Nhận diện Biển báo Giao thông bằng Siamese Neural Networks

Đoàn Anh Quân[1], Lê Việt Hoàng[2], Hà Tuấn Kiệt[3]

[1*]Khoa Toán - Tin, Trường Đại học Khoa học Tự nhiên, TP. Hồ Chí Minh, Việt Nam.

Contributing authors: 23110111@student.hcmus.edu.vn; 23110155@student.hcmus.edu.vn; 23110176@student.hcmus.edu.vn;

Tóm tắt nội dung

Bài báo cáo này trình bày một hệ thống nhận diện biển báo giao thông tự động kết hợp giữa các kỹ thuật xử lý ảnh truyền thống và mạng nơ-ron Siamese (Siamese Neural Networks - SNNs). Quy trình gồm hai giai đoạn chính: phát hiện vị trí biển báo bằng phương pháp chuyển đổi không gian màu HSI, nhị phân hóa và tách đường viền; sau đó sử dụng SNNs để nhận diện và phân loại biển báo. Mô hình được huấn luyện trên bộ dữ liệu GTSRB (German Traffic Sign Recognition Benchmark) với 43 loại biển báo và đạt độ chính xác 79.3% trên tập kiểm thử 12.630 ảnh. Kết quả cho thấy sự hiệu quả của việc kết hợp xử lý ảnh cổ điển với học sâu trong bài toán nhận dạng biển báo giao thông thực tế.

Keywords: Nhận dạng biển báo, Siamese Neural Networks, Xử lý ảnh, GTSRB, One-shot learning, Computer vision

1 Giới thiệu

Nhận diện biển báo giao thông tự động là một trong những ứng dụng quan trọng của thị giác máy tính, đặc biệt trong lĩnh vực phát triển các hệ thống hỗ trợ lái xe thông minh (Advanced Driver Assistance Systems - ADAS) và xe tự hành. Việc phát hiện và phân loại chính xác các biển báo giao thông trong điều kiện thực tế là một thách thức lớn do sự đa dạng về điều kiện ánh sáng, góc nhìn, hiện tượng che khuất một phần biển báo, và các yếu tố môi trường khác [1].

Trong những năm gần đây, các phương pháp học sâu (deep learning) đã đạt được những thành công vượt bậc trong lĩnh vực thị giác máy tính [2]. Đặc biệt, Siamese

Neural Networks đã chứng minh hiệu quả trong các bài toán one-shot learning và metric learning, nơi mà số lượng mẫu huấn luyện cho mỗi lớp có thể hạn chế [2].

Trong bài báo cáo này, chúng tôi đề xuất một phương pháp tiếp cận hai giai đoạn: giai đoạn đầu sử dụng các kỹ thuật xử lý ảnh truyền thống dựa trên không gian màu HSI [3] và thuật toán phân ngưỡng Otsu [4] để phát hiện vị trí biển báo trong ảnh; giai đoạn thứ hai áp dụng mạng nơ-ron Siamese để nhận diện và phân loại các biển báo đã được phát hiện. Hệ thống được xây dựng và đánh giá trên bộ dữ liệu GTSRB [1], một trong những bộ dữ liệu chuẩn được sử dụng rộng rãi trong nghiên cứu về nhận dạng biển báo giao thông.

2 Nhận diện Biển báo Giao thông (Detection)

2.1 Phương pháp Phát hiện Vị trí và Tách Biển báo

Biển báo giao thông thường có các màu cơ bản như đỏ, xanh, vàng và được đặt trong môi trường phức tạp với điều kiện ánh sáng thay đổi. Các máy ảnh thông thường sử dụng phổ màu RGB, tuy nhiên kết quả phân đoạn trên RGB dễ bị ảnh hưởng bởi cường độ chiếu sáng [5].

Để giải quyết vấn đề này, chúng tôi chuyển đổi sang không gian màu HSI (Hue, Saturation, Intensity) để giảm thiểu ảnh hưởng của ánh sáng và thời tiết [3]. Không gian màu HSI tách biệt thông tin màu sắc (Hue) và độ bão hòa (Saturation) khỏi thông tin cường độ sáng (Intensity), giúp hệ thống nhận diện màu sắc đặc trưng của biển báo một cách robust hơn trong các điều kiện chiếu sáng khác nhau.

2.1.1 Chuyển đổi từ RGB sang HSI

Các thành phần r, g, b được chuẩn hóa từ R, G, B theo công thức [5]:

$$r = \frac{R}{R+G+B}; \quad g = \frac{G}{R+G+B}; \quad b = \frac{B}{R+G+B} \tag{1}$$

Công thức tính các thành phần H, S, I:

1. Thành phần H (Hue - Góc màu):

$$H = \begin{cases} \theta & \text{nếu } b \leq g \\ 2\pi - \theta & \text{nếu } b > g \end{cases} \tag{2}$$

Trong đó góc θ được tính bằng:

$$\theta = \arccos\left(\frac{\frac{1}{2}[(r-g)+(r-b)]}{\sqrt{(r-g)^2+(r-b)(g-b)}}\right)$$

2. Thành phần S (Saturation - Độ bão hòa):

$$S = 1 - 3 \times \min(r, g, b) \tag{3}$$

3. Thành phần I (Intensity - Cường độ sáng):

$$I = \frac{R+G+B}{3 \times 255} \tag{4}$$

Hình 1: Ảnh gốc trong không gian màu RGB

Hình 2: Ảnh sau khi chuyển sang không gian màu HSI

Hình 1 và 2 minh họa sự khác biệt giữa biểu diễn RGB và HSI của cùng một ảnh đầu vào, cho thấy không gian màu HSI giúp tách biệt thông tin màu sắc khỏi cường độ sáng.

2.2 Nhị phân hóa (Binarization)

Nhị phân hóa là quá trình biến đổi ảnh xám thành ảnh nhị phân, trong đó các điểm ảnh chỉ có hai giá trị: 0 (Đen) và 255 (Trắng) [5]. Nguyên lý thực hiện so sánh cường độ sáng $I(x, y)$ với ngưỡng T:

- Nếu $I(x, y) \leq T \rightarrow I_{NP}(x, y) = 0$ (Đen).
- Nếu $I(x, y) > T \rightarrow I_{NP}(x, y) = 255$ (Trắng).

