ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KHOA HỌC MÁY TÍNH

BÁO CÁO CUỐI KÌ
CÁC VẤN ĐỀ CHỌN LỌC
TRONG THỊ GIÁC MÁY TÍNH - CS420

SIGNATURE VERIFICATION
XÁC MINH CHỮ KÝ

Giảng viên: TS. Mai Tiến Dũng
TS. Đỗ Văn Tiến

Nhóm sinh viên:

Tên	MSSV
Hồ Yến Nhi	21520380
Huỳnh Phạm Đức Lâm	21521050

12/2024, TP Hồ Chí Minh

Mục lục

CHƯƠNG 1. BÀI TOÁN

1.1 Giới thiệu bài toán

Chữ ký viết tay từ lâu đã được công nhận là một trong những phương thức xác thực sinh trắc học phổ biến, được sử dụng rộng rãi trong việc xác minh danh tính, tài liệu và nhiều mục đích khác. Xác minh chữ ký là một nhiệm vụ quan trọng nhằm đảm bảo tính hợp pháp của các giao dịch và ngăn chặn các giao dịch bất hợp pháp.

1.2 Mục tiêu của bài toán

Mục tiêu của bài toán là phát triển một giải pháp tự động để phân biệt chữ ký thật và giả, giúp giảm tải công việc cho các chuyên gia và tăng tốc độ xử lý. Giải pháp này sẽ thay thế quá trình kiểm tra thủ công, vốn đòi hỏi sự tỉ mỉ, kỹ năng và kinh nghiệm từ các chuyên gia, đồng thời vẫn đảm bảo độ chính xác cao trong việc xác minh chữ ký.

1.3 Mô tả bài toán

1.3.1 Dữ liệu đầu vào (Input)

Đầu vào của hệ thống bao gồm hai thành phần chính:

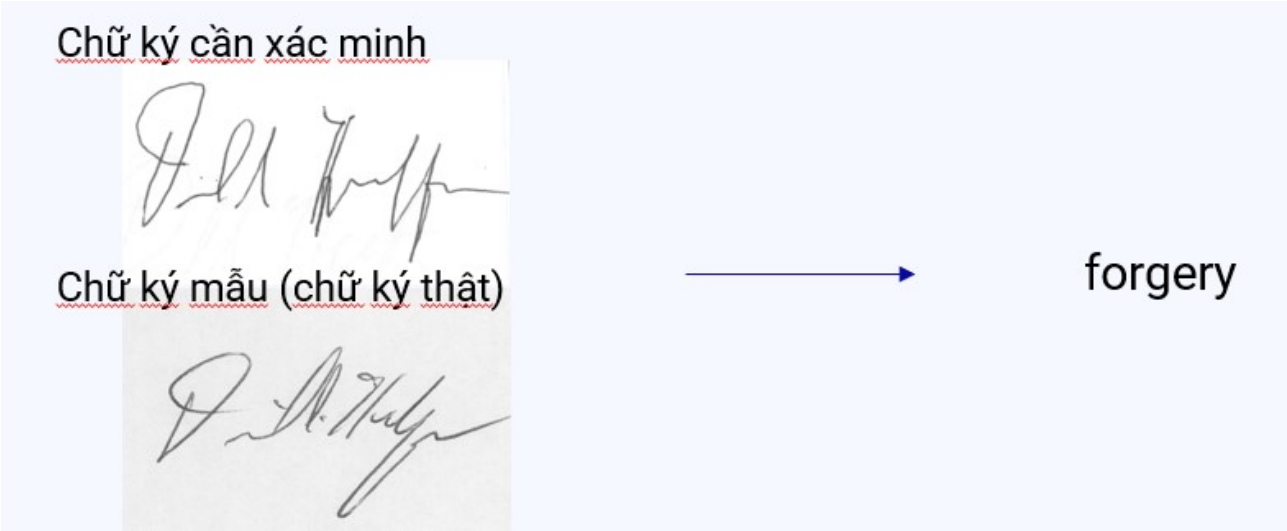
- Thứ nhất, là một ảnh chữ ký cần xác minh, đây là chữ ký chưa được kiểm tra và sẽ được so sánh với các mẫu chữ ký thật để xác định tính xác thực.
- Thứ hai, là một hoặc nhiều ảnh mẫu chữ ký thật, được sử dụng để làm cơ sở so sánh với chữ ký cần xác minh. Những mẫu chữ ký thật này giúp hệ thống học và nhận diện các đặc điểm chung của chữ ký hợp lệ.

1.3.2 Dữ liệu đầu ra (Output)

Kết quả của hệ thống xác minh chữ ký sẽ trả về một trong hai nhãn: genuine (chữ ký thật) hoặc forgery (chữ ký giả). Khi hệ thống nhận được ảnh chữ ký cần xác minh và so sánh với các mẫu chữ ký thật, nó sẽ phân tích các đặc điểm của chữ ký và đưa ra kết luận.

- Nếu chữ ký cần xác minh phù hợp với các mẫu chữ ký thật, kết quả trả về sẽ là genuine.
- Ngược lại, nếu chữ ký cần xác minh có sự khác biệt đáng kể so với mẫu chữ ký thật, hệ thống sẽ phân loại nó là forgery (chữ ký giả).

Kết quả này sẽ giúp xác minh tính hợp lệ của chữ ký trong các giao dịch hoặc tài liệu cần kiểm tra.



Hình 1: Ảnh minh họa Input Output của bài toán Xác minh chữ ký.

1.4 Các phương pháp tiếp cận trong xác minh chữ ký

Có hai phương pháp tiếp cận chính trong xác minh chữ ký:

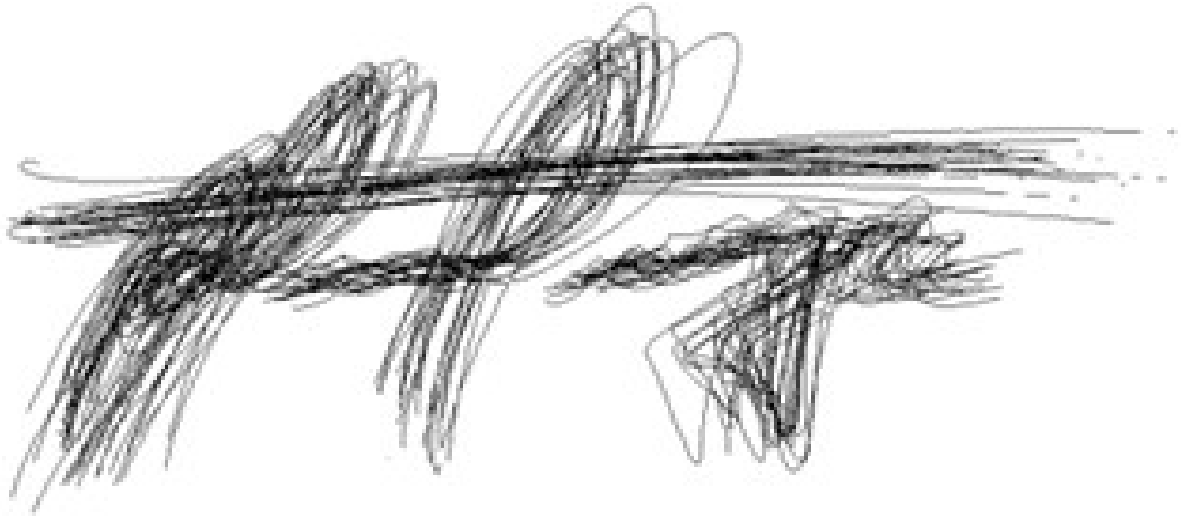
- Writer-independent (WI): Sử dụng một model chung để xác minh chữ ký cho bất kỳ người nào dựa trên các mẫu chữ ký của họ. Phương pháp này không yêu cầu việc huấn luyện lại mô hình cho mỗi người dùng, mà sử dụng một mô hình tổng quát để phân biệt chữ ký thật và giả cho nhiều người khác nhau.
- Writer-dependent (WD): Ứng với từng người sẽ tạo ra một model riêng để thực hiện việc xác minh chữ ký dựa trên mẫu chữ ký của họ. Phương pháp này yêu cầu việc huấn luyện mô hình riêng cho mỗi cá nhân dựa trên các mẫu chữ ký của họ, giúp tăng độ chính xác trong việc xác minh chữ ký của từng người.

