Tạp chí Tin học và Điều khiển học, T.22, S.2 (2006), 141–154

PHƯƠNG PHÁP NHẬN DẠNG TỪ VIẾT TAY DỰA TRÊN MÔ HÌNH MẠNG NƠRON KẾT HỢP VỚI THỐNG KÊ TỪ VỰNG

NGUYỄN THỊ THANH TÂN, LƯƠNG CHI MAI

Viện Công nghệ thông tin

Abstract. This paper presents a new method for off-line word handwriting recognition using four layer neural networks combined with vocabulary statistics. The process of recognition is, basically, a sequence of tasks, which are composed of two following sub-tasks: 1) Detect all possible cut positions on the current segment of input image based on its histogram and a set of defined characteristic curves; 2) Recognize those specific cut positions and remove bad cut positions based on the occurence probability of pair of charaters. The resulted set of word candidates is fed into the system to choose the best one. This approach was experimented on English handwriting words extracted from AIM database gaining the accuracy of 78% for properly written words, i.e. neat and eligible words with no overlapped characters.

Tóm tắt. Bài báo này đề xuất một phương pháp mới nhận dạng từ viết tay dựa trên mô hình mạng nơron kết hợp với thống kê từ vựng. Về cơ bản, quá trình nhận dạng là một chuỗi các thao tác xử lý được thực hiện một cách tuần tự trên từng phần ảnh, trong đó mỗi thao tác xử lý bao gồm hai bước chính: 1) Hệ thống xác định tất cả các vị trí cắt có thể có trên phần ảnh hiện tại dựa vào biểu đồ tần suất và tập các đường cong đặc trưng đã được định nghĩa; 2) Nhận dạng các vị trí cắt, loại bỏ các khả năng sai dựa trên xác suất xuất hiện của các cặp ký tự. Kết quả thu được của quá trình nhận dạng là một tập các từ ứng cử viên. Lúc này hệ thống sẽ sử dụng dữ liệu thống kê từ vựng để lựa chọn ứng cử viên tốt nhất từ tập trên. Hiện tại phương pháp này đã được thử nghiệm trên bộ dữ liệu được trích từ cơ sở dữ liệu từ tiếng Anh viết tay IAM, cho độ chính xác trên 78% đối với các từ được viết tương đối rõ ràng (chữ viết ngay ngắn, không viết ngoáy, các ký tự không lồng nhau).

1. GIỚI THIỆU

Khi đề cập đến bài toán nhận dạng chữ viết tay, người ta thường chia thành hai loại: Chữ viết tay ngoại tuyến (off-line handwriting) - kiểu chữ được viết trên văn bản giấy sau đó đưa vào máy tính thông qua các thiết bị scanner, camera... và chữ viết tay trực tuyến (on-line handwriting) - kiểu chữ viết trực tiếp bằng thiết bị bút từ trên các màn hình cảm ứng của máy tính (Tablet PC), các thiết bị cầm tay (PDA) hoặc trên các thiết bị bảng điện tử được nối trực tiếp với máy tính. Đối tượng chính được xét đến trong bài báo này là các văn bản chữ viết tay ngoại tuyến chứa các từ được viết tương đối rõ ràng (chữ viết ngay ngắn, không viết ngoáy, các ký tự không lồng nhau). Mục đích nghiên cứu của chúng tôi là xây dựng được thuật toán nhận dạng chữ Việt viết tay có chất lượng đủ tốt để nâng cao chất lượng nhận dạng của phần mềm nhận dạng chữ Việt in VnDOCR trong nhận dạng các

văn bản chữ in đầu vào có lẫn một số từ viết tay hoặc các văn bản có chất lượng không tốt (chữ in bị nhoè, bị dính, rất khó để phân biệt ranh giới chính xác giữa các ký tự).

Để giải quyết bài toán nhận dạng chữ viết tay, các nhóm nghiên cứu trên thế giới hiện nay đã và đang tập trung theo hai hướng chính ([1, 15, 16]).

+ Hướng thứ nhất (dựa trên hướng tiếp cận nhận dạng chữ in) ([5]): Để nhận dạng một khối văn bản (khối text), trước tiên khối văn bản đó phải được phân nhỏ thành từng dòng (line segmentation), các dòng này sau đó được phân nhỏ tiếp thành từng từ (word segmentation), cuối cùng tiến hành phân đoạn từ thành các ký tự (character segmentation) và đây mới là đầu vào thực sự của engine nhận dạng. Ưu điểm của cách tiếp cận này là kích thước của đối tượng cần nhận dạng nhỏ và tương đối đồng đều (kích thước không quá chênh lệch nhau), như vậy việc trích chọn các đặc trưng nhận dạng sẽ đơn giản hơn và độ chính xác của engine nhận dạng sẽ cao hơn. Tuy nhiên, điểm khó khăn nhất của cách tiếp cận này cũng chính là điểm khó khăn của bài toán nhận dạng chữ viết tay: Làm sao để xác định được chính xác vị trí phân tách của các ký tự trên một từ khi chúng bị viết dính vào nhau? Để giải quyết khó khăn này, một số nhóm tác giả đã đề xuất phương pháp phân đoạn từ (character segmentation) theo hướng heuristic ([3, 12]). Tuy nhiên hiện cũng chưa có phương pháp nào phân tách được với độ chính xác đến 90% và như vậy ở giai đoạn sau nếu có sử dụng các mô hình ngôn ngữ (n-grams) để hiệu chỉnh các kết quả nhận dạng thì cũng không thể tăng được chất lượng nhận dạng.

