL775: Biometric Security
Assignment - 3
Online Signature Verification

Due by: 11th April 2025

1 Introduction

Online signature verification is a biometric authentication method that utilizes the dynamic characteristics of a handwritten signature to verify an individual's identity. Unlike offline verification, which relies solely on static images, online verification captures dynamic features such as pen velocity, pressure, and stroke order. This assignment explores a system for online signature verification using the SVC2004 dataset, which contains genuine and forged signatures collected under controlled conditions.

Refer to the paper *"Online Signature Verification on Mobile Devices"* for the implementation details.

2 System Overview

The proposed system consists of three primary components:

- **Feature Extractor**: Processes an input signature and extracts meaningful features.
- **Template Generator**: Constructs a user-specific signature template using multiple enrolled samples.
- **Matcher**: Compares a test signature against the stored template to determine authenticity.

3 Feature Extraction

The feature extraction process transforms time-series data of a signature into a sequence of Cartesian vectors, attributes, and their derivatives. Each Cartesian vector is also converted into a vector in the polar coordinate system. Histograms from these vector sequences serve as signature features.

Let $X = \{x_1, x_2, \dots, x_n\}, \quad Y = \{y_1, y_2, \dots, y_n\}, \quad P = \{p_1, p_2, \dots, p_n\}$ represent the x-coordinates, y-coordinates, and pressure attributes, respectively, of a signature with length n.

3.1 First-Order Derivatives

The first-order derivatives of the positional attributes are computed as:

$$X_1 = \{x_1^i \mid x_1^i = x_{i+1} - x_i\}, \quad Y_1 = \{y_1^i \mid y_1^i = y_{i+1} - y_i\}$$

3.2 Higher-Order Derivatives

For $k > 1$, higher-order derivatives are computed iteratively:

$$\begin{aligned} X_k &= \{x_k^i \mid x_k^i = x_{k-1}^{i+1} - x_{k-1}^i\}, \\ Y_k &= \{y_k^i \mid y_k^i = y_{k-1}^{i+1} - y_{k-1}^i\}, \\ P_k &= \{p_k^i \mid p_k^i = p_{k-1}^{i+1} - p_{k-1}^i\} \end{aligned}$$

3.3 Vector Sequence Construction

Then, a sequence of vectors is constructed:

$$V = \{v_i^* \mid i = 1, 2, \ldots, n\}$$

where each vector element

$$v_i^* = \{v_i^1 \parallel \cdots \parallel v_i^j\}$$

is the concatenation of v_i^k, which is a five-tuple consisting of the k-th order derivative of the Cartesian and polar coordinates along with pressure attributes:

$$v_i^k = \langle x_i^k, y_i^k, r_i^k, \theta_i^k, p_i^k \rangle$$

where:

$$\theta_i^k = \tan^{-1}\left(\frac{y_i^k}{x_i^k}\right), \quad r_i^k = \sqrt{(x_i^k)^2 + (y_i^k)^2}$$

for $i = 1, 2, \ldots, n - k$.

4 Histograms

Below are some examples of histograms that are derived from the distributions of attributes in V.

4.1 One-Dimensional Histograms

1. **Angle Distribution (Φ_1):** Measures the direction of movement at each signature point.

$$\Phi^1(i) = \tan^{-1}\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right)$$

   Binned over $[-\pi, \pi]$.

2. **First Derivative of Angles (ϕ_2):** Captures changes in signature curvature.

$$\Phi^2(i) = \Phi^1(i+1) - \Phi^1(i)$$

3. **Speed Distribution (R^1):** Represents the magnitude of movement between consecutive points.

$$R^1(i) = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

4.2 Two-Dimensional Histograms

1. **Speed-Angle Dependency** ($\langle \Phi^1, R^1 \rangle$): Captures how speed relates to direction.
   - The x-axis corresponds to angle bins (Φ^1).
   - The y-axis corresponds to speed bins (R^1).
   - The count of points in each bin forms the histogram.
2. **Angular Relationship Over Time** ($\langle \Phi^1, \Phi^1_{d(1,2)} \rangle$): Captures how three consecutive angles evolve over time.
   - Compute the angular coordinate at each point:
   $$\Phi^1(i) = \tan^{-1}\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right)$$
   - Compute the first-order angular difference to track directional change:
   $$\Phi^1_d(i) = \Phi^1(i+1) - \Phi^1(i)$$
   - Compute the second-order angular difference to capture variations in curvature:
   $$\Phi^1_{d(1,2)}(i) = \Phi^1_d(i+1) - \Phi^1_d(i)$$
   - Construct the histogram by binning pairs $(\Phi^1(i), \Phi^1_d(i))$ into a 2D grid:
     - The x-axis corresponds to bins of $\Phi^1(i)$, ranging from $-\pi$ to π.
     - The y-axis corresponds to bins of $\Phi^1_d(i)$.
     - Each bin counts the occurrences of values falling into the corresponding range.