Trong bài báo cáo này thì nhóm em sử dụng các phương pháp Writer-independent để giải quyết bài toán này.

1.5 Thách thức của bài toán

Bài toán xác minh chữ ký gặp phải không ít khó khăn, với các thách thức chủ yếu như:

- **Biến động cao trong cùng một lớp chữ ký (high intra-class variance)**: Chữ ký của một người có thể thay đổi theo thời gian hoặc tình huống, điều này gây khó khăn trong việc phân biệt chữ ký thật và chữ ký giả.



Hình 2: Ảnh hợp các nét chữ ký của cùng một người.

- **Chữ ký giả ngày càng tinh vi**: Chữ ký giả trở nên phức tạp hơn, khiến việc phân biệt chúng với chữ ký thật trở nên khó khăn hơn.
- **Thiếu dữ liệu chữ ký giả**: Việc thiếu dữ liệu chữ ký giả để huấn luyện mô hình làm giảm khả năng học các đặc điểm của chữ ký giả.
- **Số lượng chữ ký mẫu hạn chế**: Mỗi người dùng thường chỉ cung cấp một số lượng chữ ký mẫu rất ít, điều này hạn chế khả năng huấn luyện mô hình và ảnh hưởng đến độ chính xác trong quá trình xác minh.

CHƯƠNG 2. CÁC NGHIÊN CỨU - HỆ THỐNG LIÊN QUAN

Nghiên cứu về xác minh chữ ký ngoại tuyến đã bắt đầu từ những năm 1970. Qua nhiều thập kỷ, các phương pháp tiếp cận khác nhau đã được đề xuất và tóm lược trong các nghiên cứu [1], [2], và [3].

Vấn đề trong xác minh chữ ký ngoại tuyến là làm thế nào để tạo ra một mô hình hoặc hệ thống có khả năng phân biệt giữa chữ ký thật và chữ ký giả. Để làm điều này, hệ thống cần được "học" từ một tập dữ liệu bao gồm các mẫu chữ ký thật của một cá nhân. Những mẫu chữ ký này đóng vai trò làm cơ sở để mô hình nhận dạng các đặc điểm của chữ ký thật, từ đó phát hiện ra những chữ ký không hợp lệ (chữ ký giả). Chữ ký giả được chia thành hai loại chính: giả ngẫu nhiên, khi kẻ mạo danh sử dụng chữ ký của mình để giả làm người khác, và giả chuyên nghiệp, khi chữ ký được sao chép từ các mẫu chữ ký thật [4].

Giống như các lĩnh vực nhận dạng mẫu khác, nghiên cứu về xác minh chữ ký ngoại tuyến tập trung vào việc phát triển các phương pháp mô tả đặc trưng hiệu quả để trích xuất thông tin từ chữ ký.

Trong phần này, nhóm em sẽ tóm tắt các nghiên cứu trước đây theo hai khía cạnh chính: trích xuất đặc trưng thủ công (Handcrafted Feature Extractor) và các phương pháp học sâu (Deep Learning).

2.1 Bộ trích xuất đặc trưng thủ công (Handcrafted Feature Extractor)

Trước khi các mô hình học sâu trở nên phổ biến trong nghiên cứu, đã có rất nhiều phương pháp trích xuất đặc trưng được phát triển để giải quyết vấn đề này. Các phương pháp bao gồm:

- Đặc trưng hình học, được xây dựng dựa trên các yếu tố hình dạng [5], [6] và [7].
- Đặc trưng định hướng, chẳng hạn như Directional-PDF (hàm mật độ xác suất theo hướng) [8] và PHOG (biểu đồ kim tự tháp của các gradient định hướng) [9].
- Đặc trưng kết cấu, ví dụ như Local Binary Patterns (LBP - mẫu nhị phân cục bộ) [10], [11] và GLCM (ma trận đồng xuất hiện mức xám) [12], [13].

Ngoài ra, còn có các thuật toán trích xuất đặc trưng "giả động" dựa trên nghiên cứu đồ thị học (graphometry). Những thuật toán này cố gắng khôi phục thông tin động, như tốc độ viết, tính liên tục và sự đồng đều, từ hình ảnh chữ ký tĩnh [14].

2.2 Các phương pháp học sâu (Deep learning)

Sau khi học sâu được áp dụng vào lĩnh vực xác minh chữ ký, các nghiên cứu ban đầu chủ yếu sử dụng bộ dữ liệu riêng và không đạt được nhiều thành công rõ rệt. Ribeiro và cộng sự [15] đã sử dụng các Mô hình Boltzmann Hạn chế (RBMs) để học đặc trưng từ hình ảnh chữ ký,

nhưng chỉ trình bày kết quả dưới dạng hình ảnh các trọng số mà không kiểm tra chúng trong việc xác minh chữ ký. Khalajzadeh [16] sử dụng Mạng nơ-ron Tích chập (CNN) để xác minh chữ ký tiếng Ba Tư, nhưng chỉ xét các chữ ký giả ngẫu nhiên trong nghiên cứu của mình.

Hafemann và cộng sự [17] đề xuất phương pháp học các đặc trưng độc lập với người viết bằng cách sử dụng CNN trên một tập dữ liệu phát triển D, sau đó sử dụng các đặc trưng này để huấn luyện các bộ phân loại phụ thuộc vào người viết trên tập khai thác ϵ. Trong nghiên cứu tiếp theo [4], các tác giả cũng đề xuất một phương pháp đa nhiệm, trong đó CNN được huấn luyện cả trên chữ ký thật và chữ ký giả chuyên nghiệp.

