

**BỘ GIÁO DỤC VÀ ĐÀO TẠO    VIỆN KHOA HỌC VÀ CÔNG NGHỆ VIỆT NAM**  
**VIỆN CÔNG NGHỆ THÔNG TIN**



**PHẠM ANH PHƯƠNG**

**NGHIÊN CỨU ỨNG DỤNG PHƯƠNG PHÁP  
MÁY VÉC TƠ TỰA TRONG NHẬN DẠNG  
CHỮ VIỆT VIẾT TAY RỜI RẠC**

**Chuyên ngành:** BẢO ĐẢM TOÁN HỌC CHO MÁY TÍNH VÀ  
HỆ THỐNG TÍNH TOÁN

**Mã số:**                    62 46 35 01

**TÓM TẮT LUẬN ÁN TIẾN SĨ**

**HÀ NỘI – 2010**

**Công trình được hoàn thành tại: Viện Công nghệ Thông tin – Viện Khoa học và Công nghệ Việt Nam.**

**Người hướng dẫn khoa học:**

1. PGS.TS Ngô Quốc Tạo
2. PGS.TS Lương Chi Mai

**Phản biện 1:** PGS.TS Hồ Sĩ Đàm

**Phản biện 2:** PGS.TS Nguyễn Thiện Luận

**Phản biện 3:** PGS.TS Huỳnh Quyết Thắng

Luận án sẽ được bảo vệ trước Hội đồng chấm luận án cấp Nhà nước họp tại: Hội trường Viện Công nghệ Thông tin Hà Nội vào lúc 16 giờ 00 ngày 04 tháng 6 năm 2010

Có thể tìm hiểu luận án tại: Thư viện Quốc gia, Thư viện Viện Công nghệ Thông tin Hà Nội.

## TÀI LIỆU THAM KHẢO

- [10] Bùi Minh Trí (2006), “Quy hoạch toán học”, *Nhà xuất bản Khoa học và kỹ thuật*, Hà nội.
- [11] Christopher J. C. Burges (1998), “A Tutorial on Support Vector Machines for Pattern Recognition”, *Data Mining and Knowledge Discovery*, ISSN:1384-5810, Vol. 2, No. 2, pp. 121-167.
- [12] C. J. C. Burges (1996), “Simplified support vector decision rules”, *Proc. 13th International Conference on Machine Learning*, San Mateo, CA, pp. 71–77.
- [13] J. Platt (1999), “Fast Training of Support Vector Machines Using Sequential Minimal Optimization”, In *Advances in Kernel Methods - Support Vector Learning*, Cambridge, M.A, MIT Press, pp. 185-208.
- [14] Osumi E., Freund R., Girosi F. (1997), “An Improved Training Algorithm for Support Vector Machines”, *Proc IEEE NNSP '97*, pp. 276-285.
- [15] Nguyễn Thị Thanh Tân, Lương Chi Mai (2006), “Phương pháp nhận dạng từ viết tay dựa trên mô hình mạng nơ ron kết hợp với thống kê từ vựng”, *Tạp chí Tin học và Điều khiển học*, Tập 22, số 2, tr. 141-154.

# PHẦN MỞ ĐẦU

## 1. Tính cấp thiết của đề tài

Đến thời điểm này, trên thế giới cũng như ở Việt Nam, bài toán nhận dạng chữ viết tay vẫn còn là vấn đề thách thức lớn đối với các nhà nghiên cứu.

**Tình hình nghiên cứu trên thế giới:** Từ những năm 1990 đến nay, các hệ thống nhận dạng thời gian thực được xây dựng và phát triển trên cơ sở các phương pháp luận phân lớp trong lĩnh vực học máy kết hợp với các kỹ thuật xử lý ảnh một cách hiệu quả. Một số phương pháp học máy tiên tiến như mạng nơ ron, mô hình Markov ẩn, SVM,... đã được các nhà nghiên cứu trong và ngoài nước áp dụng để phát triển các ứng dụng trong lĩnh vực nhận dạng chữ.

**Tình hình nghiên cứu trong nước:** Trong những năm gần đây, lĩnh vực nhận dạng chữ viết tay đã được nhiều nhà nghiên cứu trong nước đặc biệt quan tâm. Một số nhóm nghiên cứu điển hình như: GS.TSKH. Hoàng Kiếm và các cộng sự (2001) ở Đại Học Quốc Gia TPHCM đã cài đặt và thử nghiệm hệ thống nhận dạng chữ số và chữ viết tay rời rạc trên các phiếu xuất nhập cảnh, các tác giả Lê Hoài Bắc và Lê Hoàng Thái (2001) đã nghiên cứu bài toán nhận dạng chữ viết tay dựa trên mạng nơ ron và giải thuật di truyền, nhóm nghiên cứu ở phòng Nhận dạng và Công nghệ Tri thức của Viện Công nghệ Thông tin với nhiều công trình nghiên cứu về nhận dạng chữ viết tay dựa trên mô hình Markov ẩn, mạng nơ ron và SVM, nhóm nghiên cứu của TS. Nguyễn Việt Hà và các cộng sự (2005) ở Đại Học Quốc Gia Hà Nội đã nghiên cứu đề xuất giải pháp mô hình liên mạng nơ ron trong nhận dạng ký tự viết tay tiếng Việt,...

Mặc dù trong nước đã có nhiều kết quả nghiên cứu về nhận dạng chữ viết tay, tuy nhiên các kết quả chủ yếu tập trung vào việc nhận dạng chữ số và chữ cái hệ La Tinh, rất ít công trình nghiên cứu đề xuất các giải pháp cho việc nhận dạng chữ viết tay tiếng Việt.

## 2. Mục tiêu của luận án

- Nghiên cứu các phương pháp nhận dạng chữ viết tay đang được áp dụng rộng rãi trong các hệ thống nhận dạng chữ viết trong và ngoài nước. Trên cơ sở các nghiên cứu này, kế thừa và triển khai ứng dụng vào việc nhận dạng chữ viết tay tiếng Việt.
- Nghiên cứu đề xuất các giải pháp hiệu quả cho việc nhận dạng chữ Việt viết tay rời rạc.
- Nghiên cứu đề xuất các phương pháp trích chọn đặc trưng nhằm tăng độ chính xác nhận dạng chữ viết tay.
- Nghiên cứu cải tiến tốc độ nhận dạng chữ Việt viết tay rời rạc.
- Xây dựng một cơ sở dữ liệu chữ viết tay tiếng Việt phục vụ cho nghiên cứu thực nghiệm.

## 3. Phạm vi và phương pháp nghiên cứu

Luận án giới hạn phạm vi nghiên cứu trong khuôn khổ chữ Việt in viết tay rời rạc. Chữ viết tay rời rạc ở đây được hiểu là các ký tự viết tay tách biệt, giữa phần đầu và phần chữ phải tách rời.

Phương pháp nghiên cứu của luận án dựa trên cơ sở khảo sát, kế thừa các kết quả mới nhất. Từ đó xây dựng, phát triển, cải tiến các thuật toán và kiểm chứng bằng thực nghiệm.

#### 4. Các đóng góp mới của luận án

- Nghiên cứu xây dựng thuật toán nhận dạng chữ viết tay rời rạc theo phương pháp phân lớp SVM với các chiến lược một đối một (OVO) và một đối phần còn lại (OVR). Phân tích, đánh giá ưu và nhược điểm của kỹ thuật phân lớp SVM trong nhận dạng chữ viết tay rời rạc thông qua các kết quả thực nghiệm trên các tập dữ liệu chuẩn USPS, MNIST và dữ liệu chữ viết tay tiếng Việt.
- Đề xuất giải pháp hiệu quả cho bài toán nhận dạng chữ Việt viết tay rời rạc áp dụng phương pháp phân lớp SVM.
- Đề xuất phép biến đổi ảnh hai chiều thành chuỗi đặc trưng hiệu quả cho bài toán nhận dạng chữ viết tay rời rạc và đã chứng minh được tính duy nhất của chuỗi đặc trưng theo phép biến đổi này.
- Cải tiến tốc độ nhận dạng chữ Việt viết tay rời rạc bằng cách áp dụng kỹ thuật giảm số chiều của các vector đặc trưng đầu vào và giảm số vector tựa của các máy phân lớp SVM nhị phân.
- Xây dựng được một cơ sở dữ liệu chữ viết tay tiếng Việt với hơn 100000 mẫu ký tự chữ viết tay rời rạc đã gán nhãn.

#### 5. Bố cục của luận án

Luận án được phân thành ba chương với cấu trúc như sau:

Chương 1 giới thiệu tổng quan về lĩnh vực nhận dạng chữ viết tay, các giai đoạn cơ bản của một hệ nhận dạng chữ viết tay và cuối cùng là phần tổng hợp các phương pháp nhận dạng đã và đang được áp dụng rộng rãi trong lĩnh vực nhận dạng chữ viết tay.

Chương 2 tập trung nghiên cứu triển khai ứng dụng nhận dạng chữ viết tay rời rạc trên cơ sở phân lớp SVM.

Chương 3 tiếp tục nghiên cứu phát triển, đề xuất các giải pháp hiệu quả cho việc nhận dạng chữ Việt viết tay rời rạc.

Cuối cùng là phần kết luận và hướng phát triển của luận án.

## CHƯƠNG 1: TỔNG QUAN VỀ NHẬN DẠNG CHỮ VIẾT TAY

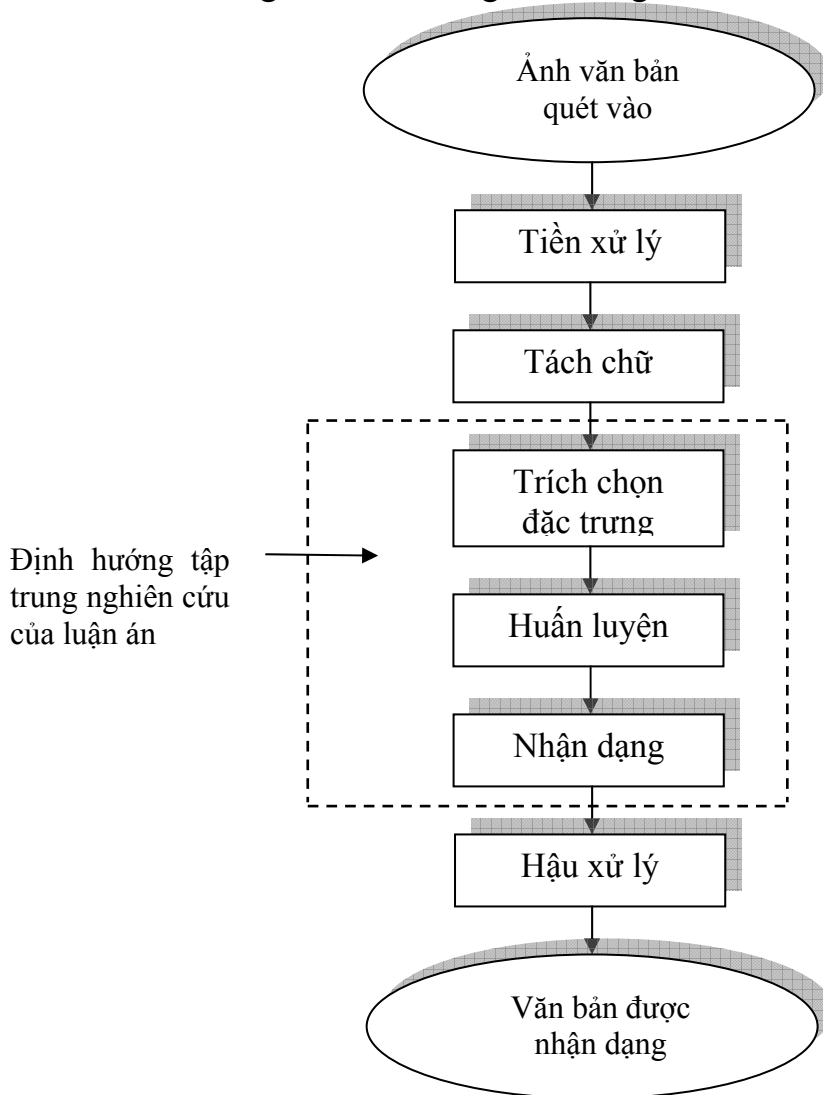
### 1.1. GIỚI THIỆU

Nhận dạng chữ là lĩnh vực được nhiều nhà nghiên cứu quan tâm và cho đến nay lĩnh vực này cũng đã đạt được nhiều thành tựu lớn lao cả về mặt lý thuyết lẫn ứng dụng thực tế. Lĩnh vực nhận dạng chữ được chia làm hai loại: Nhận dạng chữ in và nhận dạng chữ viết tay.