+ Hướng tiếp cận thứ hai (holistic word recognition) ([4]): Chỉ phân tách khối văn bản cần nhận dạng đến mức từ (thực hiện bước line segmentation và word segmentation), sau đó sẽ tiến hành nhận dạng toàn bộ từ dựa trên các đặc trưng của nó ([9]). Về mặt lý thuyết thì cách tiếp cận này sẽ tránh được vấn đề khó khăn của cách tiếp cận thứ nhất và thích hợp hơn với bài toán nhận dạng chữ viết tay. Tuy nhiên, để xây dựng được một phương pháp trích chọn đặc trưng của từ đủ tốt mà đảm bảo được tốc độ tính toán (tốc độ huấn luyện, tốc độ nhận dạng) và chất lượng nhận dạng (không phụ thuộc nhiều vào kích thước, độ nghiêng, độ dày/mỏng của nét chữ,...) lại là một vấn đề khó và hiện vẫn là một vấn đề mở.

Trong bài báo này chúng tôi đề xuất một phương pháp nhận dạng từ theo hướng tiếp cận khác với các hướng tiếp cận trên. Với cách tiếp cận truyền thống thì *phân đoạn từ* (tách thành từng ký tự) và *nhận dạng ký tự* là hai bước tuần tự, tách rời nhau trong quá trình nhận dạng. Trong khi đó, với cách tiếp cận của chúng tôi, hai bước này không tách rời nhau mà được thực hiện một cách đan xen nhau. Về cơ bản, quá trình nhận dạng có thể được mô tả như một chuỗi các thao tác xử lý, thực hiện đệ quy trên từng phần ảnh đầu vào tương ứng, trong đó mỗi *thao tác xử lý* bao gồm hai bước chính:

1) Hệ thống xác định tất cả các vị trí cắt có thể có trên phần ảnh đầu vào hiện tại dựa vào biểu đồ tần suất và tập các đường cong đặc trưng (định nghĩa ở Phần 2).

2) Nhận dạng các vị trí cắt bằng mô hình mạng nơron bốn lớp [7] và loại bỏ các khả năng sai dựa trên xác suất xuất hiện của các cặp ký tự. Thuật toán nhận dạng sẽ bắt đầu thao tác xử lý đầu tiên trên ma trận ảnh gốc của từ cần nhận dạng, sau khi thực hiện xong Bước 1 (xác định các vị trị trí cắt) sẽ tìm được một tập các vị trí cắt có thể có, thuật toán sẽ xét lần lượt K vị trí cắt đầu tiên trong danh sách các vị trí cắt đã tìm được. Tại mỗi vị trí cắt, ảnh đầu vào sẽ được chia thành hai phần: phần ảnh của ký tự được tách ra và phần ảnh của

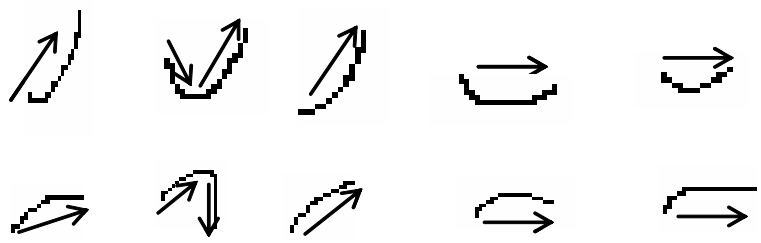
các ký tự còn lại chưa được tách. Lúc này sẽ thực hiện Bước 2 (nhận dạng và kiểm tra vị trí cắt) trên phần ảnh được cắt ra, nếu thành công (vị trí cắt này hợp lệ) thì sẽ ghi nhận vị trí này và lặp lại quá trình xử lý trên tức là thực hiện thao tác xử lý thứ hai với phần ảnh còn lại, quá trình này được thực hiện đệ quy cho đến khi kích thước của phần ảnh còn lại bằng 0. Sau đó, thuật toán lại lặp lại quá trình xử lý tương tự (thực hiện các thao tác xử lý tiếp theo) với các vị trí còn lại chưa được xét. Cuối cùng, sau khi tất cả các vị trí cắt đã được xét, thuật toán sẽ cho ra một tập các từ ứng cử viên. Lúc này hệ thống sẽ sử dụng dữ liệu thống kê từ vựng để lựa chọn ứng cử viên tốt nhất từ tập trên.

Mục 2 sẽ mô tả chi tiết hơn các bước của thuật toán xác định các vị trí cắt trên ảnh đầu vào, Mục 3 mô tả cụ về cấu trúc mạng nơron bốn lớp, cách thức trích chọn đặc trưng của đối tượng cần nhận dạng, các kết quả đối sánh và thử nghiệm mạng, Mục 4 đề cập đến một số khái niệm đã sử dụng trong thống kê từ vựng, quá trình tạo từ điển từ vựng để lưu trữ các thông tin phục vụ cho nhận dạng, Mục 5 mô tả chi tiết về thuật toán nhận dạng từ dựa trên thuật toán xác định vị trí cắt, mạng nơron bốn lớp và từ điển từ vựng với các kết quả thực nghiệm cụ thể. Cuối cùng phần kết luận sẽ tổng kết lại những kết quả hiện tại đã đạt được và một số đề xuất cho hướng phát triển tiếp theo.