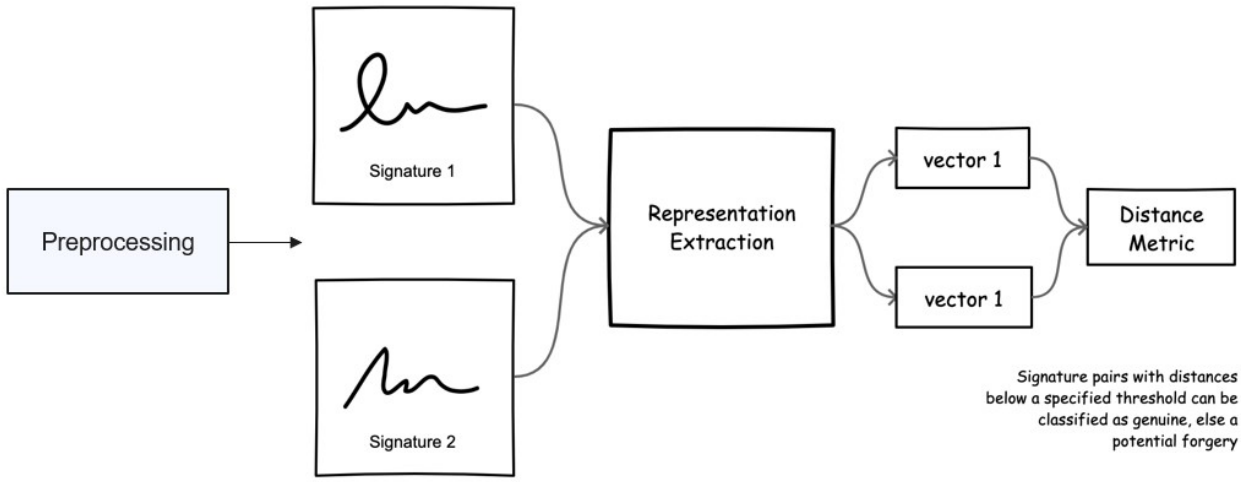
Zhang và cộng sự [18] sử dụng Mạng đối kháng tạo sinh (GAN) [19] để học đặc trưng, trong đó một mô hình tạo sinh học cách tạo ra hình ảnh chữ ký và một mô hình phân biệt học cách phân biệt chữ ký thật và chữ ký giả. Sau khi huấn luyện, các lớp tích chập của mô hình phân biệt được sử dụng làm bộ trích xuất đặc trưng.

Rantzsch và cộng sự [20] đã áp dụng học theo số liệu để xây dựng một phương pháp độc lập với người viết, trong đó khoảng cách giữa các chữ ký được học để phục vụ xác minh. Trong quá trình huấn luyện, các bộ ba chữ ký (X_r, X^+, X^-) được đưa vào mạng nơ-ron: X_r là mẫu tham chiếu, X^+ là chữ ký thật và X^- là chữ ký giả. Mạng nơ-ron học cách giảm khoảng cách giữa X_r và X^+, đồng thời tăng khoảng cách giữa X_r và X^-. Sau khi huấn luyện, chữ ký có khoảng cách dưới một ngưỡng nhất định với mẫu tham chiếu được xem là chữ ký thật, còn chữ ký có khoảng cách vượt quá ngưỡng sẽ bị coi là chữ ký giả.

CHƯƠNG 3. PHƯƠNG PHÁP

3.1 Kiến trúc chung

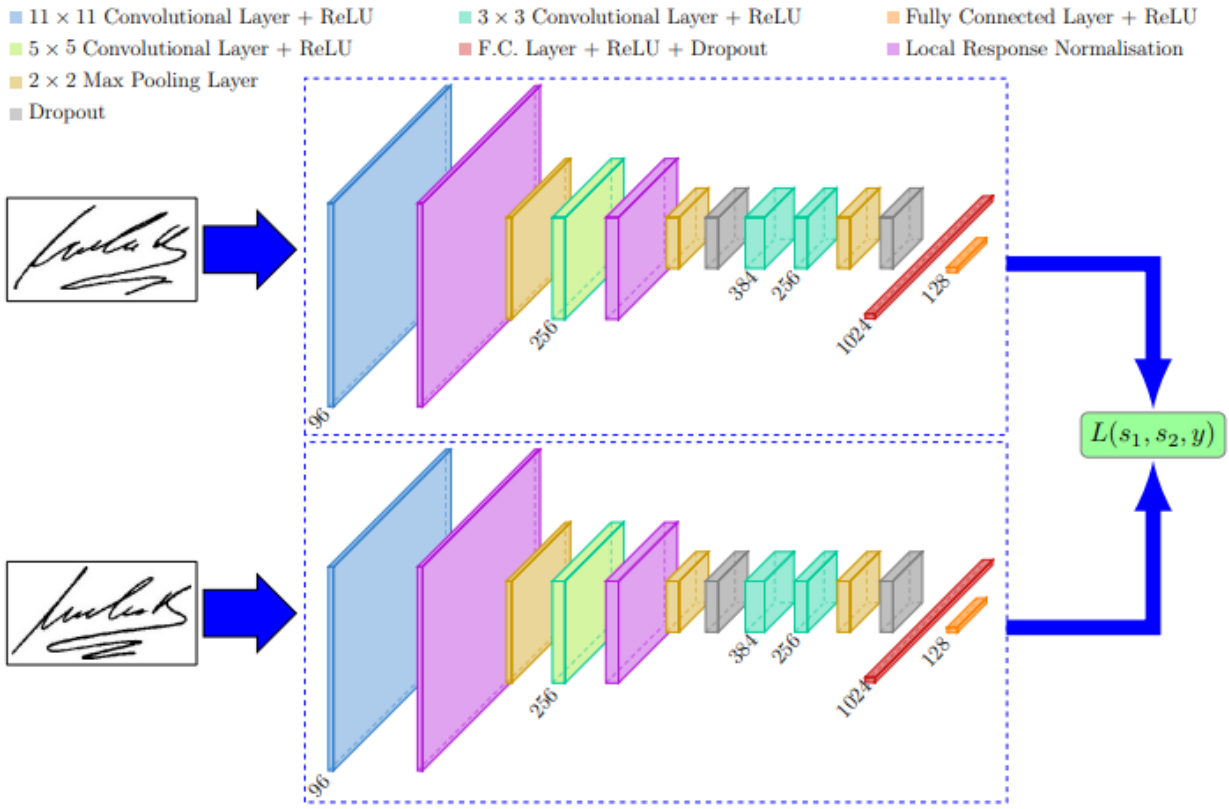
Quy trình xác minh chữ ký bắt đầu với bước tiền xử lý (preprocessing), sau đó tiến hành trích xuất đặc trưng (extract features) từ hai loại chữ ký: chữ ký mẫu và chữ ký cần xác minh. Mô hình sẽ phân tích và rút ra các đặc trưng quan trọng từ ảnh chữ ký, chuyển đổi chúng thành các embedding vectors. Tiếp theo, sẽ tính khoảng cách giữa hai vector này, đại diện cho chữ ký cần xác minh và mẫu chữ ký thật. Cuối cùng, dựa trên khoảng cách đã tính được, hệ thống sẽ phân loại chữ ký là genuine (thật) hoặc forgery (giả).



Hình 3: Pipeline của bài toán Xác minh chữ ký.

3.2 SigNet

SigNet là một mô hình mạng nơ-ron tích chập (CNN) được thiết kế chuyên biệt để xác thực chữ ký tay, cung cấp giải pháp hiệu quả cho bài toán nhận dạng chữ ký thật và giả. Mô hình này được xây dựng dựa trên kiến trúc mạng Siamese, với hai nhánh CNN giống hệt nhau và sử dụng chung trọng số. SigNet được huấn luyện bằng hàm contrastive loss, đảm bảo rằng chữ ký thuộc cùng một người (cùng lớp) sẽ có khoảng cách nhỏ trong không gian đặc trưng, trong khi chữ ký của các người khác nhau (khác lớp) sẽ có khoảng cách lớn. Với khả năng trích xuất đặc trưng mạnh mẽ từ chữ ký, SigNet đã chứng minh hiệu quả cao trong việc nhận dạng chữ ký trên các bộ dữ liệu tiêu chuẩn, trở thành một công cụ hữu ích trong lĩnh vực xác thực danh tính và bảo mật.