Dự án sử dụng 2 phương pháp kết hợp để lọc biển báo:

2.2.1 Phương pháp 1: Dựa trên kênh Saturation (S)

Kênh S thể hiện độ "thuần khiết"hoặc "rực rỡ"của màu sắc. Biển báo giao thông thường có màu sắc rất rõ ràng và bão hòa cao, trong khi nền xung quanh thường có màu nhạt hơn. Thuật toán Otsu [4] sẽ tự động phân tích biểu đồ (Histogram) của ảnh để tìm ra ngưỡng tối ưu nhất giúp tách biệt hai vùng:

- **Vùng nền (Background):** Độ bão hòa thấp (xám, nhạt).
- **Vùng đối tượng (Foreground):** Độ bão hòa cao (đậm màu, rực rỡ).

Thuật toán Otsu tìm ngưỡng tối ưu bằng cách tối đa hóa phương sai giữa các lớp (between-class variance) [4].

```
# Phan tich histogram de tim nguong toi uu
_, binary_s = cv.threshold(s_channel, 0, 255,
                           cv.THRESH_BINARY + cv.THRESH_OTSU)
```

Listing 1: Phân ngưỡng kênh S với thuật toán Otsu

Kết quả là ảnh đen trắng, trong đó màu trắng tương ứng với vùng có màu sắc rõ ràng (khả năng cao là biển báo), như được minh họa trong Hình 3.

Hình 3: Kết quả nhị phân hóa dựa trên kênh Saturation với thuật toán Otsu

2.2.2 Phương pháp 2: Dựa trên màu sắc (Color Thresholding)

Ngoài phương pháp dựa trên độ bão hòa, chúng tôi áp dụng các ngưỡng cụ thể trên các kênh H, S, I để lọc ra các màu đặc trưng của biển báo (Đỏ, Vàng, Xanh, Đen) theo đề xuất của Sugiharto và cộng sự [3]. Các giá trị ngưỡng được trình bày trong Bảng 1.

Bảng 1: Giá trị ngưỡng cho các màu đặc trưng của biển báo giao thông [3]

Màu sắc	**Giá trị H**	**Giá trị S**	**Giá trị I**
Đỏ (Red)	$H \leq 10$ hoặc $240 \leq H$	$S \geq 30$	-
Vàng (Yellow)	$18 \leq H \leq 45$	$S \geq 148$	-
Xanh (Blue)	$120 < H \leq 175$	$S \geq 127$	-
Đen (Black)	-	$S \leq 50$	$I \leq 60$

Kết quả của phương pháp color thresholding được thể hiện trong Hình 4, cho thấy các vùng màu đặc trưng của biển báo được tách biệt rõ ràng.

Hình 4: Kết quả nhị phân hóa dựa trên color thresholding với các ngưỡng HSI

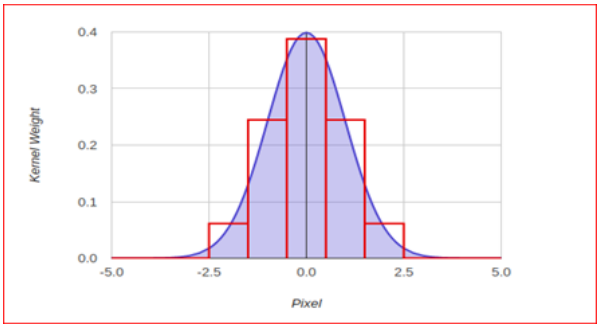
2.3 Giảm nhiễu (Noise Reduction)

Sau quá trình nhị phân hóa, ảnh thường chứa nhiễu dưới dạng các chấm hạt nhỏ phân bố ngẫu nhiên làm biến dạng chi tiết. Để xử lý vấn đề này, chúng tôi sử dụng Bộ lọc Gauss (Gaussian Filter) [5], một trong những bộ lọc làm mượt hiệu quả nhất trong xử lý ảnh.

Bộ lọc Gauss được thực hiện bằng cách nhân chập ảnh đầu vào với một ma trận lọc (kernel) có trọng số tuân theo phân phối Gaussian. Ý tưởng chính là:

- Giá trị điểm ảnh trung tâm có trọng số lớn nhất.
- Các điểm ảnh càng xa trung tâm thì trọng số càng giảm dần theo hàm Gauss (phân phối chuẩn).

Điều này giúp làm mượt ảnh, loại bỏ nhiễu nhưng vẫn giữ lại các đặc trưng quan trọng của biển báo [5]. Hình 5 minh họa hai ma trận lọc Gauss với kích thước khác nhau.



Hình 5: Ví dụ về hai ma trận lọc Gauss với kích thước 3×3 và 5×5

2.4 Tách biển báo với Contours

Để xác định chính xác biên của biển báo, chúng tôi sử dụng thuật toán Suzuki's Tracing [6] thông qua hàm `findContours` trong thư viện OpenCV. Phương pháp này có ưu điểm là phân biệt được đường biên ngoài (Outer) và đường biên trong (Hole), đồng thời xây dựng được cấu trúc phân cấp của các đường viền lồng nhau.

Cú pháp hàm trong OpenCV:

```
findContours(
    InputOutputArray image,
    OutputArrayOfArrays contours,
    OutputArray hierarchy,
    int mode,
    int method,
    Point offset=Point()
)
```

Listing 2: Cú pháp hàm findContours trong OpenCV

Giải thích các tham số chi tiết:

- `image`: Ảnh đầu vào (ảnh nhị phân sau quá trình tiền xử lý).
- `contours`: Danh sách lưu trữ các đường biên tìm được (vector các điểm).
- `hierarchy`: Chứa thông tin phân cấp (số đường viền, quan hệ cha-con giữa các contours).
- `mode`: Chế độ lấy đường viền:
  - `CV_RETR_EXTERNAL`: Chỉ lấy các đường biên bên ngoài cùng.
  - `CV_RETR_LIST`: Lấy tất cả các đường viền tìm được.
  - `CV_RETR_CCOMP`: Lấy tất cả và chia làm 2 cấp (biên ngoài và biên trong).
  - `CV_RETR_TREE`: Lấy tất cả và tạo hệ thống phân cấp đầy đủ các đường lồng nhau.
- `method`: Phương pháp xấp xỉ đường viền:
  - `CV_CHAIN_APPROX_NONE`: Lưu trữ tất cả các điểm của đường viền.
  - `CV_CHAIN_APPROX_SIMPLE`: Nén các đoạn thẳng (ví dụ: hình chữ nhật chỉ lưu 4 đỉnh).
  - `CV_CHAIN_APPROX_TC89_L1`: Áp dụng thuật toán xấp xỉ Teh-Chin.