2. PHƯƠNG PHÁP XÁC ĐỊNH CÁC VỊ TRÍ CẮT TRÊN ẢNH ĐẦU VÀO

Từ những nghiên cứu một cách trực quan về chữ viết tay cho thấy: Trên một văn bản viết tay bất kỳ thường có những ký tự được viết dính nhau, các vị trí dính nhau thường nằm ở phần giao giữa biên bên trái của ký tự thứ nhất và biên bên phải của ký tự thứ hai và đường nối giữa các ký tự thường có dạng các đường cong đặc trưng với độ cong và kích thước khác nhau (Hình 1).

Hình 1. Một số kiểu chữ viết tay - chữ viết dính nhau



Hình 2. Một số kiểu đường cong đặc trưng nối giữa hai ký tự

Nghiên cứu trên tập mẫu từ viết tay đã thu thập được (từ nhiều người viết khác nhau), chúng tôi nhận thấy các đường cong đặc trưng nối giữa hai ký tự thường ở một số dạng cơ bản như trên Hình 2. Trên cơ sở đó, chúng tôi đề xuất thuật toán xác định các điểm cắt trên ảnh đầu vào dựa trên việc xác định các vị trí có mật độ thấp trên biểu đồ tần xuất theo chiều thẳng đứng của khối ảnh và xác định tâm điểm của các đường cong đặc trưng:

Thuật toán 2.1. Thuật toán xác định các vị trí cắt trên một ảnh đầu vào

INPUT:

- StartPos, EndPos: tọa độ bắt đầu và kết thúc của vùng ảnh hiện thời đang được xét (tính theo ảnh gốc).

- Ma trận ảnh gốc của từ cần nhận dạng.

- Tập luật xác định các đường cong đặc trưng.

OUTPUT:

Danh sách các vị trí cắt đã tìm được, trong đó thông tin về mỗi vị trí cắt là: tọa độ bắt đầu (được tính bằng tọa độ bắt đầu của vùng ảnh hiện tại), và tọa độ của vị trí cắt tìm được.

PROCESS:

Bước 1: Tính histogram (biểu đồ tần suất) theo chiều thẳng đứng của ảnh đầu vào.

Bước 2: Tìm kiếm tất cả các vị trí có mật độ tần suất thấp (≈ 0), trường hợp nhiều vị trí liền nhau cùng có mật độ tần suất $= 0$, ta sẽ chọn vị trí đầu tiên và bỏ qua các vị trí liền kề nó cho đến khi gặp điểm có mật độ tần suất $\neq 0$. Lưu các vị trí cắt đã tìm được vào một danh sách (list).

Bước 3: Xác định tất cả các đường cong đặc trưng trên ảnh (dựa vào tập các đường cong đã được định nghĩa ở trên).

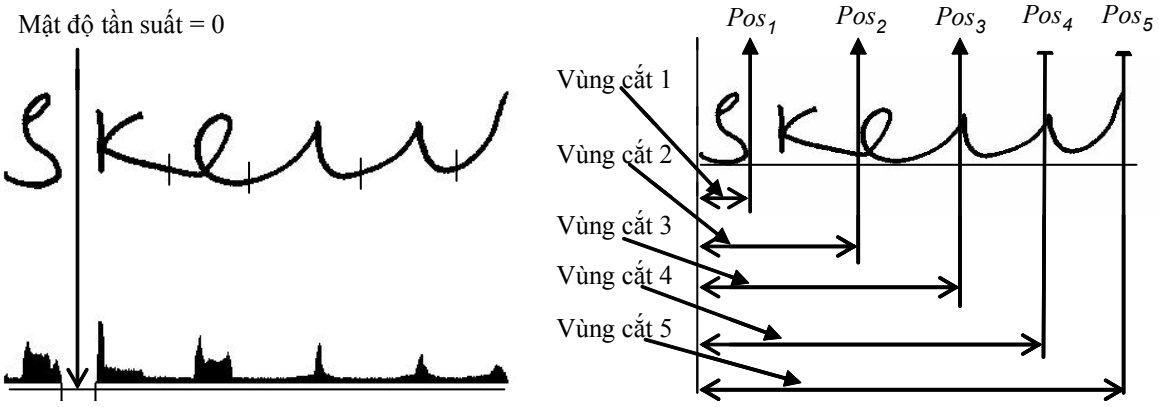
Bước 4: Với mỗi đường cong đặc trưng đã tìm được, tiến hành:

1. Tìm kiếm các điểm chạc (thường là điểm chạc 3, chạc 4) trong khoảng $[x - \delta, x + \delta]$, với x là tọa độ điểm mút bên phải của đường cong đặc trưng và δ là hệ số để điều chỉnh khoảng tìm kiếm. Nếu có nhiều điểm cùng thỏa mãn sẽ ưu tiên chọn các điểm phía bên phải. Nếu không tìm được điểm chạc nào, ta sẽ chọn điểm mút hoặc trung điểm của đường cong này làm điểm cắt.

2. Lưu vị trí cắt này: Lưu tọa độ bắt đầu của vùng ảnh đang xét (StartPos) và tọa độ điểm cắt vừa tìm được vào danh sách trên.

Bước 5: Sắp xếp lại danh sách theo chiều tăng dần của tọa độ x của điểm cắt. Trả lại kết quả là danh sách các vị trí cắt đã tìm được.