Hình 4: Mô hình SigNet

Kiến trúc mạng Siamese gồm 2 mạng CNN giống hệt nhau và các kernel giảm dần từ 11x11 đến 3x3. Các mạng CNN này chia sẻ trọng số với nhau và việc huấn luyện các mạng này được cập nhật cho giống nhau trong quá trình huấn luyện. Sau khi trích xuất được vector cuối cùng từ các thông tin đầu vào, ta có thể tính hàm contrastive loss thường thấy trong kiến trúc này từ độ đo khoảng cách Euclid giữa 2 vector đặc trưng cuối cùng.

$$L(s_1, s_2, y) = \alpha(1-y)D_w^2 + \beta y \max(0, m - D_w)^2 \tag{1}$$

với s_1, s_2 là 2 ảnh đầu vào, y là hàm nhị phân cho biết 2 ảnh có cùng class hay không, D_w là khoảng cách Euclid của 2 vector đặc trưng ứng với mỗi ảnh đầu vào.

Kiến trúc CNN dựa trên nghiên cứu của Krizhevsky [21]để giải quyết bài toán nhận dạng hình ảnh. Mạng nhận đầu vào là ảnh kích thước 155 × 220, áp dụng các lớp tích chập và pooling với thông số chi tiết trong Bảng 1. Các lớp tích chập gồm: lớp đầu sử dụng 96 bộ lọc kích thước 11 × 11, lớp thứ hai với 256 bộ lọc 5 × 5, và hai lớp tiếp theo lần lượt có 384 và 256 bộ lọc 3 × 3. ReLU được dùng làm hàm kích hoạt, và Local Response Normalization được áp dụng. Dropout được sử dụng ở các lớp pooling cuối và fully connected với tỷ lệ 0.3 và 0.5. Mạng có hai lớp fully connected, lần lượt chứa 1024 và 128 neuron, cho ra vector đặc trưng cuối cùng kích thước 128.

Lớp	Kích thước	Tham số
Conv2D	96x11x11	stride = 4
Batch Normalization	-	$\epsilon = 10^{-6}$, momentum = 0.9
MaxPooling2D	96x3x3	stride = 2
Conv2D	256x5x5	stride = 1, pad = 2
Batch Normalization	-	$\epsilon = 10^{-6}$, momentum = 0.9
MaxPooling2D + Dropout	256x3x3	stride = 2, p = 0.3
Conv2D	384x3x3	stride = 1, pad = 1
Conv2D	256x3x3	stride = 1, pad = 1
MaxPooling2D + Dropout	256x3x3	stride = 2, p = 0.3
Fully Connected + Dropout	1024	p = 0.5
Fully Connected	128	

Bảng 1: Các lớp của mô hình CNN

3.3 SigmML

SigmML là một tiếp cận dựa trên đa tạp ma trận xác định đối xứng (SPD manifold) để học một hàm khoảng cách giữa 2 mẫu chữ ký đầu vào.

Đầu tiên, hình ảnh chữ ký ban đầu I_{raw} được xử lý trước theo các bước phổ biến trong tài liệu [22]. Quá trình này bao gồm:

- Nhị phân hóa bằng phương pháp Otsu.
- Làm mảnh (thinning) để giảm độ dày đường nét của chữ ký xuống còn 1 pixel, từ đó trích xuất các thông tin quan trọng I từ hình ảnh gốc I_{raw}.

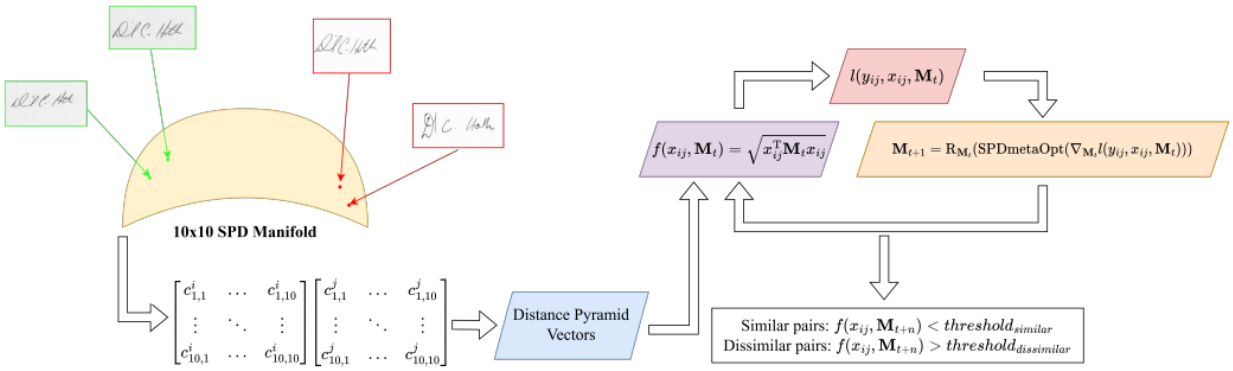
Tiếp theo, 10 bộ lọc ảnh được áp dụng lên I, được định nghĩa như sau:

$$F(I) = \left\{ \begin{matrix} I,\ |I_x|,\ |I_y|,\ \sqrt{I_x^2 + I_y^2},\ \tan^{-1}\left(\frac{I_y}{I_x}\right), \\ |I_{xx}|,\ |I_{yy}|,\ |I_{xy}|,\ x,\ y \end{matrix} \right\}. \tag{2}$$

Mỗi lớp trong stack $F(I)$ biểu diễn:

- Cường độ xám I.
- Đạo hàm bậc 1 theo hướng x và y (I_x, I_y).
- Độ lớn và hướng của gradient.
- Đạo hàm bậc 2 (I_{xx}, I_{yy}, I_{xy}).
- Vị trí hàng x và cột y đã chuẩn hóa.

Sau đó, ta tính ma trận hiệp phương sai trên stack 10 hình đó, chính là ma trận SPD là biểu diễn của 1 mẫu chữ ký.



Hình 5: Mô hình SigmML

Distance Pyramid

$$d(C_i, C_j) = \|\mathrm{logm}\left((C_i)^{-\frac{1}{2}} C_j (C_i)^{-\frac{1}{2}}\right)\|_F \tag{3}$$

Với công thức (3) là độ đo Rao-Fisher để tính khoảng cách giữa 2 điểm trong đa tạp SPD,

Giả sử ta có một cặp ma trận hiệp phương sai (C_i, C_j), ta có thể tạo ra một vector độ tương đồng của chúng, gọi là vector Distance Pyramid, DP. Đây là một tập hợp các số thực, mỗi số là khoảng cách giữa các phần con của hai ma trận hiệp phương sai. Cách tính khoảng cách này sử dụng công thức (3) và được áp dụng cho các phần con của ma trận C_i và C_j.

Cụ thể, mỗi phần tử trong vector DP được tính toán bằng cách lấy khoảng cách giữa các phần con của ma trận C_i và C_j. Các phần con này là các ma trận con của C_i và C_j, được chọn từ các chỉ số từ a đến b. Khi thay đổi các giá trị a và b, ta có thể tạo ra các vector DP với số chiều khác nhau.