Đến thời điểm này, công nghệ nhận dạng chữ in đã đạt được những giải pháp tốt để ứng dụng vào các sản phẩm thương mại. Tuy nhiên, nhận dạng chữ viết tay vẫn còn là vấn đề thách thức lớn đối với các nhà nghiên cứu. Nhận dạng chữ viết tay được phân ra làm hai loại: *nhận dạng chữ viết tay on-line* và *nhận dạng chữ viết tay off-line*.

## 1.2. CÁC GIAI ĐOẠN CƠ BẢN CỦA MỘT HỆ NHẬN DẠNG CHỮ VIẾT

Một hệ nhận dạng chữ viết bao gồm năm giai đoạn chính sau đây (Hình 1.1).



Hình 1.1. Sơ đồ tổng quát của một hệ thống nhận dạng chữ viết tay.

Luận án chỉ tập trung nghiên cứu ba giai đoạn chính: trích chọn đặc trưng, huấn luyện phân lớp và nhận dạng.

## 1.3. CÁC PHƯƠNG PHÁP TRÍCH CHỌN ĐẶC TRƯNG

### 1.3.1. Biến đổi toàn cục và khai triển chuỗi

Một tín hiệu liên tục thường chứa nhiều thông tin và có thể sử dụng để làm đặc trưng cho mục đích phân lớp. Các đặc trưng này cũng có thể được trích chọn bằng cách xấp xỉ các tín hiệu liên tục thành các tín hiệu rời rạc. Một cách để biểu diễn tín hiệu là sử dụng tổ hợp tuyến tính của một dãy các hàm đơn giản hơn. Các hệ số của tổ hợp tuyến tính cung cấp tri thức giải mã vừa đủ, chẳng hạn như các phép biến đổi hoặc khai triển chuỗi. Một số biến dạng khác như các phép dịch chuyển và phép quay là bất biến dưới các phép biến đổi toàn cục và khai triển chuỗi.

### 1.3.2. Đặc trưng thống kê

Các đặc trưng thống kê của ảnh văn bản bảo toàn các kiểu biến đổi đa dạng về hình dáng của chữ. Mặc dù các kiểu đặc trưng này không thể xây dựng lại ảnh gốc, nhưng nó được sử dụng để thu nhỏ số chiều của tập đặc trưng nhằm tăng tốc độ và giảm thiểu độ phức tạp tính toán.

### 1.3.3. Đặc trưng hình học và hình thái

Các tính chất cục bộ và toàn cục của các ký tự có thể được biểu diễn bằng các đặc trưng hình học và hình thái. Các đặc trưng này có thể giải mã một số tri thức về cấu trúc của đối tượng ảnh hoặc có thể cung cấp các tri thức để phát hiện ra đối tượng.

## 1.4. CÁC PHƯƠNG PHÁP NHẬN DẠNG CHỮ VIẾT TAY

### 1.4.1. Đối sánh mẫu

Kỹ thuật nhận dạng chữ đơn giản nhất dựa trên cơ sở đối sánh các nguyên mẫu (prototype) để nhận dạng ký tự hoặc từ. Nói chung, toán tử đối sánh xác định mức độ giống nhau giữa hai vectơ (nhóm các điểm, hình dạng, độ cong...) trong một không gian đặc trưng.

### 1.4.2. Phương pháp tiếp cận cấu trúc

Cách tiếp cận theo cấu trúc dựa vào việc mô tả đối tượng nhờ một số khái niệm biểu diễn đối tượng cơ sở trong ngôn ngữ tự nhiên. Một số dạng nguyên thủy thường dùng để mô tả đối tượng như đoạn thẳng, cung,... Mỗi đối tượng được mô tả như một sự kết hợp của các dạng nguyên thủy. Các quy tắc kết hợp các dạng nguyên thủy được xây dựng giống như việc nghiên cứu văn phạm trong một ngôn ngữ, do đó quá trình nhận dạng là quá trình phân tích cú pháp. Phương pháp này giải quyết bài toán nhận dạng chữ một cách tổng quát. Tuy nhiên, vẫn còn nhiều vấn đề liên quan đến nhận dạng cú pháp chưa được giải quyết.

### 1.4.3. Các phương pháp thống kê

Hầu hết các kỹ thuật thống kê đều dựa trên cơ sở ba giả thuyết chính sau:

- Phân bố của tập đặc trưng là phân bố Gauss hoặc trong trường hợp xấu nhất là phân bố đều.
- Có các số liệu thống kê đầy đủ có thể dùng cho mỗi lớp.
- Cho tập ảnh  $\{I\}$ , tập ảnh này có thể trích chọn một tập đặc trưng  $\{f_i\} \in F$ ,  $i \in \{1, \dots, n\}$  mà tập đặc trưng này đại diện cho mỗi lớp mẫu riêng biệt.

### 1.4.4. Các phương pháp học máy tiên tiến

#### 1.4.4.1. Mô hình Markov ẩn

HMM áp dụng tốt đối với việc nhận dạng chữ viết tay on-line, đặc biệt là nhận dạng chữ viết tay ở mức từ.

#### 1.4.4.2. Mạng nơ ron

Mạng nơ ron tỏ ra phù hợp với các bài toán đối sánh, phân loại mẫu, xấp xỉ hàm, tối ưu hoá, lượng tử hoá vectơ và phân hoạch không gian dữ liệu. Áp dụng mạng nơ ron vào các hệ thống nhận dạng đạt độ chính xác phân lớp khá cao, có thể so sánh với các hướng tiếp cận phân lớp khác như tiếp cận cấu trúc, thống kê, ...

#### 1.4.4.3. Máy vectơ tựa

Trong những năm gần đây, SVM được biết đến như một hướng tiếp cận phân lớp hiệu quả và đang được áp dụng rộng rãi trong nhiều ứng dụng thực tế. Ưu điểm của SVM là khả năng phân lớp với độ chính xác cao, điều này được đảm bảo bởi các tính chất của siêu phẳng tối ưu và cách sử dụng hàm nhân (kernel). Tuy nhiên, tốc độ phân lớp của SVM bị đánh giá là chậm hơn so với các phương pháp phân lớp khác.

### 1.4.5. Kết hợp các phương pháp nhận dạng

Các phương pháp phân lớp đã được đề cập ở trên đều có thể áp dụng đối với các hệ nhận dạng chữ viết tay. Mỗi kỹ thuật phân lớp đều có những ưu điểm và nhược điểm riêng. Vấn đề đặt ra là các phương pháp trên có thể kết hợp với nhau theo một

cách nào đó để nâng cao hiệu quả nhận dạng hay không? Nhiều công trình nghiên cứu các kiến trúc phân lớp theo ý tưởng kết hợp các kỹ thuật phân lớp đã nêu trên

## 1.5. NGHIÊN CỨU HIỆN THỜI VỀ NHẬN DẠNG CHỮ VIẾT TAY

### 1.5.1. Các nghiên cứu nhận dạng chữ viết tay trên thế giới

Kể từ năm 1999, khi Flatt đề xuất thuật toán SMO để giải bài toán tối ưu trong kỹ thuật phân lớp SVM thì các nhà nghiên cứu tập trung áp dụng phương pháp phân lớp SVM vào các ứng dụng nhận dạng chữ viết tay hoặc kết hợp SVM với các phương pháp truyền thống khác như mạng nơ ron,...

### 1.5.2. Các nghiên cứu về nhận dạng chữ viết tay tiếng Việt

Trong những năm gần đây, vấn đề nhận dạng chữ viết tay đang được nhiều nhà nghiên cứu trong nước đặc biệt quan tâm về cả hai mặt lý thuyết lẫn triển khai ứng dụng. Tuy nhiên các kết quả nghiên cứu lý thuyết chủ yếu chỉ tập trung vào nhận dạng chữ số hoặc chữ cái tiếng Việt không dấu. Chỉ một số ít công trình nghiên cứu đề xuất giải pháp cụ thể cho việc nhận dạng chữ viết tay tiếng Việt.

## 1.6. KẾT LUẬN

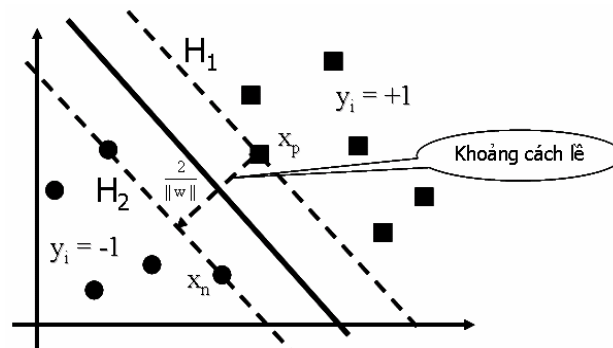
Chương này đã giới thiệu tổng quan về lĩnh vực nhận dạng chữ viết. Trong những năm gần đây, phương pháp phân lớp SVM đã được giới nghiên cứu đặc biệt quan tâm bởi nó được xây dựng trên nền tảng toán học chặt chẽ và đã được áp dụng khá thành công trong một số lĩnh vực nhận dạng cũng như khai phá dữ liệu. Chương tiếp theo của luận án sẽ tập trung nghiên cứu sâu hơn về phương pháp phân lớp SVM trong ứng dụng nhận dạng chữ viết tay rời rạc.