5 Feature Vector Construction

Each histogram must be computed by splitting the attribute value range into equal-width bins and counting the number of elements in each bin. The histograms that need to be computed are summarized below. Refer to the paper for more details on it.

Before constructing the feature vector, we compute histograms using two types of frequency representations:

- **Absolute Frequency**: The raw count of elements in each bin, which implicitly depends on the signature length.
- **Relative Frequency**: The normalized count (divided by the total number of elements), making it independent of signature length.

Table 1: Descriptions of Histograms Used in the Proposed Technique

No.	Histogram	Input Attributes	Min	Max	Output Attr.
1	Φ^1	$\{\theta^1_1, \ldots, \theta^1_n\}$	$-\pi$	π	Rel. freq.
2	Φ^2	$\{\theta^2_1, \ldots, \theta^2_n\}$	$-\pi$	π	Rel. freq.
3	$\langle \Phi^1, \Phi_d^{1(1,2)} \rangle$	$\{\theta^1_1, \ldots, \theta^1_{n-1}, \theta^1_{n-2}\}, \{\theta^1_2, \ldots, \theta^1_n\}$	$-\pi$	π	Abs. freq.
4	R^1	$\{r^1_1, \ldots, r^1_n\}$	0	$\mu + 3\sigma$	Abs. freq.
5	R^2	$\{r^2_1, \ldots, r^2_n\}$	0	$\mu + 3\sigma$	Abs. freq.
6	X^1	$\{x^1_1, \ldots, x^1_n\}$	$\mu - 3\sigma$	$\mu + 3\sigma$	Rel. freq.
7	Y^1	$\{y^1_1, \ldots, y^1_n\}$	$\mu - 3\sigma$	$\mu + 3\sigma$	Rel. freq.
8	X^2	$\{x^2_1, \ldots, x^2_n\}$	$\mu - 3\sigma$	$\mu + 3\sigma$	Rel. freq.
9	Y^2	$\{y^2_1, \ldots, y^2_n\}$	$\mu - 3\sigma$	$\mu + 3\sigma$	Rel. freq.
10	$\langle X^1, X^2 \rangle$	$\{x^1_1, \ldots, x^1_n\}, \{x^2_1, \ldots, x^2_n\}$	$\mu - 3\sigma$	$\mu + 3\sigma$	Rel. freq.
11	$\langle Y^1, Y^2 \rangle$	$\{y^1_1, \ldots, y^1_n\}, \{y^2_1, \ldots, y^2_n\}$	$\mu - 3\sigma$	$\mu + 3\sigma$	Rel. freq.
12	$\langle \Phi^1, R^1 \rangle_{(1)}$	$\{\theta^1_1, \ldots, \theta^1_{n/2}\}, \{r^1_1, \ldots, r^1_{n/2}\}$	$-\pi$	π	Rel. freq.
	$\langle \Phi^1, R^1 \rangle_{(2)}$	$\{\theta^1_{n/2+1}, \ldots, \theta^1_n\}, \{r^1_{n/2+1}, \ldots, r^1_n\}$	0	$\mu + 3\sigma$	Rel. freq.
13	$\langle \Phi^2, R^2 \rangle_{(1)}$	$\{\theta^2_1, \ldots, \theta^2_{n/2}\}, \{r^2_1, \ldots, r^2_{n/2}\}$	$-\pi$	π	Rel. freq.
	$\langle \Phi^2, R^2 \rangle_{(2)}$	$\{\theta^2_{n/2+1}, \ldots, \theta^2_n\}, \{r^2_{n/2+1}, \ldots, r^2_n\}$	0	$\mu + 3\sigma$	Rel. freq.
14	$\langle \Phi^1, R^2 \rangle_{(1)}$	$\{\theta^1_1, \ldots, \theta^1_{n/2}\}, \{r^2_1, \ldots, r^2_{n/2}\}$	$-\pi$	π	Rel. freq.
	$\langle \Phi^1, R^2 \rangle_{(2)}$	$\{\theta^1_{n/2+1}, \ldots, \theta^1_n\}, \{r^2_{n/2+1}, \ldots, r^2_n\}$	0	$\mu + 3\sigma$	Rel. freq.
15	$P_1^{(1)}$	$\{p_1, \ldots, p_{n/2}\}$	0	$\mu + 3\sigma$	Abs. freq.
	$P_1^{(2)}$	$\{p_{n/2+1}, \ldots, p_n\}$	0	$\mu + 3\sigma$	Abs. freq.
16	$P_2^{(1)}$	$\{p_1, \ldots, p_{n/2}\}$	$\mu - 3\sigma$	$\mu + 3\sigma$	Rel. freq.
	$P_2^{(2)}$	$\{p_{n/2+1}, \ldots, p_n\}$	$\mu - 3\sigma$	$\mu + 3\sigma$	Rel. freq.