Ví dụ:

$$\mathrm{DP}_{\{1,10\}} = \{d(C_i, C_j)_{1:10}\}$$

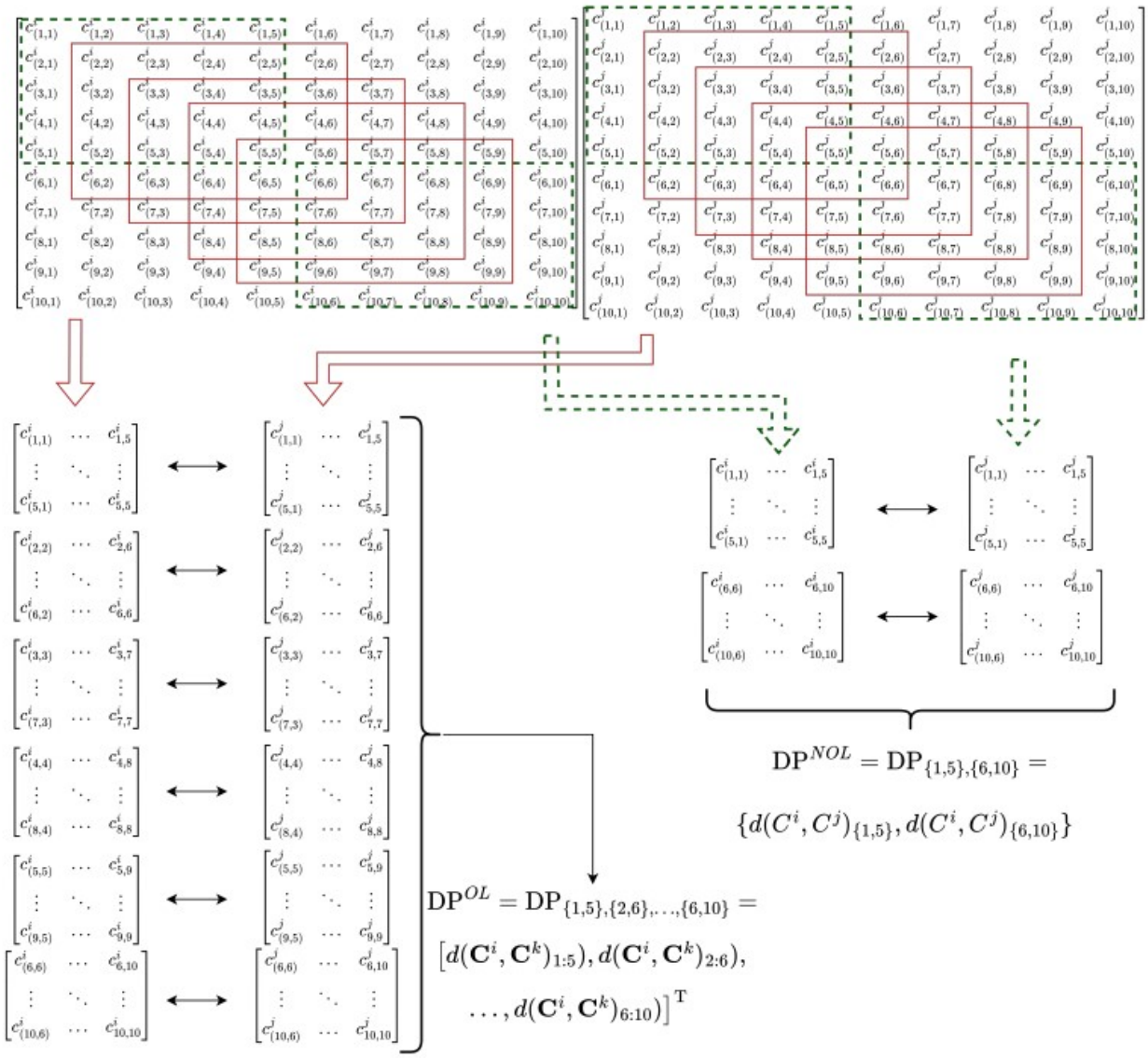
tạo ra một vector chỉ có một số thực.

$$\mathrm{DP}_{\{1,10\},\{1,3\},\{2,7\}} = \{d(C_i, C_j)_{1:10}, d(C_i, C_j)_{1:3}, d(C_i, C_j)_{2:7}\}$$

tạo ra một vector có ba số thực.

Để đơn giản hóa, sử dụng hai cách khác nhau để tạo ra các vector DP này.

- Một cách là không cho các cửa sổ chồng lắp lên nhau, gọi là **DPNOL (non-overlapping)**. Cách này sử dụng các cửa sổ không chồng lấn nhau.
- Cách còn lại là cho các cửa sổ chồng lắp lên nhau, gọi là **DPOL (overlapping)**. Cách này sử dụng các cửa sổ có chồng lấn.



Hình 6: Ví dụ minh họa về cách tính hai vector DP.

Hình 6, minh họa quá trình tạo ra hai vector DP khác nhau với cửa sổ kích thước 5×5, một không chồng lấn nhau và một có chồng lấn. Cả hai cửa sổ đỏ và xanh lục đều được sử dụng để tạo vector DP có chồng lấn DPOL = DP{1,5},{2,6},...,{6,10}. Chỉ có các cửa sổ xanh lục được sử dụng để tạo vector DP không chồng lấn DPNOL = DP{1,5},{6,10}.

Sử dụng hàm *contrastive loss* làm hàm mục tiêu để cập nhật tham số hiệp phương sai M. Khi đó, hàm loss của ta sẽ được định nghĩa:

$$d(x_{ij,kl}, M(t)) = \sqrt{x_{ij,kl}^T M(t) x_{ij,kl}} \tag{4}$$

$$
\begin{aligned}
l(D, S, M(t)) = & \frac{1}{|D|} \sum_{i,j,k,l \in D} \left((1 - y_{ij,kl}) \max(0, \zeta_d - d)^2 \right) \\
& + \frac{1}{|S|} \sum_{i,j,k,l \in S} \left(y_{ij,kl} \max(0, d - \zeta_s)^2 \right)
\end{aligned} \tag{5}
$$

$$
L(D, S, L(t)) = \frac{1}{m} \sum_{n=1}^{m} l(D, S, L(t), n) \tag{6}
$$

với $x_{ij,kl}$ là mẫu chữ kí thứ j và l của người thứ i và k. Khi đó, ma trận M sẽ chính là ma trận tham số của hàm Mahanalobis, chính là ma trận cần tìm để tính khoảng cách giữa 2 mẫu chữ ký của ta.

Bằng cách sử dụng một biến thể của LSTM là mLSTM hoạt động trên ma trận của Zhi Gao [23], ta có thể thực hiện cập nhật ma trận M là không làm mất đi yếu tố đối xứng của nó.

CHƯƠNG 4. KẾT QUẢ THỰC NGHIỆM

4.1 Datasets

Nghiên cứu trong lĩnh vực này, đặc biệt là những nghiên cứu áp dụng học sâu, đòi hỏi các bộ dữ liệu khá lớn. Hầu hết các bộ dữ liệu công khai hiện nay được thu thập theo một quy trình: Người dùng cung cấp các mẫu chữ ký thật của họ trên một tờ giấy chia thành nhiều ô. Sau đó, những người giả mạo sẽ được cung cấp các mẫu chữ ký này và thực hiện bắt chước chữ ký một hoặc nhiều lần. Các mẫu chữ ký thu thập được sẽ được quét và xử lý trước. Trong số các bộ dữ liệu hiện có, CEDAR [24] và BHSig260 [25] phần Bengali là hai bộ được sử dụng phổ biến nhất nên nhóm em quyết định sử dụng hai bộ dữ liệu này cho bài báo cáo của mình.

4.1.1 CEDAR

Bộ dữ liệu CEDAR Signature chứa các chữ ký viết tay được sử dụng cho bài toán xác thực chữ ký. Mỗi người trong số 55 người đóng góp 24 chữ ký thật, tổng cộng tạo ra 1.320 chữ ký thật. Những người này cũng được yêu cầu giả mạo chữ ký của 3 người khác, mỗi người ký giả mạo 8 lần, tạo ra 1.320 chữ ký giả. Các chữ ký được quét ở độ phân giải 300 dpi ảnh xám (gray-scale), sau đó được nhị phân hóa. Trong quá trình tiền xử lý ảnh, áp dụng loại bỏ salt-pepper noise và chuẩn hóa độ nghiêng. Tóm lại, bộ dữ liệu này cung cấp 24 chữ ký thật và 24 chữ ký giả cho mỗi người ký.