## CHƯƠNG 2: NHẬN DẠNG CHỮ VIẾT TAY RỜI RẠC VỚI PHƯƠNG PHÁP MÁY VECTƠ TỰA

### 2.1. PHƯƠNG PHÁP MÁY VECTƠ TỰA

Phương pháp phân lớp SVM gốc được thiết kế để giải bài toán phân lớp nhị phân. Bài toán phân lớp nhị phân được phát biểu như sau: Cho tập dữ liệu huấn luyện gồm  $l$  mẫu  $\{(x_1, y_1), \dots, (x_l, y_l)\}$  trong đó  $x_i \in \mathbb{R}^D$  và  $y_i \in \{\pm 1\}$ ,  $\forall i \in \{1, \dots, l\}$ . Tìm một hàm tuyến tính  $f(x) = w^T \cdot x + b$  để tách tập dữ liệu huấn luyện thành hai lớp.

#### 2.1.1. SVM tuyến tính



Hình 2.2. Siêu phẳng tách tuyến tính.

Ý tưởng chính của SVM là tìm siêu phẳng phân cách sao cho khoảng cách lề giữa hai lớp mẫu huấn luyện đạt cực đại.

Mục tiêu của SVM là tìm siêu phẳng phân cách với khoảng cách lề cực đại:



$$f^* = \arg \max_f \delta_f \quad (2.4)$$

tức là tìm siêu phẳng  $H: w^T \cdot x + b = 0$  và hai siêu phẳng  $H_1: w^T \cdot x + b = +1$ ,  $H_2: w^T \cdot x + b = -1$  song song với  $H$  sao cho khoảng cách giữa  $H_1$  và  $H_2$  đạt cực đại (Hình 2.2).

Khoảng cách của một điểm nằm trên  $H_1$  hoặc  $H_2$  tới  $H$  là  $\frac{|w^T \cdot x + b|}{\|w\|_2} = \frac{1}{\|w\|_2}$  nên

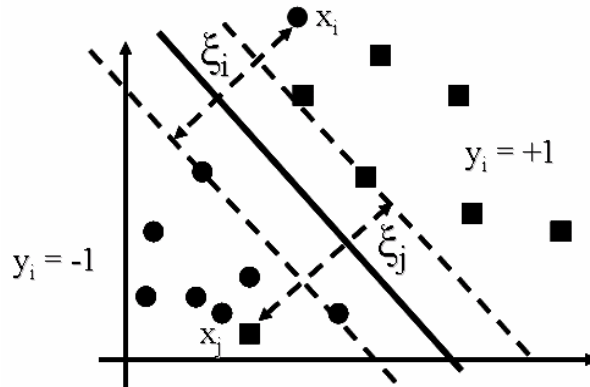
khoảng cách giữa  $H_1$  và  $H_2$  là  $\frac{2}{\|w\|_2}$ . Vì vậy, có thể tìm siêu phẳng với khoảng cách

lớn cực đại bằng cách cực tiểu  $\|w\|_2 = w^T \cdot w$ . Từ đó, bài toán tìm siêu phẳng tối ưu có thể phát biểu lại như sau:

$$\min_{w,b} \frac{1}{2} w^T \cdot w \quad (2.8)$$

$$\text{sao cho } y_i(w^T \cdot x_i + b) \geq 1, i=1, \dots, l \quad (2.9)$$

Kết quả của việc giải bài toán (2.8) sẽ thu được những mẫu nằm trên  $H_1$  và  $H_2$ . Các mẫu này được gọi là các **vector tựa**, chỉ có các mẫu này mới tham gia vào việc xác định siêu phẳng tối ưu, còn các mẫu khác có thể loại bỏ.



Hình 2.4. Phân lớp mềm.

Để nới lỏng điều kiện phân lớp, thêm vào các biến trễ không âm  $\xi_i \geq 0$  (Hình 2.4) sao cho:

$$y_i(w^T \cdot x_i + b) \geq 1 - \xi_i, i=1, \dots, l \quad (2.10)$$

$$\xi_i \geq 0, \forall i \quad (2.11)$$

Bài toán tìm siêu phẳng tối ưu được phát biểu lại như sau:

$$\min_{w,b} \frac{1}{2} w^T \cdot w + C \sum_{i=1}^l \xi_i \quad (2.15)$$

$$\text{sao cho } y_i(w^T \cdot x_i + b) \geq 1 - \xi_i, i=1, \dots, l \quad (2.16)$$

$$\xi_i \geq 0, i=1, \dots, l \quad (2.17)$$

Phương pháp giải bài toán tối ưu lồi có ràng buộc được trình bày tóm tắt như sau:

**Định nghĩa 2.4 [9]:** Cho bài toán tối ưu với tập lồi  $\Omega \subseteq R^D$

$$\min f(w), w \in \Omega \quad (2.18)$$

$$\text{sao cho } g_i(w) \leq 0, i=1, \dots, k \quad (2.19)$$

$$h_i(w) = 0, i=1, \dots, m \quad (2.20)$$

Hàm Lagrange tổng quát gắn với bài toán trên là hàm có dạng

$$L(w, \alpha, \beta) = f(w) - \sum_{i=1}^k \alpha_i g_i(w) + \sum_{i=1}^m \beta_i h_i(w) \quad (2.21)$$

**Định nghĩa 2.5 [9]:** Bài toán Lagrange đối ngẫu của bài toán gốc (2.18) là bài toán có dạng

$$\max \quad \theta(\alpha, \beta) \quad (2.22)$$

$$\text{sao cho } \alpha \geq 0 \quad (2.23)$$

trong đó  $\theta(\alpha, \beta) = \inf_{w \in R^D} L(w, \alpha, \beta)$ .

Việc giải bài toán tối ưu (2.15) sẽ tương đương với việc giải bài toán đối ngẫu sau đây:

$$\max_{\alpha} \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j x_i \cdot x_j \quad (2.35)$$

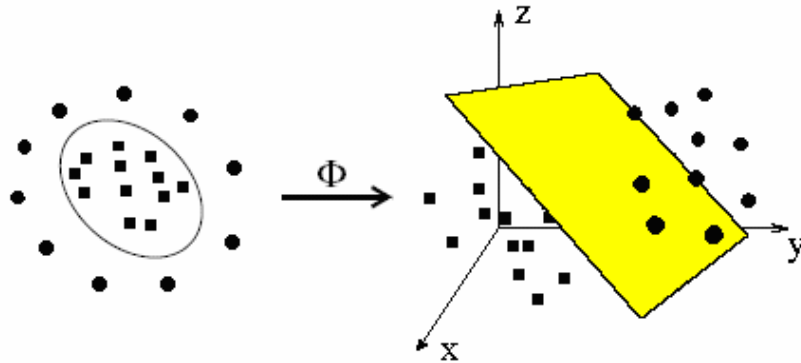
$$\text{sao cho } \sum_{i=1}^l y_i \alpha_i = 0 \quad (2.36)$$

$$0 \leq \alpha_i \leq C, \quad i = 1, \dots, l \quad (2.37)$$

### 2.1.2. SVM phi tuyến

Với tập dữ liệu không khả tách tuyến tính, có thể ánh xạ chúng sang một không gian khác với số chiều cao hơn sao cho trong không gian mới này tập dữ liệu sẽ khả tách tuyến tính (Hình 2.5). Như vậy, hàm mục tiêu  $L_D$  có thể viết lại:

$$L_D = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j \Phi(x_i) \cdot \Phi(x_j) \quad (2.42)$$



Hình 2.5. Ánh xạ dữ liệu vào không gian đặc trưng.

**Định nghĩa 2.6 [10]:** Cho ánh xạ  $\Phi$  từ không gian  $X$  vào không gian đặc trưng  $F$ .  $\forall x, y \in X$ , một hàm nhân  $K$  có dạng

$$K(x, y) = \Phi(x) \cdot \Phi(y) \quad (2.43)$$

**Định lý 2.2 [10]:** (Định lý Mercer) Để đảm bảo cho một hàm liên tục đối xứng  $K(x, y)$  trong không gian  $L_2(C)$  có một khai triển

$$K(x, y) = \sum_{k=1}^{\infty} a_k z_k(x) z_k(y) \quad (2.44)$$

với các hệ số  $a_k > 0$  (tức  $K(x, y)$  là tích vô hướng trong không gian đặc trưng), thì điều kiện cần và đủ là

$$\iint_{C \times C} K(x, y) g(x) g(y) dx dy \geq 0 \quad (2.45)$$

với mọi  $g \in L_2(C)$  ( $C$  là tập compact của  $R^D$ ).

Tóm lại, bài toán tìm siêu phẳng tối ưu trong không gian đặc trưng sẽ được giải thông qua bài toán QP sau:

$$\max_{\alpha} \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j K(x_i, x_j) \quad (2.46)$$

$$\text{sao cho } \sum_{i=1}^l y_i \alpha_i = 0 \quad (2.47)$$

$$0 \leq \alpha_i \leq C, i = 1, \dots, l \quad (2.48)$$

Và hàm quyết định phân lớp cuối cùng có dạng:

$$f(x) = \text{sign} \left( \sum_{\alpha_i \neq 0} y_i \alpha_i K(x_i, x) + b \right) \quad (2.49)$$

### 2.1.3. Các thuật toán huấn luyện SVM

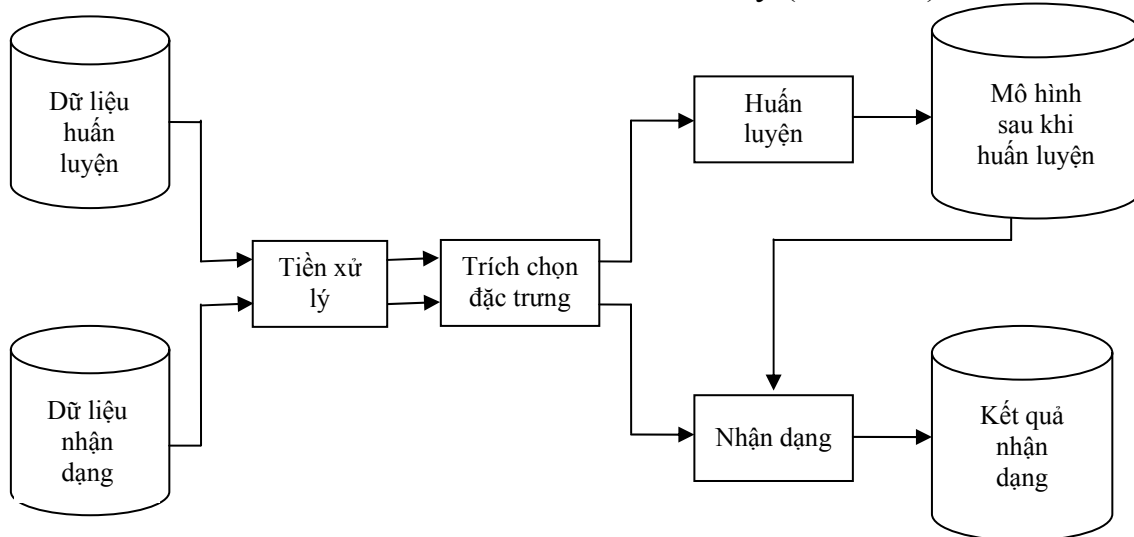
Việc huấn luyện phân lớp SVM nhị phân đã được Vapnik đề xuất thông qua thuật toán chặt khúc, thuật toán này vẫn còn nhiều nhược điểm và đã được Osuna khắc phục trong thuật toán phân rã [13] và Flatt tiếp tục cải tiến, đề xuất thuật toán SMO [12] để giải bài toán QP mà không cần lưu trữ ma trận Gram.