Việc xác định các vị trí cắt dựa vào các điểm có mật độ tần suất thấp cho phép ta xác định được điểm cắt chính xác giữa hai ký tự không dính nhau. Việc xác định các vị trí cắt dựa trên các đường cong đặc trưng cho phép xác định được vị trí cắt tương đối giữa các ký tự dính nhau. Chẳng hạn, với ảnh đầu vào của từ "skew", ở lần duyệt đầu tiên, thuật toán sẽ tìm được 5 vị trí cắt ($pos_1 \rightarrow pos_5$) như trên Hình 5. Trong đó vị trí cắt thứ nhất là điểm có mật độ tần suất nhỏ nhất ($= 0$), các vị trí còn lại tìm được dựa trên các đường cong đặc trưng.



Hình 3. Kết quả của thuật toán xác định các vị trí cắt khác nhau trên ảnh đầu vào

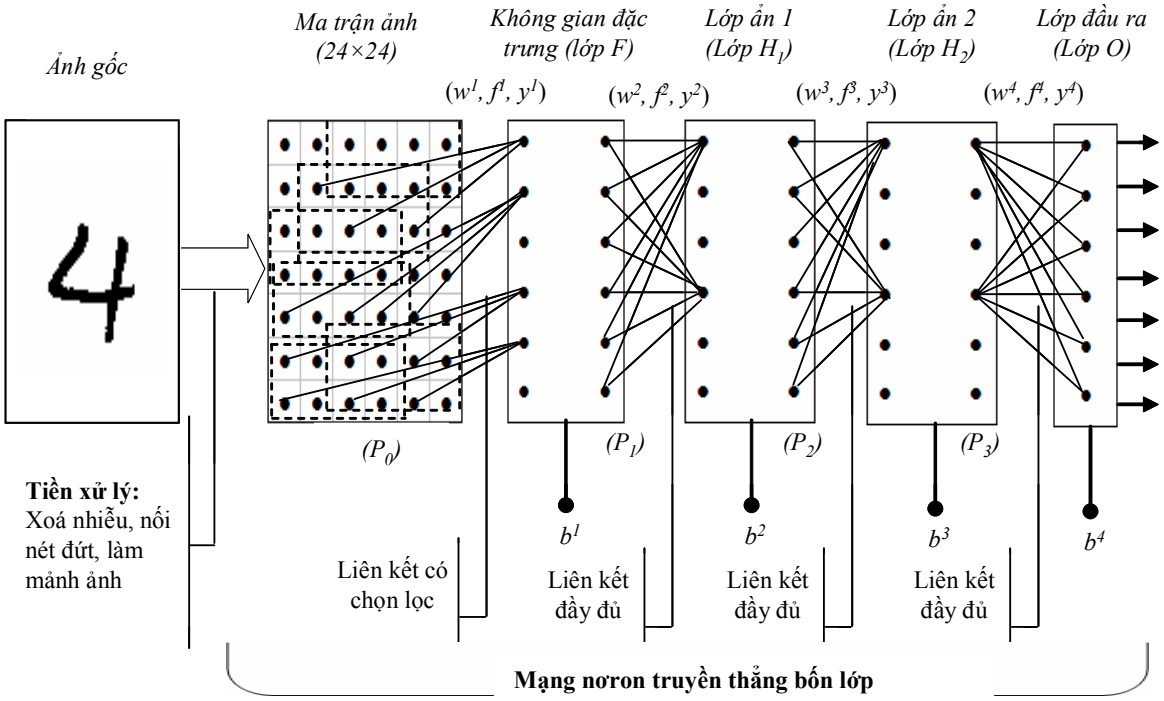
3. XÂY DỰNG MẠNG NƠRON BỐN LỚP ĐỂ NHẬN DẠNG CÁC VỊ TRÍ CẮT

Như chúng ta đều biết, đối với bài toán nhận dạng chữ viết tay, ngoài vấn đề làm sao để xác định được chính xác vị trí phân cách giữa các ký tự, chất lượng của thuật toán còn phải đảm bảo không bị ảnh hưởng nhiều bởi độ nghiêng, dộ dày mỏng và kích thước của các ký tự. Thực nghiệm đã cho thấy các thuật toán nhận dạng truyền thống như đối sánh mẫu, nhận dạng theo cấu trúc không đảm bảo được chất lượng khi sử dụng cho nhận dang chữ viết tay. Hiện đã có nhiều phương pháp thích hợp cho nhận dạng chữ viết tay đã được đề xuất như: phương pháp nhận dạng bằng quy hoạch động (dynamic programming) [14], bằng mô hình mạng nơron [11], mô hình support vector machine (SVM) [5], mô hình markov ẩn (HMM) [2]. Ở đây chúng tôi theo hướng tiếp cận nhận dạng bằng mô hình mạng nơron. Việc sử dụng mô hình mạng nơron cho nhận dạng có một số ưu điểm như: cài đặt đơn giản, việc huấn luyện mạng không phụ thuộc nhiều vào thứ tự của các mẫu học, việc biểu diễn và lưu trữ các tri thức đã học là đơn giản.