4.1.2 BHSig260-Bengali

Bộ dữ liệu BHSig260 gồm các chữ ký của 260 người, trong đó 100 người ký bằng tiếng Bengali và 160 người ký bằng tiếng Hindi. Mặc dù bộ dữ liệu này dưới dạng tổng hợp, nhưng nhóm chỉ tiến hành thực nghiệm các phương pháp trên bộ dữ liệu chữ ký tiếng Bengali. Trong phần Bengali, mỗi người ký 24 chữ ký thật và 30 chữ ký giả, tạo ra tổng cộng $100 \times 24 = 2.400$ chữ ký thật và $100 \times 30 = 3.000$ chữ ký giả bằng tiếng Bengali.

4.2 Độ đo

4.2.1 False Acceptance Rate (FAR)

False Acceptance Rate (FAR) là tỷ lệ sai lệch trong hệ thống xác minh, phản ánh khả năng hệ thống chấp nhận chữ ký giả là chữ ký thật. Tỷ lệ này được tính bằng cách chia số lượng các trường hợp False Positive (chấp nhận chữ ký giả là thật) cho tổng số trường hợp là False Positive và True Negative (chữ ký giả xác nhận là giả). FAR có thể được sử dụng để đánh giá mức độ an toàn và độ chính xác của hệ thống xác minh chữ ký.

Công thức tính FAR như sau:

$$FAR = \frac{\text{False Positive}}{\text{False Positive} + \text{True Negative}} \tag{7}$$

- False Positive (FP): Chữ ký giả bị xác nhận là thật.
- True Negative (TN): Chữ ký giả xác nhận là giả.

FAR càng thấp, hệ thống xác minh chữ ký càng chính xác và đáng tin cậy trong việc phân biệt chữ ký thật và giả.

4.2.2 False Negative Rate (FNR)

False Negative Rate (FNR) là tỷ lệ sai lệch trong hệ thống xác minh, phản ánh khả năng hệ thống từ chối chữ ký thật. Tỷ lệ này được tính bằng cách chia số lượng các trường hợp False Negative (chữ ký thật bị từ chối) cho tổng số trường hợp là False Negative và True Positive (chữ ký thật xác nhận là thật). FNR có thể được sử dụng để đánh giá mức độ chính xác và độ tin cậy của hệ thống trong việc nhận diện chữ ký thật.

Công thức tính FNR như sau:

$$FNR = \frac{\text{False Negative}}{\text{False Negative} + \text{True Positive}} \tag{8}$$

- False Negative (FN): Chữ ký thật bị từ chối, tức là hệ thống không nhận diện là chữ ký hợp lệ dù nó là thật.
- True Positive (TP): Chữ ký thật được xác nhận là thật.

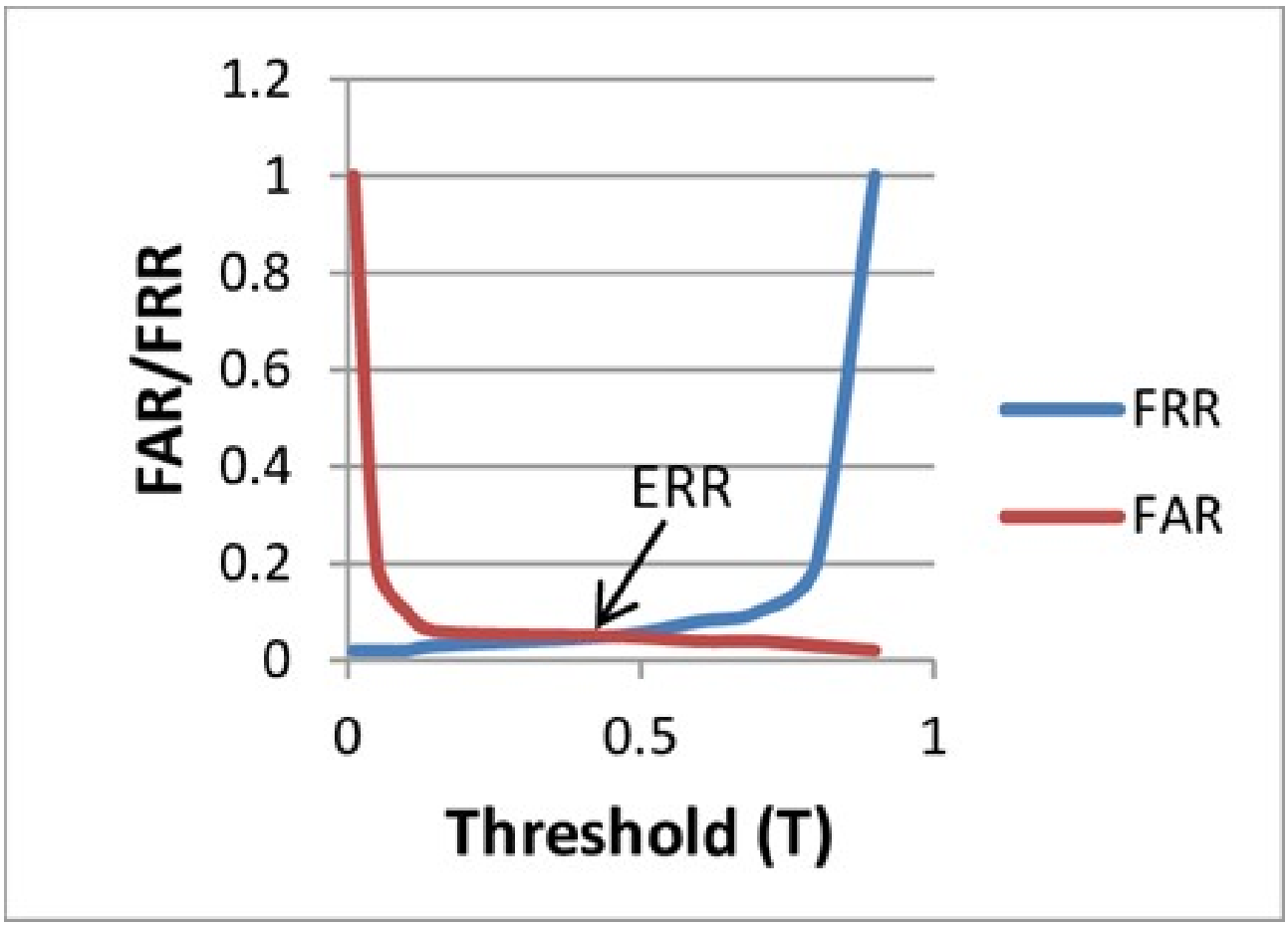
FNR càng thấp, hệ thống xác minh chữ ký càng chính xác và đáng tin cậy trong việc nhận diện chữ ký thật.

4.2.3 Equal Error Rate (EER)

Equal Error Rate (EER) là điểm mà tỷ lệ False Acceptance Rate (FAR) và False Rejection Rate (FRR) gặp nhau, tức là khi giá trị của cả hai tỷ lệ này bằng nhau. Đây là một chỉ số quan trọng trong việc đánh giá hiệu suất của hệ thống xác minh chữ ký. EER giúp xác định điểm cân bằng giữa việc hệ thống từ chối các chữ ký thật và việc hệ thống chấp nhận chữ ký giả là thật.