### 2.1.4. SVM đa lớp

Một số chiến lược thường dùng để áp dụng cho bài toán SVM đa lớp là: Chiến lược một đối một (OVO), chiến lược một đối phần còn lại (OVR) và chiến lược phân cấp.

## 2.2. MÔ HÌNH NHẬN DẠNG CHỮ VIẾT TAY RỜI RẠC

Mô hình được thực hiện theo hai bước chính sau đây (Hình 2.6):



Hình 2.6. Mô hình nhận dạng chữ viết tay rời rạc.

**Bước 1:** Xây dựng mô hình huấn luyện.

**Bước 2:** Nhận dạng.

Dựa vào giá trị các tham số của hàm quyết định thu được ở bước 1, một mẫu mới x sau khi đã qua các khâu tiền xử lý và trích chọn đặc trưng sẽ được đưa vào tính toán thông qua hàm quyết định để xác định lớp của mẫu x.

### 2.2.1. Tiền xử lý

Sau khi đã khử nhiễu, ảnh được chuẩn hóa về kích thước chuẩn  $16 \times 16$ , đây là kích thước vừa đủ để biểu diễn ảnh ký tự. Chuẩn hóa kích thước ảnh được thực hiện theo các bước sau:

Bước 1: Nhị phân hóa ảnh.

Bước 2: Tìm hình chữ nhật R bé nhất chứa các điểm đen trên ảnh.

Bước 3: Lấy vùng ảnh I nằm trong hình chữ nhật R.

Bước 4: Chuẩn hóa ảnh I về kích thước chuẩn  $16 \times 16$ .

### 2.2.2. Trích chọn đặc trưng

Sau khi chuẩn hóa về kích thước chuẩn  $16 \times 16$ , giữ nguyên ma trận ảnh để làm đặc trưng cho việc huấn luyện phân lớp và nhận dạng. Như vậy, đối với ảnh kích thước  $16 \times 16$ , đặc trưng được trích chọn đại diện cho mỗi ký tự là vectơ 256 chiều.

### 2.2.3. Lựa chọn thuật toán huấn luyện phân lớp

Trong phần cài đặt thực nghiệm, tác giả áp dụng thuật toán SMO để huấn luyện phân lớp SVM nhị phân, sử dụng và kế thừa một số chức năng của phần mềm mã nguồn mở LibSVM để phát triển ứng dụng nhận dạng chữ viết tay rời rạc.

### 2.2.4. Thuật toán nhận dạng chữ viết tay rời rạc

Cả hai chiến lược phân lớp OVO và OVR đều có thể áp dụng để phân lớp dữ liệu một cách tổng quát mà không cần phải can thiệp sâu để phân tích các đặc trưng khác nhau giữa các lớp dữ liệu. Vì vậy trong Chương này, luận án sẽ lựa chọn hai chiến lược phân lớp này để cài đặt thử nghiệm thuật toán nhận dạng đối với dữ liệu chữ viết tay rời rạc.

---

#### Thuật toán 2.1: SVMClassify

*//Thuật toán nhận dạng theo các chiến lược phân lớp SVM*

#### Input:

- Mẫu  $x$ ;
- Số lớp  $N$ ;
- Chiến lược phân lớp Strategy;
- Các mô hình đã huấn luyện {OVOModel, OVRModel}

#### Output:

label; *// Nhãn lớp của mẫu  $x$*

#### Method

##### Case Strategy of

**OVO:** *// Chiến lược một đối một*

Khởi tạo Count[i] = 0; *//  $i=0, \dots, N-1$*

LoadModel(OVOModel);

**for** (i=0; i < N-1; i++)

**for** (j=i+1; j < N; j++)

        Count[BinarySVM(x, i, j)]++;

Count[label]=Max(Count[i]);

**OVR:** *// Chiến lược một đối phần còn lại*

LoadModel(OVRModel);

label=-1;

**for** (i=0; i < N; i++)

    { label=BinarySVM(x, i, Rest);

        if(label=i) break; }

**EndCase;**

**Return** label;

Trong đó,  $BinarySVM(x,i,j)$  là hàm xếp  $x$  vào một trong hai lớp  $i$  hoặc  $j$ .

### 2.3. KẾT QUẢ THỰC NGHIỆM

Các kết quả thực nghiệm được cài đặt và chạy thử nghiệm trên môi trường Window XP, máy PC Pentium 4 tốc độ 2.4 Ghz với dung lượng bộ nhớ RAM 1Gb.

#### 2.3.1. Chuẩn bị các bộ dữ liệu thực nghiệm

##### 2.3.1.1. Các bộ dữ liệu chuẩn USPS và MNIST

Bộ dữ liệu USPS bao gồm 7291 mẫu dùng để huấn luyện và 2007 mẫu khác để nhận dạng, mỗi mẫu là một ảnh đa cấp xám kích thước  $16 \times 16$ . Bộ dữ liệu MNIST bao gồm 60.000 mẫu huấn luyện và 10.000 mẫu khác để nhận dạng, mỗi mẫu là một ảnh kích thước  $28 \times 28$ .

##### 2.3.1.2. Dữ liệu chữ viết tay tiếng Việt

Tác giả xây dựng bộ dữ liệu chữ viết tay tiếng Việt (**VietData**) phục vụ cho việc thực nghiệm bao gồm 89 lớp chữ cái in hoa, mỗi lớp chọn ra 200 mẫu, như vậy bộ dữ liệu VietData có tổng cộng 17800 mẫu.

#### 2.3.2. Kết quả thực nghiệm trên các bộ dữ liệu chuẩn USPS và MNIST

Đầu tiên luận án thử nghiệm hiệu quả Thuật toán 2.1 trên các bộ dữ liệu chuẩn USPS và MNIST với các chiến lược OVO và OVR. Mô hình SVM được sử dụng với hàm nhân Gauss và các tham số  $C = 10$  (tham số hàm phạt), Cache = 1000 (kích thước vùng nhớ để lưu trữ các vector tựa).

*Bảng 2.1. Kết quả thực nghiệm trên tập USPS với hàm nhân RBF ( $\sigma = 0.05$ ).*

Chiến lược	Số vector tựa	Thời gian huấn luyện	Thời gian Test	Độ chính xác
OVR	7874	195 giây	15 giây	95,3%
OVO	13952	82 giây	25 giây	95,3%

*Bảng 2.2. Kết quả thực nghiệm trên tập MNIST với hàm nhân RBF ( $\sigma = 0.08$ ).*

Chiến lược	Số vector tựa	Thời gian huấn luyện	Thời gian Test	Độ chính xác
OVR	8542	> 9 giờ	~ 3 phút	96,1%
OVO	31280	~ 2 giờ	~ 5 phút	97,2%

Các kết quả thực nghiệm ở Bảng 2.1 và 2.2 cho thấy các chiến lược OVO và OVR đều có các ưu điểm và nhược điểm riêng.

Tác giả tiếp tục thử nghiệm huấn luyện và nhận dạng theo mô hình phân lớp SVM với các hàm nhân và tham số khác nhau. Bảng 2.3 là kết quả thực nghiệm trên tập dữ liệu USPS sử dụng các hàm nhân khác nhau với các tham số  $C=10$ , Cache=1000.

Kết quả thực nghiệm ở Bảng 2.3 cho thấy sử dụng các hàm nhân nào có độ biến thiên đa dạng hơn sẽ cho độ chính xác phân lớp cao hơn.

*Bảng 2.3. Kết quả thực nghiệm với các hàm nhân khác nhau trên tập USPS.*

Hàm Kernel	Huấn luyện		Test	
	Thời gian	Số mẫu phân lớp sai	Thời gian	Độ chính xác
Tuyến tính	9 phút	322/7291	5 giây	90.0%
Đa thức bậc 2	2 phút	424/7291	20 giây	92.7%
RBF ( $\sigma=0.05$ )	3 phút	1/7291	50 giây	95.3%

Phần tiếp theo sẽ thử nghiệm huấn luyện mô hình SVM trên tập dữ liệu USPS với các tham số  $C=10$ ,  $Cache=1000$ , hàm nhân Gauss với các giá trị  $\sigma$  khác nhau (Bảng 2.4).

*Bảng 2.4. Kết quả huấn luyện với hàm nhân Gauss.*

$\sigma$	Số mẫu học	Số vectơ tựa	Số mẫu phân lớp sai
0.5	7291	43767	3788/7291
0.05	7291	12454	1/7291
0.01	7291	4423	9/7291

Kết quả ở Bảng 2.4 cho thấy chất lượng của mô hình huấn luyện SVM phụ thuộc vào việc lựa chọn giá trị của tham số  $\sigma$ .

Tiếp tục thử nghiệm huấn luyện mô hình SVM với các kích thước Cache khác nhau với  $C=10$ , hàm nhân RBF ( $\sigma=0.05$ ) trên tập dữ liệu USPS (Bảng 2.5).

*Bảng 2.5. Kết quả huấn luyện với kích thước cache khác nhau.*

Cache	Số mẫu học	Số vectơ tựa	Số mẫu phân lớp sai
1000	7291	12454	1/7291
300	7291	11151	48/7291
100	7291	4486	590/7291

Bảng 2.5 cho thấy chất lượng của thuật toán huấn luyện SVM cũng phụ thuộc vào không gian nhớ của máy tính.

Cuối cùng, tác giả so sánh các kết quả nhận dạng theo phương pháp phân lớp SVM so với các phương pháp sử dụng mô hình mạng nơ ron 4 lớp (144 nơ ron lớp vào, 72+36 nơ ron ở các lớp ẩn, 10 nơ ron lớp ra) [15] trên cùng một bộ dữ liệu chuẩn MNIST (Bảng 2.6).

*Bảng 2.6. So sánh kết quả nhận dạng của SVM với mô hình mạng nơ ron.*

Các thông số	Mạng nơ ron	SVM
Số mẫu học	60.000	60.000
Thời gian học	~ 24 giờ	~ 2 giờ
Số mẫu test	10.000	10.000
Thời gian test	~ 2 phút	~ 5 phút
Tỷ lệ test lỗi (%)	4.6	<b>2.8</b>

Kết quả ở Bảng 2.6 cho thấy kết quả nhận dạng theo mô hình SVM có độ chính xác cao hơn so với mô hình mạng nơ ron, tuy nhiên tốc độ nhận dạng của SVM lại chậm hơn.

### 2.3.3. Kết quả thực nghiệm trên dữ liệu chữ viết tay tiếng Việt

Việc thực nghiệm trên dữ liệu chữ viết tay tiếng Việt được tiến hành theo phương thức thẩm định chéo (Cross-Validation). Bộ dữ liệu **VietData** được chia thành  $k$  phần (ở đây  $k$  được chọn =10), sau đó sử dụng  $k-1$  phần để huấn luyện và 1 phần còn lại để nhận dạng, quá trình được lặp đi lặp lại  $k$  lần. Các kết quả thực nghiệm được thể hiện trên Bảng 2.7.