3.1. Cấu trúc mạng và việc trích chọn đặc trưng nhận dạng đối tượng

Với bất kỳ một mạng nơron nào thì đầu vào bao giờ cũng là các véc tơ đặc trưng của đối tượng cần nhận dạng. Có ba cách tiếp cận thường được sử dụng để xác định véc tơ đặc trưng của một đối tượng [6, 10]: Cách thứ nhất là lấy trực tiếp từng điểm ảnh trên ma trận ảnh đầu vào làm đầu vào của mạng, cách thứ hai là xây dựng các hàm để tính toán các đặc trưng và sử dụng kết quả của các hàm đó làm đầu vào của mạng, cách thứ ba là thiết kế các lớp mạng nơron để tính toán các đặc trưng đó một cách tự động. Cách tiếp cận nhất có ưu điểm là thực hiện đơn giản, không tốn nhiều thời gian tìm hiểu về đối tượng. Tuy nhiên, với cách tiếp cận này thì chất lượng nhận dạng của mạng thường không cao do bị ảnh hưởng nhiều bởi nhiễu. Ngoài ra, thời gian huấn luyện và tốc độ tính toán của mạng chậm do số lượng các liên kết trong mạng lớn. Cách tiếp cận thứ hai là cách tiếp cận hay được sử dụng nhất, có thể dùng cho nhiều phương pháp nhận dạng khác nhau chứ không chỉ riêng mạng nơron. Tuy nhiên, cách tiếp cận này phụ thuộc rất nhiều vào những tiêu chuẩn về đặc trưng

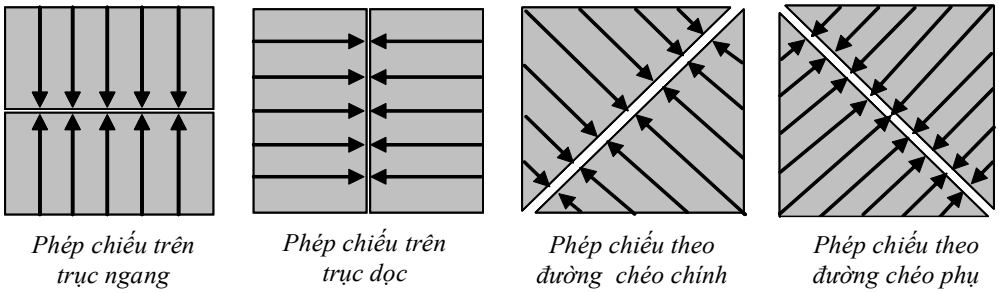
của đối tượng mà chúng ta đã đưa ra, nếu những tiêu chuẩn đưa ra không tốt thì những đặc trưng chọn được sẽ không phải là các đặc trưng nhận dạng đối tượng. Ở đây, chúng tôi đi theo cách tiếp cận thứ ba, chúng tôi đã xây dựng cấu trúc mạng bốn lớp [7, 8] (không kể lớp đầu vào).



Hình 4. Cấu trúc mạng nơron bốn lớp - trích chọn đặc trưng tự động

Lớp mạng được xây dựng để trích chọn các đặc trưng là lớp đầu tiên của mạng (ký hiệu lớp F - Hình 4). Đầu vào của lớp này là một mặt phẳng kích thước 24×24 (ký hiệu P_0), mỗi phần tử của P_0 tương ứng với một điểm ảnh của ma trận ảnh đầu vào (ánh xạ 1-1). Các noron của lớp F được phân bố trên mặt phẳng P_1 kích thước 6×24, trong đó mỗi nơron được liên kết với một vùng đặc trưng trên mặt phẳng đầu vào. Một vùng đặc trưng trên mặt phẳng đầu vào được định nghĩa là vùng chứa tập các điểm ảnh được chọn lựa theo một số quy tắc đã định trước, cụ thể ở đây chúng tôi đã chọn vùng đặc trưng theo các phép chiếu trên hai trục ngang/dọc và các phép chiếu theo hai đường chéo chính/phụ (Hình 5), trong đó bao gồm 24 vùng theo phép chiếu trên trục ngang, 24 vùng theo phép chiếu trên trục dọc, 48 vùng theo phép chiếu trên đường chéo chính và 48 vùng theo phép chiếu trên đường chéo phụ. Với cách cấu trúc như vậy, trong quá trình huấn luyện mạng (theo thuật toán truyền thẳng với sai số lan truyền ngược), mỗi nơron của lớp F sẽ không bị ảnh hưởng bởi tất cả các phần tử của mặt phẳng đầu vào mà chỉ chịu tác động của những phần tử nằm trong vùng đặc trưng liên kết với nó, nói cách khác là số liên kết của mỗi nơron được tính bằng số phần tử nằm trong vùng đặc trưng liên kết với nó. Cụ thể hơn, tổng giá trị giá trị kích hoạt lên nơron thứ i (ν_i^1) sẽ được tính theo công thức: $\nu_i^1 = \sum_{j=1}^{n} w_{i,j} x_j + b_i^1$, với x_j là giá trị của phần tử thứ j trên vùng đặc trưng, n là tổng số phần tử của vùng đặc trưng, b_i^1 là độ lệch

của nơron i.



Hình 5. Một số phép chiếu trên ảnh

Như vậy, rõ ràng với cách tiếp cận này, trọng số của các đặc trưng nhận dạng đối tượng (các đặc điểm chung của tập mẫu thuộc cùng một lớp đối tượng) sẽ dần dần được tăng lên, ngược lại, trọng số của các đặc trưng không dùng để nhận dạng đối tượng (không phải là những đặc trưng cơ bản của đối tượng) sẽ dần dần bị triệt tiêu trong quá trình huấn luyện mạng. Cơ chế này cho phép hạn chế được sự ảnh hưởng của nhiễu cũng độ nghiêng, độ dịch chuyển, phóng to, thu nhỏ của ảnh đầu vào. Ngoài ra, với cách tiếp cận này chúng ta cũng không cần đầu tư nhiều vào việc nghiên cứu và tìm hiểu tập mẫu để xác định được các tiêu chuẩn về đặc trưng của đối tượng như cách tiếp cận thứ hai.