Trong bài toán xác minh chữ ký, hệ thống sẽ phải đưa ra quyết định về việc xác nhận chữ ký là hợp lệ (chữ ký thật) hay không hợp lệ (chữ ký giả). Tuy nhiên, bất kỳ hệ thống nào cũng có khả năng mắc phải hai loại lỗi cơ bản:

- False Rejection Rate (FRR): Đây là tỷ lệ các chữ ký thật bị hệ thống từ chối, tức là hệ thống không nhận diện chữ ký thật là hợp lệ, dù thực tế nó đúng. Điều này có thể xảy ra khi ngưỡng xác nhận chữ ký quá nghiêm ngặt.
- False Acceptance Rate (FAR): Đây là tỷ lệ các chữ ký giả bị hệ thống nhận diện là thật, tức là hệ thống chấp nhận chữ ký giả như là chữ ký hợp lệ. Lỗi này xảy ra khi hệ thống không đủ chính xác trong việc phân biệt chữ ký giả và chữ ký thật.



Hình 7: Độ đo đánh giá EER.

Mục tiêu của hệ thống xác minh chữ ký là giảm thiểu cả hai loại lỗi này càng nhiều càng tốt, đồng thời duy trì khả năng phân biệt chính xác giữa chữ ký thật và giả. Tuy nhiên, việc giảm FAR quá mức có thể dẫn đến việc gia tăng FRR, và ngược lại, việc giảm FRR quá mức có thể dẫn đến tăng FAR. Do đó, cần phải tìm ra một điểm cân bằng giữa hai tỷ lệ này.

EER có thể được sử dụng như một chỉ số chung để so sánh hiệu suất của các mô hình xác minh chữ ký khác nhau. EER càng thấp, hệ thống xác minh chữ ký càng chính xác, đáng tin cậy và có khả năng phân biệt tốt hơn giữa chữ ký thật và giả, vì nó cho thấy rằng mô hình ít mắc phải các lỗi False Acceptance và False Rejection. Vì vậy, EER là một thước đo hiệu quả trong việc đánh giá khả năng chính xác và độ tin cậy của hệ thống xác minh chữ ký.

4.3 Môi trường thực nghiệm

Bảng trên trình bày chi tiết thông tin về môi trường và tham số huấn luyện cho các phương pháp SigmML và SigNet trong bài toán xác minh chữ ký với hai tập dữ liệu CEDAR và BHSig260-Bengali.

Phương pháp SigmML được huấn luyện trên Kaggle với GPU T4x2 trong 10 epoch, kích thước batch 512, và hoàn thành trong 1 giờ cho mỗi tập dữ liệu. Mô hình tận dụng kích thước batch lớn để tăng hiệu quả học và cần bộ nhớ GPU lớn.

Phương pháp	Dataset	Môi trường Training	Tham số Train	Thời gian
SigmML	CEDAR	Kaggle GPU T4x2	Epoch: 10, Batchsize: 512	1h
SigmML	BHSig260-Bengali	Kaggle GPU T4x2	Epoch: 10, Batchsize: 512	1h
SigNet	CEDAR	GgColab GPU T4x2	Epoch: 50 (Early stopping epoch 29), Batchsize: 128	2h
SigNet	BHSig260-Bengali	GgColab GPU T4x2	Epoch: 50 (Early stopping epoch 22), Batchsize: 128	3h

Bảng 2: Môi trường và thời gian Training của từng Model.

Phương pháp SigNet sử dụng Google Colab với GPU T4x2 và kích thước batch 128. Early Stopping giúp rút ngắn thời gian huấn luyện: dừng tại epoch 29 (2 giờ) với tập CEDAR và epoch 22 (3 giờ) với tập BHSig260-Bengali thay vì huấn luyện toàn bộ 50 epoch, giúp tránh overfitting và tiết kiệm tài nguyên.

4.4 Kết quả thực nghiệm

Phương pháp	Dataset	EER(%)
SigmML	CEDAR	0.036
SigmML	BHSig260-Bengali	0.070
SigNet	CEDAR	0.000
SigNet	BHSig260-Bengali	0.079

Bảng 3: Kết quả thực nghiệm của từng phương pháp đối với từng dataset theo độ đo EER(%).

Bảng trên thể hiện đánh giá các phương pháp SigmML và SigNet trong bài toán xác minh chữ ký với hai tập dữ liệu CEDAR và BHSig260-Bengali bằng độ đo EER (%). Qua đó có thể thấy:

- Đối với bộ dữ liệu CEDAR: Mô hình SigNet đạt EER = 0.000, cho thấy khả năng phân biệt hoàn hảo giữa chữ ký thật và chữ ký giả. Kết quả này thể hiện hiệu suất vượt trội và khả năng tổng quát hóa tốt trên bộ dữ liệu CEDAR. Bên cạnh đó, mô hình SigmML đạt EER = 0.036. Mặc dù kém hơn so với SigNet, đây vẫn là kết quả khá tốt, cho thấy độ chính xác cao trong việc xác minh chữ ký. Tuy nhiên, hiệu suất chưa đạt đến mức tuyệt đối.

- Đối với bộ dữ liệu BHSig260-Bengali: Cả hai mô hình đều có EER cao hơn so với bộ dữ liệu CEDAR, cho thấy BHSig260-Bengali có thể chứa nhiều đặc điểm phức tạp hoặc có độ đa dạng chữ ký lớn hơn. Mô hình SigmML đạt EER = 0.070, thấp hơn một chút so với

SigNet, với EER = 0.079. Điều này cho thấy SigmML có hiệu suất nhỉnh hơn trên bộ dữ liệu này.

Tóm lại, mô hình SigNet thể hiện hiệu quả vượt trội trên tập dữ liệu CEDAR, trong khi SigmML lại hoạt động tốt hơn trên BHSig260-Bengali. Vì vậy, đối với những bộ dữ liệu ít phức tạp như CEDAR, SigNet là lựa chọn ưu tiên vì khả năng phân loại gần như hoàn hảo. Còn đối với dữ liệu có độ phức tạp cao như BHSig260-Bengali, SigmML thể hiện sự ổn định hơn.

CHƯƠNG 5. KẾT LUẬN, HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Bài toán xác minh chữ ký là một bài toán quan trọng trong lĩnh vực bảo mật và xác thực danh tính, với mục tiêu phân biệt chữ ký thật và giả. Trong bài báo cáo này, nhóm em đã tìm hiểu và thực nghiệm 2 phương pháp là SigmML và SigNet để giải quyết bài toán. Các kết quả thực nghiệm từ các mô hình như SigNet và SigmML cho thấy mỗi phương pháp có hiệu suất khác nhau khi thực nghiệm trên các bộ dữ liệu có đặc điểm riêng biệt. Mặc dù SigNet thể hiện ưu thế vượt trội trên bộ dữ liệu CEDAR với khả năng phân biệt tuyệt đối chữ ký thật và giả, mô hình SigmML lại hoạt động hiệu quả hơn trên bộ dữ liệu BHSig260-Bengali là bộ dữ liệu độ phức tạp cao hơn với nhiều nét chữ hơn. Sự khác biệt này cho thấy rằng các mô hình có thể thể hiện hiệu quả khác nhau tùy thuộc vào đặc điểm của từng bộ dữ liệu.