Sau nhiều bước thử nghiệm và đánh giá, kết luận cho thấy mode `CV_RETR_EXTERNAL` kết hợp với method `CV_CHAIN_APPROX_SIMPLE` là phù hợp nhất với dự án, giúp giảm thiểu nhiễu từ các đường viền nhỏ bên trong.

Kết quả sau bước này là các đường bao hình chữ nhật (bounding box) bao quanh các khu vực nghi ngờ là biển báo, như được minh họa trong Hình 6. Các khu vực này sẽ được trích xuất và đưa vào giai đoạn nhận diện tiếp theo.

Hình 6: Kết quả phát hiện và vẽ bounding box quanh các biển báo được phát hiện

3 Tổng quan về Siamese Neural Networks

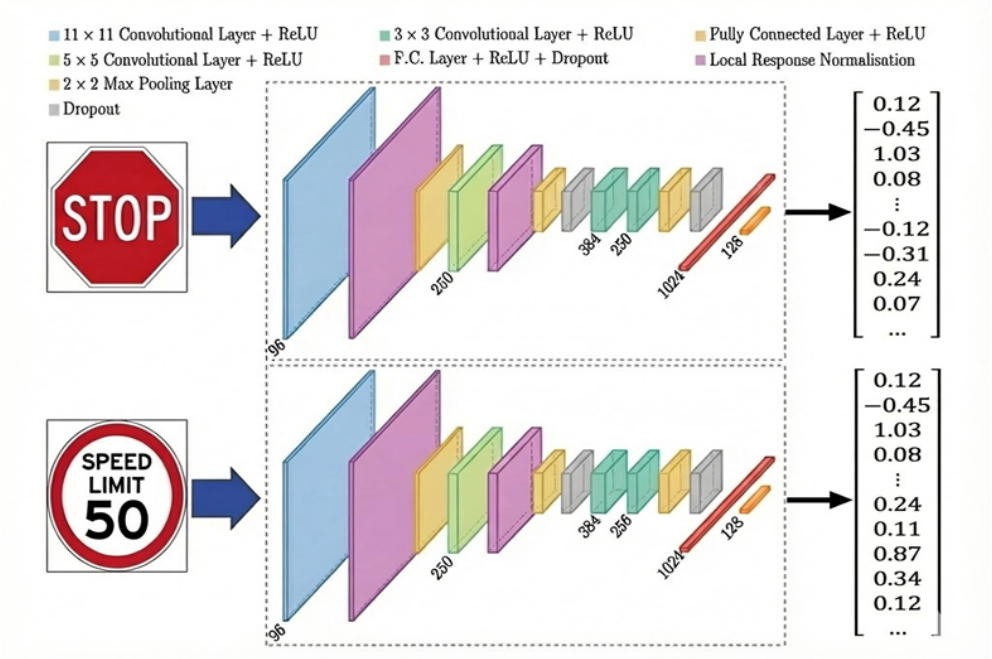
3.1 Giới thiệu chung

Siamese Neural Networks (SNNs) là một lớp kiến trúc mạng nơ-ron đặc biệt, được đề xuất bởi Bromley và cộng sự [2], bao gồm hai hoặc nhiều mạng con giống hệt nhau về cấu hình, tham số và trọng số. Đặc điểm quan trọng nhất của SNNs là các mạng con chia sẻ trọng số (weight sharing), đảm bảo rằng hai đầu vào tương tự sẽ được ánh xạ tới các điểm gần nhau trong không gian đặc trưng.

Mô hình này đã chứng minh hiệu quả vượt trội trong các bài toán One-shot Learning và Few-shot Learning [2], nơi mà số lượng mẫu huấn luyện cho mỗi lớp có thể rất hạn chế. Trong bối cảnh nhận dạng biển báo giao thông, SNNs đặc biệt hữu ích khi cần nhận diện các loại biển báo mới mà không cần huấn luyện lại toàn bộ mô hình.

Hình 7 minh họa kiến trúc tổng quát của một Siamese Network. Điểm khác biệt cơ bản giữa SNNs và các mạng nơ-ron truyền thống nằm ở mục tiêu học:

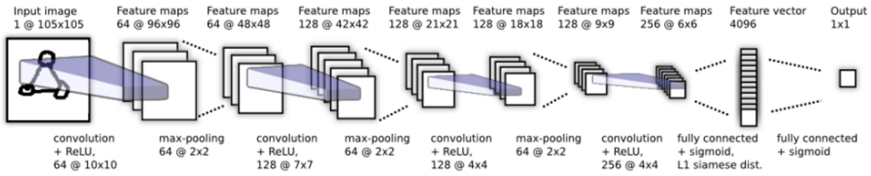
- **Mạng truyền thống (Classification Networks):** Học cách dự đoán để phân loại vào các lớp cụ thể, trong đó số lượng node ở lớp output bằng số lượng lớp (class). Output thường là xác suất thuộc về mỗi lớp.
- **Siamese Networks (Metric Learning):** Số lượng node ở lớp output có thể lớn hơn hoặc nhỏ hơn số lượng lớp, tập trung vào việc học hàm khoảng cách (distance metric) hoặc hàm tương đồng (similarity function) giữa các đầu vào. Output là một vector đặc trưng (embedding vector) cho mỗi đầu vào.



Hình 7: Kiến trúc tổng quát của Siamese Neural Network với hai nhánh mạng chia sẻ trọng số [7]

3.2 Kiến trúc mạng cho One-shot Image Recognition

Trong bài toán nhận diện ảnh one-shot, kiến trúc của Siamese Network được thiết kế để trích xuất đặc trưng từ hình ảnh đầu vào thông qua các lớp tích chập (Convolutional Layers), như được minh họa trong Hình 8.



Hình 8: Kiến trúc chi tiết của Siamese Network cho bài toán One-shot Image Recognition [2]

3.2.1 Đầu vào

Mạng nhận đầu vào là hình ảnh thang độ xám (grayscale) với độ phân giải 105×105 pixels. Việc sử dụng ảnh grayscale giúp giảm số chiều đầu vào và tập trung vào hình dạng, texture của biển báo thay vì phụ thuộc quá nhiều vào màu sắc.