3.2. Kết quả thực nghiệm

Chúng tôi đã tiến hành đối sánh khả năng nhận dạng của mạng nơron này với hai mạng nơron ba lớp: mạng nơron thứ nhất - sử dụng trực tiếp các điểm ảnh gốc làm đầu vào của mạng, mạng nơron thứ 2 - sử dụng kết quả của các hàm tính toán đặc trưng làm đầu vào của mạng (chúng tôi sử dụng 144 đặc trưng lấy từ 144 vùng đặc trưng như đã xác định ở trên, trong đó giá trị của mỗi đặc được tính bằng trung bình giá trị các điểm ảnh trong vùng đặc trưng đó). Việc thử nghiệm được tiến hành trên hai tập dữ liệu: *Tập chữ số viết tay* NIST của Viện Công nghệ và Tiêu chuẩn Quốc gia Hoa Kỳ (National Institute of Standard and Technology of the United States) bao gồm 60.000 mẫu học, 10.000 mẫu thử và tập chữ cái không dấu viết tay (được thu thập từ nhiều người viết khác nhau) gồm có 10.000 mẫu học và 1.816 mẫu thử. Kết quả thực nghiệm được thể hiện trên bảng 1.

4. TẠO TỪ ĐIỂN TỪ VỰNG PHỤC VỤ CHO QUÁ TRÌNH NHẬN DẠNG

Để có được các thông tin về từ vựng chúng tôi đã tiến hành thống kê trên 5GB dữ liệu đầu vào bao gồm các trang web (HTML), các file văn bản (text, word document), các file tài liệu thuộc các lĩnh vực như thể thao, văn hóa, kinh tế, xã hội, tin học, y tế,... trong đó xác suất xuất hiện của một chữ cái, của một cặp chữ cái và của một từ được định nghĩa như sau:

• Xác suất xuất hiện của ký tự ch trên tập dữ liệu D, ký hiệu là $XS(ch|D)$, được tính bằng tần suất xuất hiện của ký tự đó trên tổng tần suất xuất hiện của tất cả các ký tự có mặt trong tập dữ liêu đầu vào:

$$XS(ch|D) = \frac{TS(ch|D)}{\sum\limits_{ch_i \in D} TS(ch_i|D)},$$

trong đó $TS(ch|D)$ là tần suất xuất hiện của ch trên tập D, là tổng tần suất xuất hiện của tất cả các ký tự có mặt trên tập D.

Bảng 1

Các thông số	Mạng 3 lớp thứ nhất		Mạng 3 lớp thứ 2		Mạng 4 lớp	
	Tập chữ số	*Tập chữ cái*	*Tập chữ số*	*Tập chữ cái*	*Tập chữ số*	*Tập chữ cái*
Sai số mạng	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%
Số mẫu học	60.000	10.000	60.000	10.000	60.000	10.000
Thời gian học	>30 giờ	~5 giờ	~ 19 giờ	3 giờ	< 24 giờ	3.5 giờ
Tỷ lệ nhận dạng lại trên tập mẫu học	99.97% (18 mẫu nhận dạng sai/ 60.000 mẫu)	99.97% (3 mẫu nhận dạng sai/ 10.000 mẫu)	99.97% (18 mẫu nhận dạng sai/ 60.000 mẫu)	99.97% (3 mẫu nhận dạng sai/ 10.000 mẫu)	99.97% (18 mẫu nhận dạng sai/ 60.000 mẫu)	99.97% (3 mẫu nhận dạng sai/ 10.000 mẫu)
Số mẫu Test	10.000	1.816	10.000	1.816	10.000	1.816
Thời gian nhận dạng	~ 3 phút	~ 34 giây	~ 3 phút	~ 35 giây	~ 2 phút	~ 21 giây
Tỷ lệ nhận dạng đúng	82.2% (8220 mẫu nhận đúng / 10.000 mẫu)	78.5% (1426 mẫu nhận đúng/1816 mẫu)	91.7% (9170 mẫu nhận đúng / 10.000 mẫu)	~88.2%(160 1 mẫu nhận đúng / 1816 mẫu)	95.4%(9544 mẫu nhận đúng / 10.000 mẫu)	93% (1689 mẫu nhận đúng / 1816 mẫu)

• Xác suất xuất hiện của cặp ký tự xy (hay còn gọi là xác suất xuất hiện của ký tự y ngay sau ký tự x), ký hiệu $XS(xy)$, được tính bằng tần suất xuất hiện của ký tự y ngay sau ký tự x trên tổng tần suất xuất hiện của tất cả các ký tự có thể có ngay sau ký tự x:

$$XS(xy) = \frac{TS(xy|D)}{\sum\limits_{y_i \in D} TS(xy_i|D)},$$

trong đó $TS(xy|D)$ là tần suất suất hiện của cặp xy trên tập D, $\sum\limits_{y_i \in D} TS(xy_i|D)$ là tổng tần suất xuất hiện của tất cả các ký tự có thể có ngay sau x trên tập D.