Kết quả thực nghiệm ở Bảng 2.7 cho thấy tốc độ phân lớp của SVM đối với bài toán phân đa lớp là quá chậm, không thể đáp ứng được đối với một hệ thống nhận dạng thời gian thực. Vì vậy, cần phải có những giải pháp phù hợp để tăng tốc độ cũng như độ chính xác phân lớp đối với dữ liệu chữ viết tay tiếng Việt.

Bảng 2.7. Thực nghiệm trên tập dữ liệu chữ viết tay tiếng Việt.

Chiến lược	Thời gian huấn luyện	Thời gian Test	Độ chính xác
OVR	~ 49 phút	~ 2 phút	82.7%
OVO	~ 16 phút	~ 6 phút	83.6%

## 2.4. KẾT LUẬN

Chương này đã nghiên cứu triển khai ứng dụng, xây dựng thuật toán nhận dạng chữ viết tay rời rạc trên cơ sở phân lớp SVM với hai chiến lược OVO và OVR. Các kết quả nghiên cứu của luận án liên quan đến Chương này đã được tác giả công bố ở các công trình [1,2,3].

# CHƯƠNG 3: NÂNG CAO HIỆU QUẢ NHẬN DẠNG CHỮ VIẾT VIẾT TAY RỜI RẠC

## 3.1. ĐẶT VẤN ĐỀ

Khi sử dụng SVM để giải bài toán này sẽ vấp phải một số vấn đề sau:

- Để đảm bảo độ chính xác cho hệ thống nhận dạng, mỗi lớp phải có một số lượng mẫu đủ lớn để huấn luyện.
- Bùng nổ số lớp (99 lớp). Nếu sử dụng chiến lược OVO cho bài toán phân lớp thì hệ thống phải xây dựng 4851 máy phân lớp SVM nhị phân. Điều này sẽ làm cho hệ thống huấn luyện và nhận dạng rất chậm, khó có thể đáp ứng được cho các ứng dụng thời gian thực, thậm chí không đủ không gian nhớ để huấn luyện do số vectơ tựa sinh ra lớn hơn nhiều lần so với tổng số lượng mẫu tham gia huấn luyện.
- Bộ ký tự tiếng Việt có nhiều ký tự với hình dáng rất giống nhau, chỉ khác nhau chút ít về phân dấu. Do đó rất khó để phân biệt đối với chữ viết tay, điều này dẫn đến độ chính xác của hệ thống nhận dạng không cao.

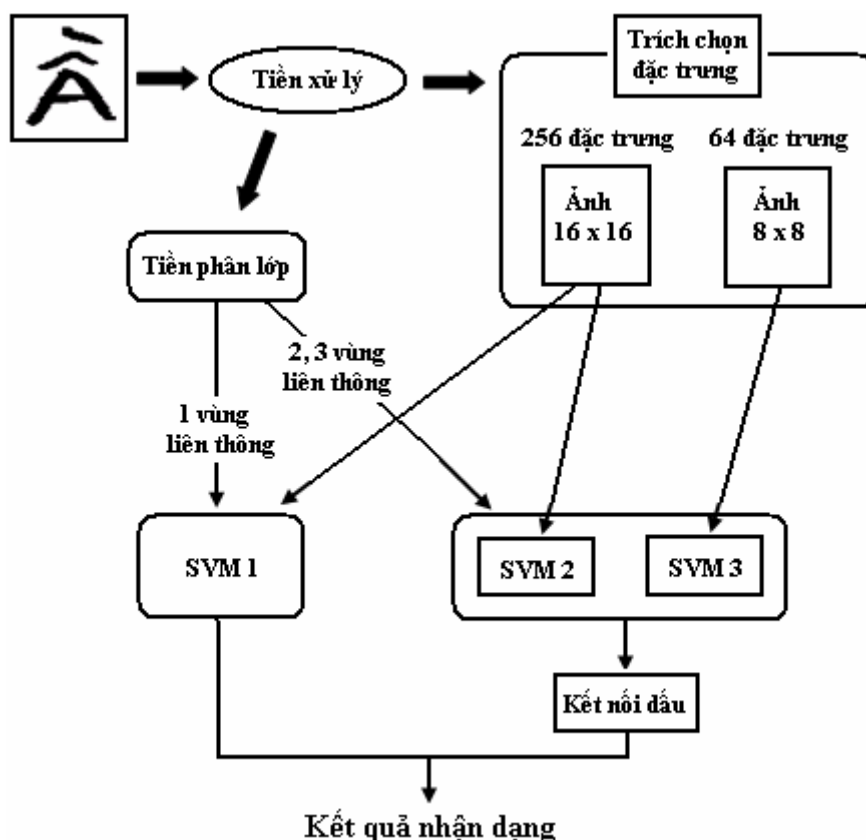
## 3.2. GIẢI PHÁP NHẬN DẠNG CHỮ VIẾT VIẾT TAY RỜI RẠC

Phần này sẽ trình bày chi tiết giải pháp nhận dạng chữ Việt viết tay rời rạc. Trên cơ sở các thành phần liên thông của ảnh, phân tập ký tự tiếng Việt thành ba nhóm và tách các ký tự có dấu thành các phần rời nhau. Sau đó xây dựng các máy phân lớp SVM để nhận dạng cho từng phần chữ và phần dấu, cuối cùng ghép nối các kết quả nhận dạng của các phần chữ và dấu để có kết quả nhận dạng cuối cùng (Hình 3.1).

### 3.2.1. Tiền xử lý

Ảnh quét vào thường hay bị nhiễu, các loại nhiễu phổ biến là nhiễu đốm và nhiễu vệt. Để khử nhiễu đốm, sử dụng các bộ lọc trung bình và lọc trung vị là hiệu quả nhất, còn đối với các nhiễu vệt thì sử dụng phương pháp khử các vùng liên thông nhỏ tỏ ra hữu hiệu hơn.

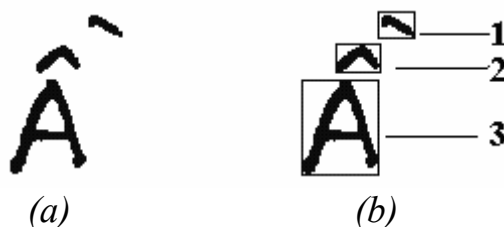
Để thuận tiện cho việc xử lý sau này, ảnh đầu vào được biến đổi từ ảnh đa cấp xám thành ảnh nhị phân.



Hình 3.1. Kiến trúc của hệ nhận dạng chữ viết tay tiếng Việt

### Chuẩn hóa ảnh theo vùng liên thông

Chuẩn hóa ảnh nhằm mục đích tạo điều kiện thuận tiện cho công đoạn tách ảnh thành từng phần chữ và dấu.



Hình 3.3. Chuẩn hóa ảnh: (a) Ảnh gốc, (b) Xác định các vùng liên thông và đánh số tự các vùng liên thông.

**Bước 1:** Xác định các vùng liên thông trên ảnh (Hình 3.3).

**Bước 2:** Sắp xếp các vùng liên thông theo thứ tự từ trên xuống (Hình 3.3b).

**Bước 3:**

- Nếu ảnh chỉ có 1 vùng liên thông: Chuẩn hóa ảnh về kích thước  $16 \times 16$  (Hình 3.4a).

- Nếu ảnh có 2 vùng liên thông: Gọi  $S(i)$  là diện tích vùng liên thông thứ  $i$ .

Nếu  $S(1) > S(2)$  thì dấu của phần liên thông 2 là dấu nặng (.) và chỉ cần chuẩn hóa vùng liên thông 1 về kích thước  $16 \times 16$ .

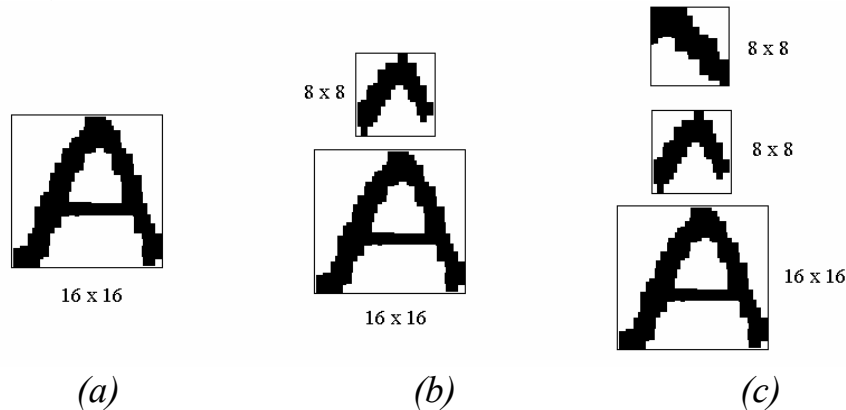
Ngược lại: Tách ảnh thành 2 phần: phần chữ và phần dấu. Chuẩn hóa phần chữ về kích thước  $16 \times 16$  và phần dấu về kích thước  $8 \times 8$  (Hình 3.4b).

- Nếu ảnh có 3 vùng liên thông:



Nếu  $S(3) = \text{Min}\{S(i)\}$  thì dấu của phần liên thông này là dấu nặng (.). Do đó chỉ cần chuẩn hóa thành phần liên thông 1 về kích thước  $8 \times 8$  và thành phần liên thông 2 về kích thước  $16 \times 16$ .

Ngược lại: Tách ảnh thành 3 phần từ các vùng liên thông. Chuẩn hóa các vùng liên thông 1 và 2 về kích thước  $8 \times 8$  và chuẩn hóa vùng liên thông 3 về kích thước  $16 \times 16$  (Hình 3.4c).



Hình 3.4. Chuẩn hóa kích thước các vùng liên thông.

### 3.2.2. Phân nhóm sơ bộ

Dựa vào số thành phần liên thông để tách bộ ký tự tiếng Việt thành 3 nhóm:

- Nhóm 1: Nhóm có 1 vùng liên thông {A, B, C, D, Đ, E, G, H, I, K, L, M, N, O, P, Q, R, S, T, U, V, X, Y, Ơ, U}.
- Nhóm 2: Nhóm có 2 vùng liên thông {Ă, Â, Æ, Å, Æ, Á, À, Ê, È, Ẽ, É, Ẻ, Ì, Î, Í, Ì, Ô, Ò, Ỏ, Õ, Ó, Ơ, Ờ, Ỡ, Ớ, Ợ, Ù, Ủ, Û, Ú, Ụ, Ừ, Ứ, Ý, Ỡ, Ỡ, Ỡ, Ỡ}.
- Nhóm 3: Nhóm có 3 vùng liên thông { Æ, Æ, Æ, Æ, Æ, Æ, Æ, Æ, Æ, Æ, Ê, Ê, Ê, Ê, Ê, Ờ, Ờ, Ỡ, Ỡ }

Việc phân nhóm sơ bộ này sẽ khắc phục được sự bùng nổ số phân lớp nhị phân, giúp cho việc giải bài toán nhận dạng chữ Việt viết tay rời rạc có thể thực hiện được trên các máy tính cá nhân cũng như tăng tốc độ nhận dạng.

### 3.2.3. Huấn luyện các máy phân lớp SVM

Với việc phân nhóm sơ bộ như trên, chỉ cần xây dựng và huấn luyện ba 3 máy phân lớp SVM. Các máy phân lớp SVM nhị phân sử dụng hàm nhân Gauss với tham số  $\sigma = 0.01$  để huấn luyện phân lớp và nhận dạng.