3.2.2 Quá trình trích xuất đặc trưng

Mạng sử dụng chuỗi các lớp tích chập để thu nhỏ kích thước ảnh nhưng tăng độ sâu đặc trưng (feature depth), cho phép mạng học được các biểu diễn ngữ nghĩa cao cấp. Chi tiết các lớp như sau:

- **Lớp 1 (Conv + ReLU + Max Pooling):**
  - Tích chập: 64 bộ lọc kích thước 10×10, stride = 1
  - Hàm kích hoạt: ReLU (Rectified Linear Unit)
  - Max Pooling: Cửa sổ 2×2, stride = 2
  - Kết quả: Feature map kích thước 48×48 với 64 kênh
- **Lớp 2 (Conv + ReLU + Max Pooling):**
  - Tích chập: 128 bộ lọc kích thước 7×7, stride = 1
  - Hàm kích hoạt: ReLU
  - Max Pooling: Cửa sổ 2×2, stride = 2
  - Kết quả: Feature map kích thước 21×21 với 128 kênh
- **Lớp 3 (Conv + ReLU + Max Pooling):**
  - Tích chập: 128 bộ lọc kích thước 4×4, stride = 1
  - Hàm kích hoạt: ReLU
  - Max Pooling: Cửa sổ 2×2, stride = 2
  - Kết quả: Feature map kích thước 9×9 với 128 kênh
- **Lớp 4 (Conv + ReLU - Không có Pooling):**
  - Tích chập: 256 bộ lọc kích thước 4×4, stride = 1
  - Hàm kích hoạt: ReLU
  - Kết quả: Feature map kích thước 6×6 với 256 kênh

3.2.3 Vector đặc trưng (Embedding)

Sau các lớp tích chập, các đặc trưng được làm phẳng (flattening) từ tensor 3D thành vector 1D, sau đó đưa qua lớp kết nối đầy đủ (Fully Connected Layer) để tạo ra một vector đặc trưng (embedding vector) có kích thước 4096 chiều. Vector này đại diện cho toàn bộ thông tin ngữ nghĩa của ảnh đầu vào trong một không gian metric có cấu trúc.

3.3 Hàm mất mát (Loss Function)

SNN sử dụng **Contrastive Loss** (hàm mất mát tương phản) được đề xuất bởi Hadsell và cộng sự [8]. Đây là hàm mất mát dựa trên khoảng cách (distance-based loss), khác với các hàm mất mát dựa trên dự đoán lỗi (prediction error) trong các mạng phân loại truyền thống.

Công thức Contrastive Loss:

$$L = (1 - Y)\frac{1}{2}(D_W)^2 + (Y)\frac{1}{2}\{\max(0, m - D_W)\}^2 \tag{5}$$

Trong đó:

- D_W: Khoảng cách Euclidean giữa hai vector đặc trưng đầu ra $G_W(X_1)$ và $G_W(X_2)$.

$$D_W = \|G_W(X_1) - G_W(X_2)\|_2 = \sqrt{\sum_{i=1}^{n}(G_W(X_1)_i - G_W(X_2)_i)^2}$$

- Y: Nhãn của cặp ảnh. $Y = 0$ nếu hai ảnh thuộc cùng lớp (similar pair), $Y = 1$ nếu hai ảnh thuộc khác lớp (dissimilar pair).
- m: Margin (biên độ), một hyperparameter xác định khoảng cách tối thiểu giữa các cặp khác lớp. Thông thường m được chọn trong khoảng 1.0 đến 2.0 [8].

Hàm mất mát này có hai thành phần:

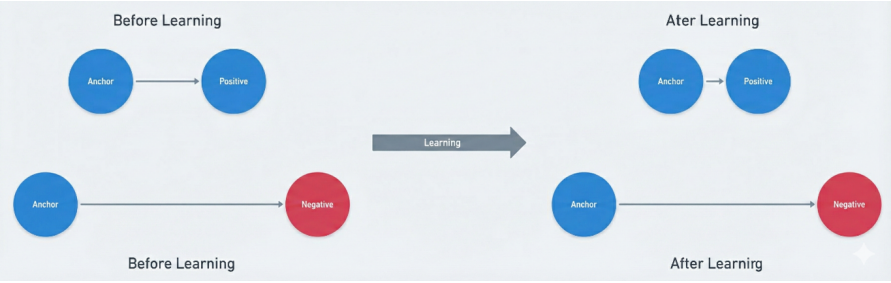
- **Thành phần thứ nhất** $(1 - Y)\frac{1}{2}(D_W)^2$: Tác động khi $Y = 0$ (cùng lớp), khuyến khích mạng giảm khoảng cách giữa các mẫu cùng lớp.
- **Thành phần thứ hai** $Y\frac{1}{2}\{\max(0, m - D_W)\}^2$: Tác động khi $Y = 1$ (khác lớp), khuyến khích mạng tăng khoảng cách giữa các mẫu khác lớp, nhưng chỉ khi khoảng cách nhỏ hơn margin m.

3.4 Tối ưu hóa và Quy trình huấn luyện

Quy trình huấn luyện Siamese Network diễn ra theo các bước sau:

1. **Khởi tạo:** Khởi tạo mạng với trọng số ngẫu nhiên (thường sử dụng Xavier initialization hoặc He initialization), khởi tạo hàm mất mát Contrastive Loss và bộ tối ưu hóa (optimizer).
2. **Forward pass - Nhánh 1:** Truyền ảnh thứ nhất X_1 của cặp ảnh qua mạng để thu được vector đặc trưng $G_W(X_1)$.
3. **Forward pass - Nhánh 2:** Truyền ảnh thứ hai X_2 của cặp ảnh qua mạng (sử dụng chung trọng số với nhánh 1) để thu được vector đặc trưng $G_W(X_2)$.
4. **Tính toán Loss:** Tính toán khoảng cách D_W giữa hai vector đặc trưng, sau đó tính Contrastive Loss dựa trên nhãn Y của cặp ảnh.
5. **Backpropagation:** Lan truyền ngược để tính toán gradient của loss theo các tham số mạng $\frac{\partial L}{\partial W}$.
6. **Cập nhật trọng số:** Sử dụng bộ tối ưu hóa (SGD, Adam, hoặc RMSprop) để cập nhật trọng số theo hướng giảm loss: $W \leftarrow W - \eta\frac{\partial L}{\partial W}$, trong đó η là learning rate.

Hình 9 minh họa trực quan quá trình học: Trước khi học, khoảng cách giữa Anchor-Positive (cùng lớp) và Anchor-Negative (khác lớp) chưa được phân tách rõ ràng. Sau khi học, các điểm Anchor và Positive được kéo gần nhau hơn trong không gian embedding, trong khi các điểm Anchor và Negative bị đẩy ra xa nhau, tạo thành các cluster rõ ràng.