5.2 Hướng phát triển

Nhìn chung, bài toán xác minh chữ ký vẫn còn nhiều thử thách, đặc biệt khi dữ liệu chữ ký có sự đa dạng cao về phong cách viết, chất lượng chữ ký hoặc thậm chí là sự giả mạo chữ ký tinh vi. Trong tương lai, nhóm em sẽ nghiên cứu và phát triển thêm mô hình để cải thiện độ chính xác cũng như giải quyết hiệu quả bài toán xác minh chữ ký.

Đối với Signet, nhóm em sẽ tìm hiểu và áp dụng các kỹ thuật tăng cường dữ liệu như xoay, nghiêng, và thay đổi tỷ lệ chữ ký nhằm giúp mô hình thích nghi tốt hơn với sự đa dạng về phong cách viết và cải thiện khả năng phát hiện chữ ký giả mạo tinh vi. Bên cạnh đó, tối ưu hóa kiến trúc mạng bằng cách sử dụng các biến thể như EfficientNet, giúp giảm chi phí tính toán mà vẫn duy trì hiệu suất cao. Ngoài ra, có thể áp dụng unsupervised learning để trích xuất đặc trưng từ dữ liệu hạn chế cũng là một hướng phát triển quan trọng nhằm tăng khả năng tổng quát hóa của mô hình.

Với SigmML, nhóm sẽ thử nghiệm các hàm loss của metric learning như Circle Loss hoặc các biến thể cải tiến của Triplet Loss để tối ưu hóa khoảng cách giữa chữ ký thật và giả. Đồng thời, việc tích hợp thêm các đặc trưng hình học của nét viết sẽ giúp tăng cường khả năng phân loại.

Nhóm cũng sẽ tiếp tục nghiên cứu các phương pháp khác cũng như các bước tiền xử lý nâng cao để cải thiện chất lượng hình ảnh đầu vào. Những định hướng này được kỳ vọng sẽ góp phần giải quyết tốt hơn bài toán xác minh chữ ký viết tay, nâng cao độ chính xác và độ tin cậy để có thể ứng dụng vào thực tế.

References

[1] F. Leclerc and R. Plamondon. "Automatic signature verification: The state of the art". In: *Pattern Recognition* 22.2 (1989), pp. 107–131.

[2] D. Impedovo and G. Pirlo. "Automatic signature verification: The state of the art". In: *IEEE Trans. Syst., Man, Cybern. C* 38.5 (2008), pp. 609–635.

[3] L. Hafemann, R. Sabourin, and L. Oliveira. "Offline handwritten signature verification - literature review". In: *7th International Conference on Image Processing Theory, Tools and Applications (IPTA)*. 2017.

[4] L. Hafemann, R. Sabourin, and L. Oliveira. "Learning features for offline handwritten signature verification using deep convolutional neural networks". In: *Pattern Recognition* 38.5 (2017), pp. 609–635.

[5] H. Baltzakis and N. Papamarkos. "A new signature verification technique based on a two-stage neural network classifier". In: *Engineering Applications of Artificial Intelligence* 14.1 (Feb. 2001), pp. 95–103.

[6] A. El-Yacoubi et al. "Offline signature verification using HMMs and cross-validation". In: *Proc. IEEE Signal Processing Society Workshop*. Vol. 2. 2000.

[7] E. Justino et al. "An off-line signature verification system using HMM and graphometric features". In: *4th IAPR International Workshop on Document Analysis Systems (DAS)*. Rio de Janeiro, 2000, pp. 211–222.

[8] J.-P. Drouhard, R. Sabourin, and M. Godbout. "A neural network approach to off-line signature verification using directional pdf". In: *Pattern Recognition* 29.3 (1996), pp. 415–424.

[9] B. Zhang. "Off-line signature verification and identification by pyramid histogram of oriented gradients". In: *International Journal of Intelligent Computing and Cybernetics* 3.4 (2010), pp. 611–630.

[10] M. Yilmaz et al. "Offline signature verification using classifier combination of hog and lbp features". In: *Biometrics (IJCB), 2011 International Joint Conference on*. IEEE, 2011, pp. 1–7.

[11] M. Yilmaz and B. Yanikoglu. "Score level fusion of classifiers in offline signature verification". In: *Information Fusion* 32 (2016), pp. 109–119.

[12] J. Hu and Y. Chen. "Offline signature verification using real adaboost classifier combination of pseudo-dynamic features". In: *Document Analysis and Recognition, 12th International Conference on*. Aug. 2013, pp. 1345–1349.

[13] J. Vargas et al. "Offline signature verification based on grey level information using texture features". In: *Pattern Recognition* 44.2 (Feb. 2011), pp. 375–385.

[14] L. Oliveira et al. "The graphology applied to signature verification". In: *12th Conference of the International Graphonomics Society*. 2005, pp. 286–290.

[15] B. Ribeiro et al. “Deep Learning Networks for Off-line Handwritten Signature Recognition”. In: *Springer*. 2011, pp. 523–532.

[16] H. Khalajzadeh, M. Mansouri, and M. Teshnehlab. “Persian signature verification using convolutional neural networks”. In: *International Journal of Engineering Research and Technology* 1 (2012).

[17] L. Hafemann, R. Sabourin, and L. Oliveira. “Writer-independent feature learning for offline signature verification using deep convolutional neural networks”. In: *International Joint Conference on Neural Networks*. July 2016, pp. 2576–2583.

[18] Z. Zhang, X. Liu, and Y. Cui. “Multi-phase offline signature verification system using deep convolutional generative adversarial networks”. In: *9th International Symposium on Computational Intelligence and Design (ISCID)*. Vol. 2. 2016, pp. 103–107.

[19] I. Goodfellow et al. “Generative adversarial nets”. In: *Advances in Neural Information Processing Systems (NIPS)*. Vol. 27. 2014, pp. 2672–2680.

[20] H. Rantzsch et al. “Signature embedding: Writer independent offline signature verification with deep metric learning”. In: *Advances in Visual Computing*. Vol. 10073. Springer, Dec. 2016, pp. 616–625.

[21] Sounak Dey et al. “SigNet: Convolutional Siamese Network for Writer Independent Offline Signature Verification”. In: *CoRR* abs/1707.02131 (2017). arXiv: 1707.02131. URL: http://arxiv.org/abs/1707.02131.

[22] Elias N. Zois et al. “A Comprehensive Study of Sparse Representation Techniques for Offline Signature Verification”. In: *IEEE Transactions on Biometrics, Behavior, and Identity Science* 1.1 (2019), pp. 68–81. DOI: 10.1109/TBIOM.2019.2901093.

[23] Zhi Gao et al. “Learning to Optimize on SPD Manifolds”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 7697–7706. DOI: 10.1109/CVPR42600.2020.00772.

[24] Meenakshi K. Kalera, Sargur Srihari, and Aihua Xu. “Offline signature verification and identification using distance statistics”. In: *International Journal of Pattern Recognition and Artificial Intelligence* 18.07 (Nov. 2004), pp. 1339–1360.

[25] Srikanta Pal et al. “Performance of an Off-Line Signature Verification Method Based on Texture Features on a Large IndicScript Signature Dataset”. In: *12th IAPR Workshop on Document Analysis Systems (DAS)*. Apr. 2016, pp. 72–77.

Bảng phân công		
MSSV	Họ tên	Nội dung công việc
21520380	Hồ Yến Nhi	1. Tìm hiểu bài toán 2. Tìm hiểu dataset 3. Tìm hiểu SigNet 4. Demo 5. Làm slide 6. Viết báo cáo
21521050	Huỳnh Phạm Đức Lâm	1. Tìm hiểu bài toán 2. Tìm hiểu dataset 3. Tìm hiểu SigmML 4. Demo 5. Thuyết trình 6. Viết báo cáo