SVM 1: Máy phân lớp đối với nhóm ký tự có 1 vùng liên thông {A, B, C, D, Đ, E, G, H, I, K, L, M, N, O, P, Q, R, S, T, U, V, X, Y, Ơ, U}.

SVM 2: đối với các ký tự có dấu thì phần chữ đều là các nguyên âm, vì vậy máy này chỉ phân lớp các nguyên âm {A, E, I, O, U, Y}.

SVM 3: phân lớp các dấu {/, \, ?, ~, ^, v} (sắc, huyền, hỏi, ngã, dấu ô, dấu ă).

Đối với SVM 1, do có nhiều ký tự có hình dạng giống nhau nên phải sử dụng chiến lược OVO để tăng độ chính xác phân lớp, chấp nhận hy sinh về mặt tốc độ. Còn đối với SVM 2 và SVM 3, sự khác biệt nhau giữa các lớp tương đối lớn nên chỉ cần sử dụng chiến lược OVR cũng đảm bảo được độ chính xác phân lớp và đồng thời tăng tốc độ nhận dạng.

### 3.2.4. Thuật toán nhận dạng chữ Việt viết tay rời rạc

Thuật toán sau đây sẽ kế thừa Thuật toán 2.1 để mở rộng và phát triển thuật toán nhận dạng chữ Việt viết tay rời rạc. Trong đó  $SVMClassify(x, N, Model)$  là hàm xếp  $x$  vào một trong  $N$  lớp với mô hình đã huấn luyện là  $Model$  theo Thuật toán 2.1.

#### Thuật toán 3.1:

```
// Thuật toán nhận dạng chữ Việt viết tay rời rạc
Input: - Mẫu  $x$ ;
        - Các mô hình đã huấn luyện {SVM1, SVM2, SVM3};
Output: label; // Nhãn lớp của mẫu  $x$ 
Method
k = RegionNum(x);
Case k of
1: label = SVMClassify(x, 25, SVM1);
2: x1 = ConnectedRegion(1);
   x2 = ConnectedRegion(2);
   if (size(x1) > size(x2))
   { Marklabel = LabelNum(".");
     Charlabel = SVMClassify(x1, 6, SVM2); }
   else
   { Marklabel = SVMClassify(x1, 6, SVM3);
     Charlabel = SVMClassify(x2, 6, SVM2); }
   label = Combine(Charlabel, Marklabel);
3: x1 = ConnectedRegion(1);
   x2 = ConnectedRegion(2);
   x3 = ConnectedRegion(3);
   if (size(x2) > size(x3))
   { Marklabel1 = LabelNum(".");
     Marklabel2 = SVMClassify(x1, 6, SVM3);
     Charlabel = SVMClassify(x2, 6, SVM2); }
   else
   { Marklabel1 = SVMClassify(x1, 6, SVM3);
     Marklabel2 = SVMClassify(x2, 6, SVM3);
     Charlabel = SVMClassify(x3, 6, SVM2); }
   label = Combine(Charlabel, Marklabel1, Marklabel2);
EndCase;
Return label;
```

### 3.2.5. Kết quả thực nghiệm

Tác giả tiếp tục xây dựng 3 tập dữ liệu phục vụ cho việc huấn luyện 3 máy phân lớp SVM, mỗi lớp sử dụng 180 mẫu:

**TrainData1:** Tập các dấu tiếng Việt {/, \, ?, ~, ^, v}, với 1080 mẫu.

**TrainData2:** Tập các nguyên âm tiếng Việt {A, E, I, O, U, Y}, với 1080 mẫu.

**TrainData3:** Tập các chữ cái tiếng Việt không dấu {A, B, C, D, Đ, E, G, H, I, K, L, M, N, O, P, Q, R, S, T, U, V, X, Y, Ơ, Ư}, với 4500 mẫu.

Bảng 3.1. So sánh kết quả nhận dạng của hai thuật toán trên tập dữ liệu VietData.

Thuật toán	Thời gian Test	Độ chính xác
Thuật toán 2.1	~ 2 phút	82.7%
Thuật toán 3.1	~ 1 phút	91.4%

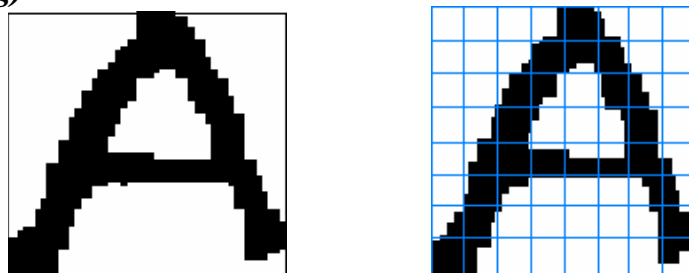
Sau khi đã huấn luyện các máy phân lớp, tác giả tiến hành thực nghiệm so sánh hiệu quả nhận dạng của hai thuật toán trên cùng tập dữ liệu VietData được xây dựng ở Chương 2. Bộ dữ liệu VietData được chia thành mười phần, Bảng 3.1 là kết quả trung bình của 10 lần test đối với Thuật toán 3.1 và thẩm định chéo đối với Thuật toán 2.1.

Kết quả so sánh ở Bảng 3.1 cho thấy hiệu quả nhận dạng trên dữ liệu chữ viết tay tiếng Việt của Thuật toán 3.1 tốt hơn Thuật toán 2.1 trên cả hai mặt: tốc độ và độ chính xác phân lớp.

### 3.3. CẢI TIẾN ĐỘ CHÍNH XÁC NHẬN DẠNG

#### 3.3.1. Kết hợp các đặc trưng thống kê

##### *Phân vùng (Zoning)*



Hình 3.5. Trích chọn đặc trưng trọng số vùng.

Với ảnh kích thước  $16 \times 16$ , chọn  $N=8$ , như vậy có  $8 \times 8 = 64$  đặc trưng.

##### *Biểu đồ chiếu (Projection histograms)*



Hình 3.6. Trích chọn các biểu đồ chiếu ngang, dọc và 2 đường chéo.

Với ảnh kích thước  $16 \times 16$ , chọn  $16$  ngang +  $16$  dọc +  $2 \times 31$  chéo =  $94$  đặc trưng.

##### *Chu tuyến (Contour Profile)*



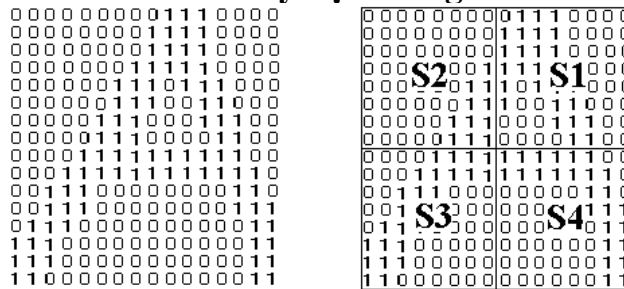
Hình 3.7. Trích chọn các khối bên ngoài của chữ.

Với ảnh kích thước  $16 \times 16$ , có  $16$  trái +  $16$  phải +  $16$  trên +  $16$  dưới =  $64$  đặc trưng.

**Lai ghép các đặc trưng**

Trong phần thực nghiệm, tác giả thử nghiệm kết hợp ba đặc trưng trên (ZPC - Zones, Projection histograms và Contour profiles) để nâng cao độ chính xác của hệ thống nhận dạng.

**3.3.2. Biến đổi ảnh hai chiều thành dãy đặc trưng**



Hình 3.8. Quá trình trích chọn đặc trưng

Từ ảnh nhị phân kích thước  $2^n \times 2^n$  (Hình 3.8), quá trình biến đổi ảnh hai chiều thành dãy đặc trưng được thực hiện theo thuật toán sau:

**Thuật toán 3.2: TransformFeature**

**Input** Ma trận vuông  $(A, n)$  cấp  $2^n$ ;

**Output** Tập các đặc trưng  $\{F_1, F_2, \dots, F_{2^n \times 2^n}\}$ ;

**Method**

```

1. Khởi tạo: Queue = ∅;
               i = 1;
2.  $F_i$  = Tổng các điểm đen trong toàn bộ ma trận  $(A, n)$ ;
   PUSH( $(A, n)$ , Queue);
3. while Queue  $\neq \emptyset$  do
   { POP(Queue,  $(A, n)$ );
     if ( $n > 1$ )
     {  $n = n \text{ DIV } 2$ ;
       Chia ảnh thành 4 phần:  $A_1, A_2, A_3, A_4$ ;
       for ( $j=1; i \leq 4; j++$ )
         PUSH( $(A_j, n)$ , Queue);
     }
   // Gọi S, S1, S2, S3, S4 là tổng các điểm đen tương ứng với
   // các khối A, A1, A2, A3, A4;
    $F_{i+1} = S1 + S2$ ;
    $F_{i+2} = S2 + S3$ ;
    $F_{i+3} = S4$ ;
    $i = i + 3$ ;
   }

```

**Return;**

**Mệnh đề** (tính duy nhất của đặc trưng theo phép biến đổi TransformFeature): Cho ma trận vuông A cấp  $2^n$ , n nguyên dương. Theo thuật toán TransformFeature thì dãy đặc trưng thu được là duy nhất đối với ma trận A.

**Chứng minh:**

Dùng phương pháp quy nạp.

Ta chứng minh mệnh đề đúng với  $n=1$ . Thật vậy, giả sử  $x_1, x_2, x_3, x_4$  là bốn phần tử của ma trận vuông cấp 2. Theo phương pháp biến đổi ảnh thành dãy đặc trưng trên, ta có hệ phương trình:

$$\begin{cases} x_1 + x_2 + x_3 + x_4 = S \\ x_1 + x_2 = S_1 + S_2 \\ x_2 + x_3 = S_2 + S_3 \\ x_4 = S_4 \end{cases} \text{ và } \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = 1 \neq 0$$
 do đó hệ phương trình có nghiệm duy

nhất. Vậy với  $n=1$ , theo thuật toán TransformFeature thì dãy đặc trưng thu được là duy nhất đối với ma trận A.

Giả sử mệnh đề đúng với  $n=k$ . Ta sẽ chứng minh mệnh đề đúng với  $n=k+1$ .

Rõ ràng ma trận vuông cấp  $2^{k+1}$  có kích thước gấp 4 lần ma trận vuông cấp  $2^k$ . Ta sẽ chứng minh rằng nếu mỗi một phần tử của ma trận vuông A cấp  $2^{k+1}$  có dãy đặc trưng thu được là duy nhất thì ma trận vuông A cũng có dãy đặc trưng thu được là duy nhất theo thuật toán TransformFeature.

Thật vậy, giả sử ma trận vuông A cấp  $2^{k+1}$  được chia thành 4 khối con  $A_1, A_2, A_3, A_4$  kích thước  $2^k$  có tổng các điểm đen tương ứng là  $S_1, S_2, S_3, S_4$ . Với cách chia thành 4 khối như vậy thì ma trận A sẽ có nghiệm duy nhất  $X_1=S_1, X_2=S_2, X_3=S_3, X_4=S_4$  tương ứng với các phần tử  $A_1, A_2, A_3, A_4$ . Mà mỗi khối  $A_i, i=1..4$  có duy nhất dãy đặc trưng theo thuật toán TransformFeature nên dãy đặc trưng của ma trận A cũng duy nhất theo thuật toán TransformFeature  $\square$ .