Hình 9: Minh họa quá trình học của Siamese Network: trước và sau khi huấn luyện, các embedding của cùng class được kéo gần nhau trong khi các embedding khác class bị đẩy xa nhau

4 Thực hành và Cài đặt

Dựa trên cơ sở lý thuyết đã trình bày, hệ thống được hiện thực hóa bằng ngôn ngữ Python 3.8 sử dụng thư viện PyTorch 1.9. Mã nguồn đầy đủ, bao gồm cả code huấn luyện và inference, có thể truy cập tại: https://github.com/surmullet/project_PR

4.1 Chuẩn bị dữ liệu

Để huấn luyện mạng Siamese, dữ liệu cần được tổ chức thành các cặp ảnh (image pairs) thay vì các ảnh đơn lẻ như trong các bài toán phân loại truyền thống. Bộ dữ liệu được lựa chọn cho nghiên cứu này là GTSRB.

4.1.1 Bộ dữ liệu GTSRB

GTSRB (German Traffic Sign Recognition Benchmark) là một bộ dữ liệu chuẩn mực lớn về nhận diện biển báo giao thông, được giới thiệu bởi Stallkamp và cộng sự tại hội nghị IJCNN 2011 [1]. Đây là bộ dữ liệu được sử dụng rộng rãi trong cộng đồng nghiên cứu để đánh giá và so sánh các thuật toán thị giác máy tính nhờ tính đa dạng, quy mô lớn và tính thực tế cao.

Hình 10 trình bày toàn bộ 43 lớp biển báo trong GTSRB. Các đặc điểm chính của bộ dữ liệu bao gồm:

- **Số lượng lớp (Classes):** Bộ dữ liệu bao gồm 43 loại biển báo giao thông khác nhau, được phân thành các nhóm chính:
  - Biển báo cấm (Prohibitory signs): Classes 0-18
  - Biển báo nguy hiểm (Danger signs): Classes 19-31
  - Biển báo chỉ dẫn (Mandatory signs): Classes 32-42

  Phân loại này tuân theo tiêu chuẩn Vienna Convention on Road Signs and Signals [1].
- **Quy mô dữ liệu:** Tổng cộng có hơn 50.000 hình ảnh được chia thành:
  - Tập huấn luyện (Training set): 39.209 ảnh
  - Tập kiểm thử (Test set): 12.630 ảnh

Phân bố số lượng mẫu giữa các lớp không đồng đều, dao động từ hơn 200 đến hơn 2.000 mẫu mỗi lớp, phản ánh tần suất xuất hiện thực tế của các loại biển báo [1].

- **Đặc điểm hình ảnh:** Các hình ảnh được cắt (cropped) từ các đoạn video quay thực tế trên đường phố ở Đức. Do đó, dữ liệu chứa đựng nhiều thách thức thực tế:
  - *Điều kiện ánh sáng thay đổi mạnh*: ngược sáng (backlit), bóng râm, buổi tối, sương mù
  - *Biến dạng hình học*: do góc quay camera, góc nhìn nghiêng (perspective distortion)
  - *Motion blur*: hình ảnh bị mờ do xe di chuyển với tốc độ cao
  - *Kích thước không đồng nhất*: dao động từ 15×15 đến 250×250 pixels
  - *Che khuất một phần*: do cây cối, các phương tiện khác
  - *Biển báo bị phai màu hoặc hư hỏng*: do thời tiết và thời gian [1]

Hình 10: Tổng quan 43 loại biển báo giao thông trong bộ dữ liệu GTSRB, bao gồm biển cấm, biển nguy hiểm và biển chỉ dẫn [1]

4.1.2 Tiền xử lý (Preprocessing)

Tất cả hình ảnh đầu vào đều trải qua quy trình biến đổi (transform) thống nhất để đảm bảo tính nhất quán:

- **Resize:** Đưa về kích thước cố định 105×105 pixels bằng phép nội suy song tuyến tính (bilinear interpolation).

- **ToTensor:** Chuyển đổi từ numpy array sang dạng Tensor của PyTorch với shape (C, H, W).
- **Normalize:** Chuẩn hóa với mean=[0.5, 0.5, 0.5] và std=[0.5, 0.5, 0.5] để đưa giá trị pixel về khoảng $[-1, 1]$, giúp quá trình huấn luyện ổn định và hội tụ nhanh hơn.

4.1.3 Tạo cặp dữ liệu (Pair Generation)

Class `Traffic_sign_dataset` được thiết kế để sinh ra các cặp ảnh huấn luyện theo hai loại:

- **Positive Pairs (Cặp dương):** Hai ảnh thuộc cùng một lớp (cate1 == cate2). Nhãn $Label = 0$. Các cặp này giúp mạng học cách kéo các embedding của cùng lớp lại gần nhau.
- **Negative Pairs (Cặp âm):** Hai ảnh thuộc hai lớp khác nhau (cate1 $\neq$ cate2). Nhãn $Label = 1$. Các cặp này giúp mạng học cách đẩy các embedding của khác lớp ra xa nhau.
- **Cân bằng dữ liệu:** Để đảm bảo tỷ lệ cân bằng giữa positive và negative pairs (thường là 1:1), code sử dụng `random.choices` để lấy mẫu ngẫu nhiên với xác suất có trọng số từ các lớp khác nhau.

Bên cạnh đó, class `DataForKnn` được sử dụng để tải dữ liệu theo dạng phân lớp truyền thống (single images with labels) phục vụ cho quá trình đánh giá bằng thuật toán K-Nearest Neighbors sau khi trích xuất features.

4.2 Cài đặt Mô hình

Kiến trúc mạng được định nghĩa trong class `Siamese` với 4 khối tích chập (Convolutional Blocks) và Batch size là 128:

- **Block 1:** Conv2d (64 filters, 10×10) $\rightarrow$ BatchNorm2d [9] $\rightarrow$ ReLU $\rightarrow$ MaxPool2d (2×2).
- **Block 2:** Conv2d (128 filters, 7×7) $\rightarrow$ BatchNorm2d $\rightarrow$ ReLU $\rightarrow$ MaxPool2d (2×2).
- **Block 3:** Conv2d (128 filters, 4×4) $\rightarrow$ BatchNorm2d $\rightarrow$ ReLU $\rightarrow$ MaxPool2d (2×2).
- **Block 4:** Conv2d (256 filters, 4×4) $\rightarrow$ BatchNorm2d $\rightarrow$ ReLU.
- **Adaptive Pooling:** `AdaptiveAvgPool2d` để đảm bảo output có kích thước cố định bất kể input size.
- **Embedding Layer:** Sau khi flatten, vector đặc trưng được đưa qua lớp Linear để tạo ra đầu ra cuối cùng có kích thước **300 chiều** (thay vì 4096 như kiến trúc gốc [2] để tối ưu hóa bộ nhớ và tốc độ tính toán).