• Xác suất xuất hiện của một từ w, ký hiệu $WXS(w)$, được tính bằng tích xác suất xuất hiện của tất cả các cặp ký tự liền kề nhau trong từ đó:

$$WXS(w) = XS(w_0w_1) \times XS(w_1w_2) \times ... \times XS(w_{n-1}w_n),$$

với n là chiều dài của từ.

5. THUẬT TOÁN NHẬN DẠNG TỪ

Với đặc thù của phương pháp (đã trình bày tương đối rõ ở Mục 1), ở đây chúng tôi sử dụng cấu trúc dữ liệu kiểu cây đa phân để lưu trữ các kết quả nhận dạng. Mỗi *thao tác xử lý* sẽ tương ứng với việc phát triển một nút trên cây. Mỗi nút sẽ lưu thông tin về kết quả

nhận dạng được từ một vị trí cắt đến các vị trí cắt có thể có ngay sau đó bao gồm ký tự nhận dạng được và danh sách các nút con:

struct CNode {
CNode *Parent; //Con trỏ đến nút cha
char *ch; //Mảng ký tự nhận dạng được tại vị trí cắt hiện tại (ít nhất là 1)
CPoint Pos //tọa độ của vị trí cắt hiện tại (tính theo ảnh gốc).
CTypedPtrList < CNode*,CPtrList > ListChild; //Danh sách nút con
}; Thuật toán được mô tả như sau:

Thuật toán 5.1. Thuật toán nhận dạng từ
INPUT: Ma trận ảnh của từ cần nhận dạng: WordIMG.
OUTPUT: Kết quả nhận dạng được.
PROCESS:
Bước 1: Khởi tạo:
- Khởi tạo cây lưu trữ *pParent*, CurrentNode = nút gốc (root) chứa ký tự $\emptyset$.
- Khởi tạo mạng nơron và các tham số cần thiết của mạng.
- Nạp từ điển từ vựng.
- Xác định vị trí bắt đầu và kết thúc của vùng ảnh hiện thời sẽ được xét:
StartPos = Tọa độ bắt đầu của ma trận ảnh gốc;
EndPos = Tọa độ kết thúc của ma trận ảnh gốc.

Bước 2: Kiểm tra xem kích thước của phần ảnh đầu vào: **EndPos - StartPos** > 0? Nếu đúng, tiến hành xác định tất cả các vị trí cắt có thể có trên phần ảnh đầu vào hiện tại (được xác định từ vị trí StarPos đến EndPos) theo Thuật toán 2.1 đã mô tả ở trên. Gọi danh sách các vị trí cắt đã tìm được là Ω. Sau đó thực hiện tiếp Bước 3.

Bước 3: Xét lần lượt K vị trí cắt đầu tiên trong danh sách các vị trí cắt đã tìm được:
For each AlnativeCut in Ω do{
1. Nhận dạng phần ảnh tương ứng bởi mạng nơron bốn lớp ở trên. Nếu kết quả nhận dạng là sai thì bỏ qua vị trí cắt này. Ngược lại, thực hiện tiếp thao tác 2 sau đây:
2. Kiểm tra xem các ký tự vừa nhận dạng được có hợp lý hay không (dựa trên xác suất xuất hiện của ký tự này ngay sau ký tự vừa nhận dạng được ở bước liền trước đó). Nếu tất cả các ký tự vừa nhận dạng được đều không hợp lệ thì bỏ qua vị trí cắt này. Ngược lại, thực hiện tiếp thao tác 3 và 4 sau đây:
3. Tạo nút mới:
newNode = new CNode;
newNode→ch = Mảng các ký tự vừa nhận dạng được;
newNode→Pos = Tọa độ của vị trí cắt hiện tại;
newNode→parent = CurrentNode;
newNode→ListChild = NULL
4. Chèn nút vừa tạo vào danh sách các nút con của nút cha hiện tại:
CurrentNode→ListChild→AddTail(newNode);
}//End For
Sau đó thực hiện Bước 4.

Bước 4: Kiểm tra xem danh sách các nút con hiện của nút cha hiện thời có rỗng không. Nếu không rỗng thì chuyển sang Bước 5. Ngược lại, danh sách rỗng (chứng tỏ vị trí cắt trước đó đã bị sai) thì tiến hành loại bỏ nút cha hiện thời này ra khỏi cây và chuyển sang Bước 6.
Bước 5: Chuyển sang nút con cả: CurrentNode = Con đầu tiên trong danh sách và thực hiện tiếp Bước 7.
Bước 6: Kiểm tra nếu danh sách em liền kề của nút cha hiện tại $\neq$ NULL, chuyển sang nút em liền kề này: **CurrentNode** = Em liền kề, thực hiện tiếp Bước 7. Ngược lại, kết thúc quá trình phát triển cây (lúc này tất cả các vị trí cắt đều đã được xét), chuyển sang Bước 8.
Bước 7: Thực hiện các thao tác sau:

1. Tính lại tọa độ bắt đầu và kết thúc của phần ảnh tiếp theo sẽ được xét:
StartPos = **CurrentNode→Pos**;
EndPos = tọa độ kết thúc của ma trận ảnh gốc.
2. Lặp lại các Bước 2, 3, 4 đối với phần ảnh đầu vào vừa xác định (**StartPos**, **EndPos**).