Trong thực nghiệm, với phần chữ kích thước  $16 \times 16$ , như vậy có:  $1 + 3 + 4 \times 3 + 4 \times 4 \times 3 + 4 \times 4 \times 4 \times 3 = 256$  đặc trưng, còn với phần dấu kích thước  $8 \times 8$ , như vậy có tất cả 64 đặc trưng.

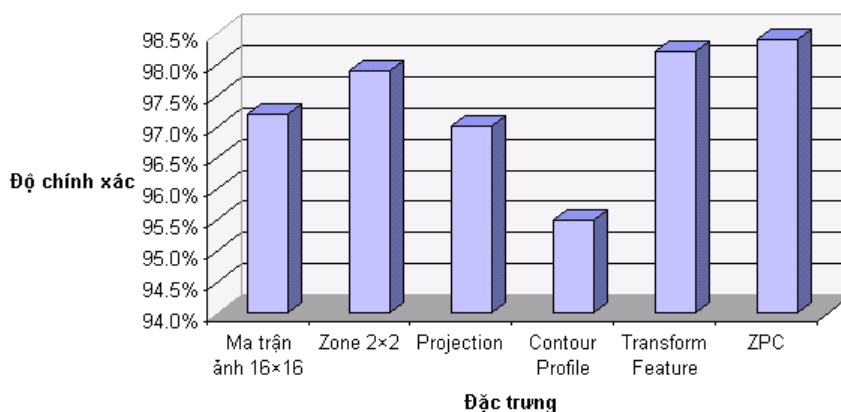
### 3.3.3. Kết quả thực nghiệm trên tập dữ liệu chuẩn MNIST

Phần này sẽ cài đặt thử nghiệm hiệu quả của các loại đặc trưng khác nhau trên tập dữ liệu MNIST theo Thuật toán 2.1 với chiến lược OVO, chọn hàm nhân RBF( $\sigma = 0.05$ ) (xem Bảng 3.2).

Bảng 3.2. Kết quả nhận dạng trên tập dữ liệu MNIST với các loại đặc trưng.

Đặc trưng	Số đặc trưng	Độ chính xác
Ma trận ảnh $16 \times 16$	256	97.2%
Zone $2 \times 2$	64	97.9%
Projection	94	97.0%
Contour Profile	64	95.5%
TransformFeature	256	98.2%
ZPC	222	98.4%

Kết quả ở Bảng 3.2 và Hình 3.10 cho thấy việc sử dụng tập đặc trưng lai ghép ZPC và đặc trưng TransformFeature đạt độ chính xác phân lớp cao hơn so với các đặc trưng khác.



Hình 3.10. So sánh hiệu quả phân lớp của các loại đặc trưng trên tập dữ liệu MNIST

### 3.3.4. Kết quả thực nghiệm trên tập dữ liệu chữ viết tay tiếng Việt

Trong thực nghiệm nhận dạng chữ viết tay tiếng Việt, luận án cài đặt thử nghiệm trên hai loại đặc trưng:

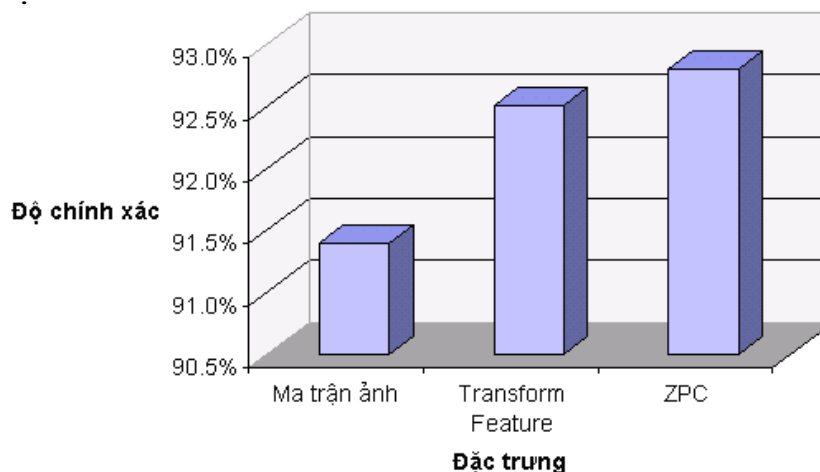
- Kết hợp các đặc trưng thống kê (ZPC - Zones, Projection histograms và Contour profiles): Với phần chữ, chọn kích thước ảnh 16x16, như vậy mỗi vectơ đầu vào có tất cả :  $64 + 94 + 64 = 222$  đặc trưng, còn phần dấu với kích thước ảnh 8x8 thì vectơ đầu vào sẽ có  $16 + 46 + 32 = 94$  đặc trưng.
- Đặc trưng TransformFeature: Với phần chữ, chọn kích thước ảnh 16x16, như vậy mỗi vectơ đầu vào có 256 đặc trưng, còn phần dấu với kích thước ảnh 8x8 thì vectơ đầu vào có tất cả 64 đặc trưng.

Để so sánh hiệu quả nhận dạng của các tập đặc trưng được đề xuất đối với dữ liệu chữ viết tay tiếng Việt, bộ dữ liệu VietData được chia thành 10 phần và Bảng 3.3 là kết quả trung bình của 10 lần test trên cùng một Thuật toán 3.1.

Bảng 3.3. So sánh hiệu quả sử dụng các tập đặc trưng khác nhau trên tập dữ liệu VietData.

Đặc trưng	Số đặc trưng	Độ chính xác
Ma trận ảnh 16x16	256	91.4%
TransformFeature	256	92.5%
ZPC	222	92.8%

Kết quả ở Bảng 3.3 và Hình 3.11 cho thấy việc áp dụng các tập đặc trưng ZPC và TransformFeature đã góp phần nâng cao độ chính xác phân lớp đối với dữ liệu chữ viết tay tiếng Việt.



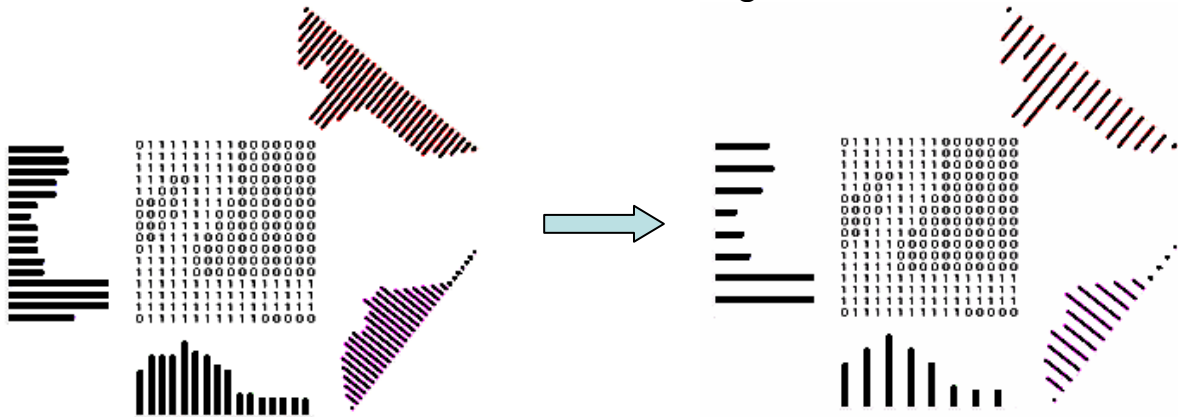
Hình 3.11. Hiệu quả phân lớp của các loại đặc trưng trên tập dữ liệu VietData.

### 3.4. CẢI TIẾN TỐC ĐỘ NHẬN DẠNG

Trong phần này, luận án tập trung cải tiến tốc độ nhận dạng bằng cách rút gọn số chiều của vectơ đặc trưng đầu vào và áp dụng phương pháp tập thu gọn [9] để giảm thiểu tối đa số vectơ tựa của các máy phân lớp SVM nhị phân nhằm cải tiến tốc độ nhận dạng.

#### 3.4.1. Rút gọn số chiều của vectơ đặc trưng

Phần này tìm cách rút gọn số chiều của tập đặc trưng ZPC. Đối với các đặc trưng biểu đồ chiều và chu tuyến, thay vì lấy toàn bộ các đặc trưng, đối với mỗi dòng, cột và hai đường chéo của ảnh, chỉ lấy xen kẽ các đặc trưng để giảm đi một nửa đặc trưng của vectơ đầu vào (Hình 3.12). Như vậy đối với ảnh kích thước 16×16, vectơ đầu vào chỉ còn lại 64 + 48 + 32 = 144 đặc trưng, còn đối với ảnh kích thước 8×8, vectơ đầu vào chỉ còn lại 16 + 24 + 16 = 56 đặc trưng.



Hình 3.12. Lấy xen kẽ các đặc trưng.

Việc rút gọn số chiều của vectơ đặc trưng sẽ làm tăng tốc độ phân lớp của giai đoạn nhận dạng. Việc làm này cũng có thể làm mất mát thông tin biểu diễn ảnh, điều này ảnh hưởng đến độ chính xác phân lớp của giai đoạn nhận dạng. Tuy nhiên, độ chính xác phân lớp của toàn hệ thống cũng có thể được cải thiện nhờ vào giai đoạn hậu xử lý.

#### 3.4.2. Cải tiến tốc độ của các máy phân lớp SVM

Phần này áp dụng phương pháp tập thu gọn theo hướng tiếp cận của Burges (1996) [12] nhằm tăng tốc độ nhận dạng của các máy phân lớp SVM nhị phân. Quá trình thu gọn các vectơ tựa được thực hiện bằng cách lựa chọn hai vectơ tựa gần nhất của cùng một lớp và thay thế chúng bằng một vectơ mới cho tới khi có được tập vectơ thu gọn tương đương với tập các vectơ tựa ban đầu.

##### 3.3.2.1. Phương pháp tập thu gọn

Đối với bài toán phân lớp nhị phân, luật quyết định SVM được cho bởi công thức:

$$y = \text{sign} \left( \sum_{i=1}^{N_{SV}} \alpha_i K(x, x_i) + b \right) \quad (3.1)$$

trong đó  $\alpha_i$  là các trọng số của các vectơ tựa  $x_i$ ,  $x$  là vectơ đầu vào cần phân lớp,  $K(x, x_i)$  là một hàm nhân trong không gian đặc trưng,  $b$  là độ lệch của siêu phẳng so với gốc tọa độ và  $N_{SV}$  là số vectơ tựa.