Batch Normalization [9] được áp dụng sau mỗi lớp tích chập để:

- Chuẩn hóa phân phối activation, giúp gradient flow tốt hơn
- Tăng tốc độ hội tụ
- Có hiệu ứng regularization nhẹ, giảm overfitting

4.3 Hàm mất mát và Tối ưu hóa

- **Loss Function:** Sử dụng `ContrastiveLoss` với biên độ $margin = 10.0$. Công thức tính khoảng cách dựa trên `F.pairwise_distance` (khoảng cách Euclidean) được implement trong PyTorch.
- **Optimizer:** Sử dụng thuật toán SGD (Stochastic Gradient Descent) với:
  - Learning rate: $lr = 0.01$
  - Momentum: 0.9 để tăng tốc hội tụ và giảm dao động
  - Weight decay: $1e^{-4}$ (L2 regularization) để ngăn chặn overfitting
- **Learning Rate Scheduler:** Sử dụng `ReduceLROnPlateau` để tự động giảm learning rate khi loss không cải thiện:
  - Patience: 20 epochs (đợi 20 epochs trước khi giảm lr)
  - Factor: 0.5 (giảm lr xuống một nửa)
  - Min lr: $1e^{-6}$ (không giảm xuống dưới ngưỡng này)

4.4 Quy trình Huấn luyện và Đánh giá

Điểm đặc biệt trong file `train_script.py` là phương pháp đánh giá kết hợp (Hybrid Approach): Sử dụng Siamese Network như một feature extractor và thuật toán KNN như một classifier.

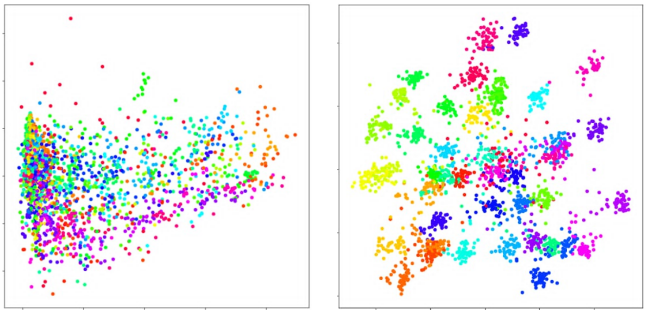
1. **Bước 1 - Train Siamese Network:**
   - Mạng Siamese được huấn luyện 200 epochs trên các cặp ảnh
   - Mục tiêu: học cách nhúng (embed) các ảnh vào không gian metric sao cho ảnh cùng lớp nằm gần nhau và khác lớp nằm xa nhau
   - Early stopping được áp dụng nếu validation loss không cải thiện sau 30 epochs
2. **Bước 2 - Feature Extraction:**
   - Sử dụng mạng Siamese đã huấn luyện (chế độ `eval()`) như một feature extractor
   - Chuyển đổi toàn bộ tập dữ liệu huấn luyện và kiểm thử thành các vector đặc trưng 300 chiều
   - Các vector này được normalize để có unit length (L_2 norm = 1)
3. **Bước 3 - KNN Classification:**
   - Một bộ phân loại K-Nearest Neighbors [10] với $k = 5$ được huấn luyện trên các vector đặc trưng của tập Train
   - KNN tìm 5 láng giềng gần nhất trong không gian embedding dựa trên khoảng cách Euclidean
   - Dự đoán nhãn bằng voting: lớp có nhiều láng giềng nhất sẽ được chọn
   - Độ chính xác (Accuracy) được tính toán trên tập Validation/Test

Phương pháp hybrid này có ưu điểm:

- Tận dụng khả năng học representation mạnh mẽ của Siamese Networks
- Đơn giản, dễ triển khai và không cần huấn luyện lại khi thêm class mới
- Hiệu quả tốt với dữ liệu không cân bằng

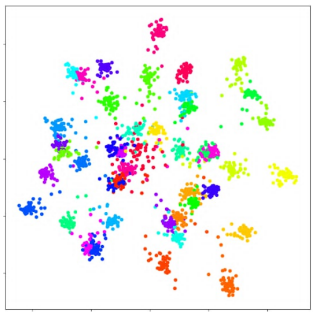
4.5 Trực quan hóa

Để kiểm chứng độ hiệu quả của việc phân cụm trong không gian embedding, mã nguồn sử dụng thuật toán PCA (Principal Component Analysis) [11] để giảm chiều dữ liệu từ 300 chiều xuống 2 chiều, sau đó vẽ biểu đồ phân tán (Scatter plot) tại các thời điểm khác nhau trong quá trình huấn luyện (epoch 1, 5, 10).

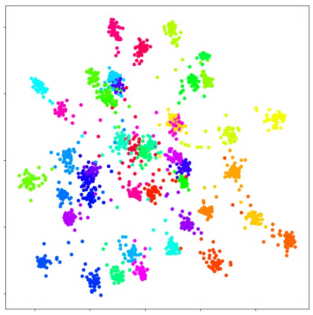


(a) Before training

(b) Epoch 1



(c) Epoch 5



(d) Epoch 10

Legend: màu sắc tương ứng với 43 classes

Hình 11: Quá trình học của Siamese Network được trực quan hóa bằng PCA: các điểm dữ liệu dần tạo thành các cluster rõ ràng theo lớp qua các epoch

Hình 11 cho thấy:

- **Before training** (Hình 11a): Các điểm phân tán ngẫu nhiên, không có cấu trúc cluster rõ ràng

- **Epoch 1** (Hình 11b): Bắt đầu xuất hiện một số nhóm nhỏ
- **Epoch 5** (Hình 11c): Các cluster trở nên rõ ràng hơn, ranh giới giữa các lớp được hình thành
- **Epoch 10** (Hình 11d): Các cluster được phân tách tốt, các điểm cùng lớp tập trung chặt chẽ

Các điểm dữ liệu được tô màu theo lớp để dễ dàng quan sát sự tách biệt giữa các nhóm biển báo. Kết quả trực quan này xác nhận rằng Siamese Network đã học được một không gian embedding có ý nghĩa, trong đó các biển báo cùng loại được nhóm lại gần nhau.