Finally, all histograms are concatenated to form a feature vector:

$$F = \{B_1 \parallel B_2 \parallel \cdots \parallel B_j\}$$

where j is the total number of histograms, and $\parallel$ is the concatenation operator.

6 Template Generation

During the enrollment phase, multiple signature samples from a user are collected, and a feature set is extracted for each sample. A user-specific uniform quantizer is then constructed based on the variance of each feature component, defining a quantization step size vector Q^u. This vector is used to quantize the feature vectors of the enrolled samples, and their average forms the user-specific template $\ddot{F}^u$. This template, along with Q^u, serves as a stable reference for verification.

6.1 Step-by-Step Process

6.1.1 Enrollment Data Collection

- Collect S signature samples from the user.
- Each sample contains M features, represented as:

$$F^{s_j} = \{f_i^{s_j} \mid i = 1, \dots, M\}$$

where F^{s_j} is the feature vector of the enrolled sample s_j for user u, and $j = 1, \dots, S$.

6.1.2 Computation of the Quantization Step Size Vector Q_u

- Compute the mean of each feature across all enrolled samples:

$$\mu_{f_i^{(u)}} = \frac{1}{S} \sum_{j=1}^{S} f_i^{s_j}$$

- Compute the standard deviation of each feature:

$$\sigma_i = \sqrt{\frac{1}{S} \sum_{j=1}^{S} (f_i^{s_j} - \mu_{f_i^{(u)}})^2}$$

- Define the quantization step size for each feature using a scaling factor β (empirically set to 1.5):

$$q_i^u = \beta \cdot \sigma_i$$

The quantization step size vector is then:

$$Q^u = \{q_i^u \mid i = 1, \dots, M\}$$

6.1.3 Quantization of Enrolled Feature Vectors

Each enrolled sample's feature vector is quantized using the step size vector:

$$\hat{f}_i^{(s_j|u)} = \frac{f_i^{s_j}}{q_i^u + \epsilon}$$

where ϵ is a small constant (e.g., 0.002 for absolute frequency histograms and 0.8 for relative frequency histograms) to prevent division by zero.

6.1.4 User-Specific Template Generation

The final template $\ddot{F}^u$ is computed as the average of all quantized feature vectors:

$$\ddot{F}^u = \{\ddot{f}_i^u = \frac{1}{S} \sum_{j=1}^{S} \hat{f}_i^{s_j} \mid i = 1, \dots, M\}$$

This template provides a robust reference for comparison during the verification process, ensuring stability despite intra-user variations.

7 Matching and Verification

During verification, given a test signature t claimed to be from user u, its feature vector $\hat{F}^{(t|u)}$ is computed using Q^u. The dissimilarity score is computed using the Manhattan distance:

$$Score = \sum_{i=1}^{M} |\hat{f}^{(t|u)} - \ddot{f}_i^u|$$

If the dissimilarity score is below a predefined threshold, the signature is accepted; otherwise, it is rejected.

8 Dataset: SVC2004

The SVC2004 dataset is a widely used benchmark for online signature verification research. It contains:

- **Genuine signatures** collected from real users.
- **Skilled forgeries** created by individuals attempting to mimic genuine signatures.
- **Time-sequenced signature data**, including x-y coordinates, pressure, and timestamps.

The dataset can be accessed at: `SVC2004 Dataset Link`.

9 Submission Instructions

1. **Code Modularity:**
   - Implement each stage of the pipeline (feature extraction, template generation, matching) in separate, well-commented files.
   - Use Python, C++, or MATLAB for coding.
2. **Submission Package:**
   - Upload your code, sample input/output files, and a PDF with clear instructions on how to run the code.
   - Ensure comments explain each function's purpose, inputs, and outputs.
   - Zip all files into a single archive for submission.
3. **Deadline and Late Submission Policy:**
   - Submit by **11th April 2025 (midnight)**.
   - Late submissions will be penalized as follows:
     - **12th-13th April 2025:** 5% deduction per day.
     - **14th-16th April 2025:** 10 marks deduction per day.
     - **After 16th April 2025:** Submissions not accepted.

4. **Plagiarism Policy:**

   - Similarity exceeding 15% in your code (compared to other students or online sources) results in severe penalties, including a potential zero grade.
   - Maintain originality; properly cite all sources.