Ý tưởng chính của phương pháp tập thu gọn là tìm số nhỏ nhất  $N_Z < N_{SV}$ , và tập thu gọn tương ứng  $\{(z_i, \beta_i)\}$ ,  $i=1 \dots N_Z$  sao cho  $\rho = |\Psi' - \Psi|$  đạt cực tiểu, tức là tìm cách xấp xỉ vectơ chuẩn  $\Psi$  của siêu phẳng tối ưu gốc bởi vectơ chuẩn  $\Psi'$  của siêu phẳng

tối ưu theo tập thu gọn sao cho sai số về độ chính xác phân lớp của SVM đạt cực tiểu, trong đó:

$$\Psi = \sum_{i=1}^{N_{SV}} \alpha_i \Phi(x_i) \text{ và } \Psi' = \sum_{j=1}^{N_Z} \beta_j \Phi(z_j)$$

với  $N_Z < N_{SV}$ ,  $z_i \in R^D$ ,  $\beta_i \in R$ .

Như vậy, để phân lớp một mẫu  $x$ , (3.1) được thay bởi

$$y = \text{sign} \left( \sum_{i=1}^{N_Z} \beta_i K(x, z_i) + b \right) \quad (3.2)$$

trong đó  $N_Z < N_{SV}$ , và tập thu gọn tương ứng là  $\{(z_i, \beta_i)\}_{i=1, \dots, N_Z}$ .

### 3.3.2.2. Thuật toán tìm tập thu gọn

Thuật toán này lặp lại việc chọn lựa hai vectơ tựa trong cùng một lớp và thử thay thế chúng bởi một vectơ mới. Thuật toán dừng lại khi không còn việc thay thế nào được thực hiện, cuối cùng cập nhật lại tất cả các vectơ thu gọn và các hệ số của nó để đạt được một xấp xỉ tốt hơn:

---

#### Thuật toán 3.3: ReduceSet

// Thuật toán tìm tập vectơ thu gọn

##### Input

Tập gồm  $N_{SV}$  vectơ tựa  $\{x_1, \dots, x_{N_{SV}}\}$ ;

Ngưỡng MMD  $\theta$ ;

##### Output

Một tập gồm  $N_Z$  vectơ thu gọn,  $N_Z < N_{SV}$ .

##### Method

1. PairList =  $\{(x_i, x_j) \mid i=1, \dots, N_{SV}, \arg \min_k (||x_i - x_k||^2), 1 \leq k \leq N_{SV}, \alpha_i \alpha_k > 0\}$ ;
  2. Sắp xếp các cặp trong PairList tăng dần theo độ sai lệch giữa hai vectơ trong mỗi cặp;
  3. PairID = 1;  $N_Z = N_{SV}$ ;
  4. **Repeat**  
Gọi  $(x_i, x_j)$  là cặp được đánh chỉ số PairID trong PairList;  
Thử thay  $x_i$  và  $x_j$  bởi  $z$  theo (3.7) hoặc (3.9) và trọng số  $\beta$  của  $z$  bởi (3.12);  
**if** (việc thay thế tạo ra  $\text{MMD} \leq \theta$ )  
    { Thay  $x_i$  và  $x_j$  bởi  $z$ ;  $N_Z = N_Z - 1$ ;  
    Cập nhật PairList;  
    PairID = 1; }  
    **else** PairID = PairID + 1;
  5. **Until** <tất cả các cặp trong PairList đều đã thử>;
  6. Tính lại các hệ số của tất cả các vectơ thu gọn bằng cách sử dụng (3.13);
  7. **Return** Tập thu gọn;
- 

### 3.4.3. Kết quả thực nghiệm

Bảng 3.4 trình bày một số kết quả thực nghiệm trên tập dữ liệu VietData theo Thuật toán 3.1 với bốn phương án:

**ORG**: sử dụng tập đặc trưng ZPC.



**RDM:** sử dụng tập đặc trưng ZPC thu gọn.

**RSM:** sử dụng tập đặc trưng ZPC và SVM với tập thu gọn.

**RDM+RSM:** kết hợp cả hai phương án RDM và RSM.

*Bảng 3.4. Kết quả nhận dạng trên tập dữ liệu VietData.*

<b>Phương án</b>	<b>Độ chính xác</b>	<b>Thời gian</b>
ORG	92.8%	128 giây
RDM	92.3%	97 giây
RSM	92.5%	23 giây
<b>RDM+RSM</b>	<b>92.2%</b>	<b>17 giây</b>

Kết quả thực nghiệm ở Bảng 3.4 cho thấy mô hình cải tiến sử dụng cả hai phương án RDM và RSM thì tốc độ nhận dạng tăng lên gần 8 lần so với phương án gốc với sai số về độ chính xác phân lớp có thể chấp nhận được. Tùy theo tiêu chí của từng ứng dụng cụ thể, nếu ưu tiên tốc độ thì có thể hy sinh chút ít về độ chính xác trong giai đoạn nhận dạng.

### **3.5. KẾT LUẬN**

Chương này đã đề xuất một giải pháp hiệu quả để triển khai ứng dụng cho bài toán nhận dạng chữ Việt viết tay rời rạc. Chương này cũng đề xuất một phép biến đổi để biểu diễn đặc trưng hiệu quả nhằm tăng độ chính xác nhận dạng chữ viết tay rời rạc. Chương này cũng đã cải tiến tốc độ nhận dạng của hệ thống bằng cách rút gọn số chiều của các vectơ đặc trưng kết hợp với giải pháp giảm tối đa số vectơ tựa của các máy phân lớp SVM nhị phân. Các kết quả của Chương này đã được tác giả công bố trong các công trình [4,5,6,7].

# PHẦN KẾT LUẬN

## 1. Các kết quả chính của luận án

- Nghiên cứu xây dựng thuật toán nhận dạng chữ viết tay rời rạc theo phương pháp phân lớp SVM với các chiến lược OVO và OVR. Phân tích, đánh giá những ưu và nhược điểm của kỹ thuật phân lớp SVM trong nhận dạng chữ viết tay rời rạc thông qua các kết quả thực nghiệm trên các tập dữ liệu chuẩn USPS, MNIST và dữ liệu chữ viết tay tiếng Việt. Các kết quả này đã được tác giả công bố trong các công trình [1,2,3,8].
- Đề xuất một giải pháp hiệu quả cho bài toán nhận dạng chữ Việt viết tay rời rạc sử dụng kỹ thuật phân lớp SVM. Thông qua việc phân tích các thành phần liên thông của ảnh để phân hoạch tập ký tự tiếng Việt thành 3 nhóm và tách các ký tự có dấu thành các phần rời nhau để nhận dạng độc lập từng phần chữ và phân dấu. Điều này giúp cho thuật toán phân lớp giảm đáng kể số phân lớp nhị phân, tăng tốc độ nhận dạng. Kết quả này đã được tác giả công bố trong [4].
- Đề xuất phép biến đổi ảnh hai chiều thành chuỗi các đặc trưng hiệu quả cho bài toán nhận dạng chữ viết tay rời rạc và đã chứng minh được tính duy nhất của chuỗi đặc trưng theo phép biến đổi này. Kết quả này đã được tác giả công bố trong [6,7].
- Cải tiến tốc độ nhận dạng chữ Việt viết tay rời rạc bằng cách áp dụng kỹ thuật giảm số chiều của các vectơ đặc trưng đầu vào và giảm số vectơ tựa của các máy phân lớp SVM nhị phân. Kết quả này đã được tác giả công bố trong [5].
- Luận án đã xây dựng được một cơ sở dữ liệu chữ viết tay tiếng Việt với hơn 100000 mẫu ký tự chữ viết tay rời rạc. Đây là một nguồn dữ liệu rất cần thiết cho việc nghiên cứu nhận dạng chữ viết tay tiếng Việt trong tương lai.

## 2. Hướng nghiên cứu

Nhận dạng chữ viết tay là lĩnh vực nghiên cứu khó, liên quan đến nhiều lĩnh vực khác như xử lý ảnh, trí tuệ nhân tạo, lý thuyết học máy, xác suất thống kê, toán ứng dụng... Vì vậy, cho đến nay nhận dạng chữ viết tay vẫn là một bài toán mở thách thức các nhà nghiên cứu. Trong tương lai, luận án sẽ tiếp tục phát triển theo các hướng sau:

- Tiếp tục nghiên cứu để xử lý các trường hợp phân dấu bị dính với phần thân chữ.
- Nghiên cứu các tính chất bất biến trên các dạng chữ viết tay để trích chọn đặc trưng phù hợp phục vụ cho việc nhận dạng.
- Nghiên cứu, phát triển mô hình nhận dạng cho chữ Việt viết thường.
- Nghiên cứu kết hợp các phương pháp phân lớp nhằm nâng cao chất lượng cũng như tốc độ nhận dạng.
- Nghiên cứu phát triển giai đoạn hậu xử lý để nâng cao chất lượng của hệ thống nhận dạng.
- Xây dựng các ứng dụng nhận dạng và xử lý các biểu mẫu tự động.

## DANH MỤC CÁC CÔNG TRÌNH ĐÃ CÔNG BỐ CỦA TÁC GIẢ

- [1] Phạm Anh Phương, Ngô Quốc Tạo, Lương Chi Mai (2006), “Ứng dụng SVM cho bài toán phân lớp nhận dạng”, *Kỷ yếu Hội thảo khoa học Quốc gia lần thứ ba về nghiên cứu, phát triển và ứng dụng Công nghệ thông tin và Truyền thông (ICT.rda'06)*, nhà xuất bản Khoa học và Kỹ thuật, Hà nội, tr. 393-400.
- [2] Phạm Anh Phương (2007), “Nhận dạng chữ viết tay hạn chế với mô hình SVM”, *Tạp chí khoa học Đại học Huế*, ISSN 1859-1388, số 42, tr. 157-165.
- [3] Phạm Anh Phương (2008), “Áp dụng một số chiến lược SVM đa lớp cho bài toán nhận dạng chữ viết tay hạn chế”, *Tạp chí khoa học Đại học Huế*, ISSN 1859-1388, số 45, tr. 109-118.
- [4] Phạm Anh Phương, Ngô Quốc Tạo, Lương Chi Mai (2008), “An Efficient Model for Isolated Vietnamese Handwritten Recognition”, *The Fourth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, IEEE Computer Society, Harbin, China, pp. 358-361.
- [5] Phạm Anh Phương, Ngô Quốc Tạo, Lương Chi Mai (2008), “Speeding Up Isolated Vietnamese Handwritten Recognition by Combining SVM and Statistical Features”, *IJCSES International Journal of Computer Sciences and Engineering Systems*, ISSN 0973-4406, Vol.2, No.4, pp. 243-247.
- [6] Phạm Anh Phương, Ngô Quốc Tạo, Lương Chi Mai (2008), “Trích chọn đặc trưng wavelet Haar kết hợp với SVM cho việc nhận dạng chữ viết tay tiếng Việt”, *Tạp chí Công nghệ Thông tin và Truyền thông*, ISSN 0866-7039, kỳ 3, số 20, tr. 36-42.
- [7] Phạm Anh Phương, Ngô Quốc Tạo, Lương Chi Mai (2009), “Kết hợp các bộ phận phân lớp SVM cho việc nhận dạng chữ Việt viết tay rời rạc”, *Tạp chí Tin học và Điều khiển học*, ISSN 1813-9663, Tập 25, số 1, tr. 88-97.
- [8] Phạm Anh Phương, “Nghiên cứu lý thuyết SVM và mô hình ứng dụng cho bài toán nhận dạng chữ viết tay hạn chế”, *Đề tài nghiên cứu khoa học cấp Bộ năm 2007-2008*.