5 Kết quả

5.1 Kết quả Detection

Từ quá trình detection được mô tả trong Section 2, hệ thống tạo ra một danh sách các khu vực quan tâm (Region of Interest - ROI) có khả năng cao là biển báo. Hình 6 đã minh họa các bounding box được vẽ quanh các vùng phát hiện.

5.2 Kết quả Recognition

Các ROI được trích xuất từ giai đoạn detection sẽ được resize về 105×105 pixels và đưa vào mô hình Siamese Network đã được huấn luyện để nhận diện. Mô hình sẽ tính toán embedding vector cho ROI, sau đó sử dụng KNN để tìm lớp gần nhất trong 43 classes của GTSRB.

Hình 12: Ví dụ kết quả nhận diện thành công: biển báo Stop (class 14) được phát hiện và nhận diện chính xác

Hình 15 cho thấy một ví dụ thành công: biển báo Stop (class 14) được hệ thống phát hiện chính xác vị trí và phân loại đúng. Bounding box màu xanh lá cây bao quanh biển báo cùng với nhãn class và confidence score được hiển thị.

Một số kết quả khác

Hình 13: Ảnh thực nghiệm (1)

(a) Background (class 43)

(b) Background (class 43)

(c) Cấm đi ngược chiều (class 17)

Hình 14: Kết quả cho thực nghiệm (1)

Hình 15: Ảnh thực nghiệm (2)

(a) Background (class 43) (b) Rẽ trái (class 34) (c) Dừng lại (class 14)

Hình 16: Kết quả cho thực nghiệm (2)

5.3 Đánh giá định lượng

Đánh giá được thực hiện trên toàn bộ tập test GTSRB gồm 12.630 bức ảnh. Các metrics được tính toán bao gồm precision, recall, F1-score và accuracy.

Bảng 2 trình bày kết quả đánh giá chi tiết. Các điểm chính cần lưu ý:

- **Overall Accuracy:** Hệ thống đạt độ chính xác tổng thể là **79.3%**, cho thấy khả năng nhận diện tốt trong điều kiện thực tế với dữ liệu đa dạng.
- **Macro Average:**
  - Precision: 0.839 - trung bình các lớp có precision khá cao

Bảng 2: Kết quả đánh giá chi tiết trên tập test GTSRB (12.630 ảnh)

Metric	Precision	Recall	F1-score	Support
Accuracy	0.793	0.793	0.793	0.793
Macro avg	0.839	0.828	0.806	12630
Weighted avg	0.911	0.793	0.823	12630

  - Recall: 0.828 - khả năng phát hiện đúng các mẫu thuộc mỗi lớp
  - F1-score: 0.806 - điểm cân bằng giữa precision và recall

- **Weighted Average:**
  - Precision: **0.911** - rất cao, cho thấy mô hình hoạt động tốt trên các lớp có nhiều mẫu huấn luyện
  - Recall: 0.793 - tương ứng với overall accuracy
  - F1-score: 0.823 - cao hơn macro average, phản ánh sự không cân bằng dữ liệu

Sự chênh lệch giữa macro average (0.806) và weighted average (0.823) cho thấy mô hình có xu hướng hoạt động tốt hơn trên các lớp phổ biến (có nhiều mẫu) so với các lớp hiếm (ít mẫu). Đây là vấn đề phổ biến trong machine learning với dữ liệu không cân bằng [1].

6 Thảo luận

6.1 Phân tích kết quả

Kết quả thực nghiệm cho thấy phương pháp kết hợp giữa xử lý ảnh truyền thống và Siamese Neural Networks mang lại hiệu quả khả quan trong bài toán nhận diện biển báo giao thông. Độ chính xác 79.3% là kết quả chấp nhận được, đặc biệt khi xét đến các thách thức của dữ liệu thực tế như điều kiện ánh sáng thay đổi, góc nhìn nghiêng, và motion blur [1].

Điểm mạnh của phương pháp:

- **Giai đoạn Detection:** Sử dụng không gian màu HSI [3] giúp giảm thiểu ảnh hưởng của điều kiện ánh sáng một cách hiệu quả. Kết hợp giữa Saturation-based thresholding và Color-based thresholding tạo ra một pipeline robust.
- **Giai đoạn Recognition:** Siamese Networks [2] cho phép học được các đặc trưng phân biệt tốt, như được thể hiện qua visualization trong Hình 11. Việc sử dụng Contrastive Loss [8] giúp tạo ra không gian embedding có cấu trúc rõ ràng.
- **Tính mở rộng:** Phương pháp hybrid (Siamese + KNN) cho phép dễ dàng thêm các loại biển báo mới mà không cần huấn luyện lại toàn bộ mô hình [10].

6.2 Hạn chế

Một số hạn chế của hệ thống đã được xác định:

- **Độ chính xác thấp trên các lớp hiếm:** Sự chênh lệch giữa macro và weighted F1-score cho thấy mô hình gặp khó khăn với các lớp có ít mẫu huấn luyện. Điều này có thể được cải thiện bằng data augmentation hoặc class balancing techniques.
- **Phụ thuộc vào Detection:** Hiệu năng tổng thể phụ thuộc mạnh vào chất lượng của giai đoạn detection. Nếu detection sai (false positive hoặc false negative), recognition sẽ không thể sửa chữa.
- **Biển báo bị che khuất:** Độ chính xác giảm đáng kể với các biển báo bị che khuất một phần hoặc bị hư hỏng do thời tiết.
- **Real-time performance:** Pipeline hiện tại chưa được tối ưu cho xử lý real-time. Tốc độ xử lý cần được cải thiện để triển khai trên xe tự hành.

6.3 So sánh với các phương pháp khác

Kết quả 79.3% của chúng tôi có thể được so sánh với các nghiên cứu khác trên GTSRB:

- Stallkamp et al. (2011) [1]: 98.98% sử dụng CNN nhiều cột (multi-column CNN) - state-of-the-art tại thời điểm đó
- Các phương pháp deep learning hiện đại: 99+% với ResNet, EfficientNet

Mặc dù độ chính xác của chúng tôi thấp hơn, phương pháp có ưu điểm về tính đơn giản, dễ triển khai, và khả năng mở rộng với các lớp mới. Đây là một trade-off hợp lý cho các ứng dụng không yêu cầu độ chính xác cực cao.

6.4 Hướng phát triển

Các hướng phát triển trong tương lai có thể bao gồm:

- **Cải thiện Detection:** Thay thế detection pipeline truyền thống bằng các kiến trúc deep learning hiện đại như YOLO, Faster R-CNN, hoặc EfficientDet để tăng độ chính xác và tốc độ.
- **Data Augmentation nâng cao:** Áp dụng các kỹ thuật augmentation như cutout, mixup, AutoAugment để tăng cường độ đa dạng của dữ liệu huấn luyện [1].
- **Ensemble Methods:** Kết hợp nhiều mô hình Siamese với các kiến trúc backbone khác nhau (VGG, ResNet, EfficientNet) để cải thiện độ robust.
- **Attention Mechanisms:** Tích hợp attention modules để mô hình tập trung vào các vùng quan trọng của biển báo, giảm ảnh hưởng của background clutter.
- **Tối ưu hóa cho Real-time:** Áp dụng model compression techniques như quantization, pruning, và knowledge distillation để giảm kích thước mô hình và tăng tốc độ inference.
- **Multi-task Learning:** Huấn luyện mô hình đồng thời cho nhiều task (detection, classification, attribute recognition) để cải thiện hiệu quả tổng thể.

7 Kết luận

Bài báo cáo đã trình bày một hệ thống nhận diện biển báo giao thông hoàn chỉnh, kết hợp giữa các kỹ thuật xử lý ảnh truyền thống (HSI color space transformation [3], Otsu thresholding [4], Gaussian filtering [5]) và mạng nơ-ron Siamese [2] với Contrastive

Loss [8]. Hệ thống đạt độ chính xác 79.3% trên bộ dữ liệu GTSRB [1] với 12.630 ảnh kiểm thử, chứng minh tính khả thi của phương pháp đề xuất.

Các đóng góp chính của nghiên cứu bao gồm:

1. Pipeline detection robust dựa trên không gian màu HSI, hiệu quả với điều kiện ánh sáng thay đổi
2. Áp dụng thành công Siamese Networks cho bài toán nhận dạng biển báo, với khả năng mở rộng cao
3. Phương pháp hybrid (Siamese + KNN) đơn giản nhưng hiệu quả
4. Visualization chi tiết quá trình học của embedding space

Nghiên cứu này góp phần vào lĩnh vực phát triển hệ thống hỗ trợ lái xe thông minh (ADAS), với khả năng ứng dụng thực tế trong các hệ thống cảnh báo an toàn giao thông và xe tự hành. Mặc dù còn một số hạn chế cần khắc phục, phương pháp đề xuất mở ra nhiều hướng nghiên cứu tiếp theo hứa hẹn.

Lời cảm ơn. Trong thời gian thực hiện và hoàn thành bài báo cáo **nhận diện biển báo giao thông bằng Siamese Neural Networks**, nhóm chúng em xin chân thành cảm ơn sự giúp đỡ nhiệt tình của Th.S Lưu Giang Nam, đã hướng dẫn, góp ý, chỉnh sửa, cũng như cung cấp nhiều thông tin quý báu và tạo điều kiện cho nhóm chúng em trong quá trình thực hiện bài báo cáo này. Chúng em cũng xin được gửi lời cảm ơn đặc biệt tới thầy Lê Hoàng Đức đã tận tình hướng dẫn, giúp đỡ chỉ bảo cho chúng em cách xây dựng bài báo cáo cũng như chỉnh sửa, đưa ra những lời khuyên giúp chúng em hoàn thành được bài báo cáo.

Mặc dù nhóm em đã rất cố gắng nhưng do kiến thức và kinh nghiệm còn hạn chế, và đây cũng là lần đầu tiên được thực hiện một báo cáo nghiên cứu hoàn chỉnh, nên báo cáo của nhóm em vẫn còn nhiều thiếu sót trong việc trình bày cũng như phân tích. Chúng em rất mong nhận được sự thông cảm và đóng góp ý kiến của quý thầy cô và các bạn để nhóm có thể hoàn thiện hơn trong các nghiên cứu tiếp theo.

Tuyên bố

- **Tài trợ:** Các tác giả tuyên bố không nhận được hỗ trợ tài chính cụ thể nào cho nghiên cứu này.
- **Xung đột lợi ích:** Các tác giả tuyên bố không có xung đột lợi ích liên quan đến việc công bố bài báo này.
- **Sự sẵn có của dữ liệu và mã nguồn:** Bộ dữ liệu hỗ trợ cho các kết quả của nghiên cứu này có sẵn trong kho lưu trữ GTSRB (https://benchmark.ini.rub.de/). Mã nguồn hiện thực nghiên cứu được cung cấp công khai tại https://github.com/surmullet/project_PR.
- **Đóng góp của tác giả:** Tất cả các tác giả đã đóng góp vào việc lên ý tưởng, thiết kế nghiên cứu, thu thập dữ liệu và phân tích kết quả. Tất cả các tác giả đã đọc và phê duyệt bản thảo cuối cùng.
- **Chấp thuận đạo đức:** Không áp dụng. Nghiên cứu không liên quan đến thử nghiệm trên người hoặc động vật.

Tài liệu

[1] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The german traffic sign recognition benchmark: a multi-class classification competition," *The 2011 International Joint Conference on Neural Networks*, pp. 1453–1460, 2011.

[2] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2, Lille, 2015.

[3] A. Sugiharto and A. Harjoko, "Traffic sign detection based on hog and phog using binary svm and k-nn," in *2016 3rd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, pp. 317–321, IEEE, 2016.

[4] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[5] R. C. Gonzalez and R. E. Woods, *Digital image processing*. Pearson, 2018.

[6] S. Suzuki and K. Abe, "Topological structural analysis of digitized binary images by border following," *Computer vision, graphics, and image processing*, vol. 30, no. 1, pp. 32–46, 1985.

[7] S. Dey, A. Dutta, J. I. Toledo, S. K. Ghosh, J. Lladós, and U. Pal, "Signet: Convolutional siamese network for writer independent offline signature verification," *arXiv preprint arXiv:1707.02131*, 2017.

[8] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, pp. 1735–1742, IEEE, 2006.

[9] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, PMLR, 2015.

[10] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[11] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.