Bước 8: Tiến hành duyệt cây theo chiều sâu để lấy danh sách các từ ứng cử viên: Ghép các ký tự theo từng nhánh của cây, mỗi nhánh trên cây sẽ cho ra một ứng cử viên. Trường hợp có nhiều hơn một ký tự được lưu ở một nút thì coi mỗi ký tự đó như một nút con thực sự và tiến hành ghép như bình thường; Sau khi kết thúc quá trình duyệt cây (tất cả các nút trên cây đã được duyệt), thực hiện tiếp Bước 9.
Bước 9: (bước cuối cùng): Chọn ứng cử viên tốt nhất và trả về kết quả (ứng cử viên tốt nhất là từ có tần suất hoặc xác suất xuất hiện cao nhất trong tập ứng cử viên đã tìm được).
Chú ý:

Vấn đề chọn giá trị của tham số K: Về nguyên tắc, chúng ta cần xem xét tất cả các vị trí cắt và tìm ra vị trí cắt thích hợp nhất. Tuy nhiên, xuất phát từ thực tế là không có chữ cái nào có thể bao gồm quá 3 vị trí cắt (trường hợp các chữ cái có nhiều nét như chữ w hoặc chữ m) nên thường tham số K có giá trị không quá 3. Việc chọn giá trị cho K cũng giảm nhỏ số ứng viên cần đánh giá của phần thống kê. Phần sau đây sẽ trình bày một số tính chất của Thuật toán 5.1.

Tính chất 1. *Gọi n là số vị trí cắt tìm được ở "thao tác xử lý" đầu tiên (trên ảnh gốc của từ cần nhận dạng - Thuật toán 2.1), số ứng cử viên tối đa được sinh ra sẽ là 2^n từ (trường hợp không giới hạn K).*

Chứng minh:

Gọi F_n là số ứng viên cho một ảnh có n vị trí cắt, ta có:

+ Trường hợp không có vị trí cắt, dễ thấy là chỉ có 1 ứng viên duy nhất, vậy $F_0 = 1 (= 2^0)$.

+ Trường hợp có n vị trí cắt, xét vị trí cắt thứ n ta có 2 trường hợp xử lý:

i. Vị trí cắt này hợp lệ: Như vậy vị trí này tách miền ảnh liên thông thành 2 phần, phần đầu có $n-1$ vị trí cắt, phần sau không có vị trí cắt. Số ứng viên của trường hợp này là $F_{n-1} \times 1$.

ii. Vị trí cắt này không hợp lệ: Như vậy miền liên thông có thể xem như miền có $n-1$ vị trí cắt. Số ứng viên của trường hợp này là F_{n-1}.

+ Như vậy ta có $F_n = 2F_{n-1}$, khai triển công thức ta có:

$$F_n = 2F_{n-1} = \ = 2^i \times F_{n-i} = \ = 2^n \times F_0 = 2^n.$$

Tính chất 2. *Với việc giới hạn K vị trí cắt đầu tiên trong danh sách các vị trí cắt đã tìm được (Chọn $K = 3$ - Thuật toán 5.1), số ứng cử viên tối đa sinh ra đối với một ảnh có n vị trí cắt (ký hiệu F_n) sẽ được xác định theo công thức truy hồi sau đây:*

$$F_0 = 1, F_1 = 2, F_2 = 4, F_n = F_{n-1} + F_{n-2} + F_{n-3} \text{ với } n \geq 3.$$

Chứng minh:

– Trường hợp có ít hơn 3 vị trí cắt, $n < 3$, dễ thấy đây chính là trường hợp đã xét ở trên. Vậy số ứng cử viên tối đa được sinh ra là $F_n = 2^n$, cụ thể:

$$F_0 = 2^0 = 1,\ F_1 = 2^1 = 2,\ F_2 = 2^2 = 4.$$

– Trường hợp có ít nhất 3 vị trí cắt ($n \geq 3$):

+ Nếu vị trí cắt đầu tiên trong số K vị trí được xét là hợp lệ thì vị trí này tách miền liên thông thành 2 phần, phần đầu có 1 vị trí cắt, phần sau có $n - 1$ vị trí cắt. Số ứng cử viên của trường hợp này là F_{n-1}.

+ Nếu vị trí cắt thứ 2 trong số K vị trí được xét là hợp lệ thì vị trí này tách miền liên thông thành 2 phần, phần đầu có 1 vị trí cắt, phần sau có $n - 20$ vị trí cắt. Số ứng cử viên của trường hợp này là F_{n-2}.

+ Nếu vị trí cắt thứ 3 trong số K vị trí được xét là hợp lệ thì vị trí này tách miền liên thông thành 2 phần, phần đầu có 1 vị trí cắt, phần sau có $n - 3$ vị trí cắt. Số ứng cử viên của trường hợp này là F_{n-3}.

Như vậy, ta có $F_n = F_{n-1} + F_{n-2} + F_{n-3}$.

Tính chất 2 cho ta thấy việc giới hạn $K(K \leq 3)$ vị trí cắt đầu tiên trong danh các vị trí cắt đã tìm được sẽ làm giảm đáng kể số ứng cử viên được sinh ra đặc biệt đối các từ có nhiều vị trí cắt, chẳng hạn: Xét một từ có 10 vị trí cắt ($n = 10$), với trường tổng quát tổng quát thì số ứng cử viên sinh tối đa sinh ra là $F_{10} = 2^{10} = 1024$ từ. Trong khi đó, nếu ta giới hạn chỉ xét $K = 3$ vị trí đầu tiên, thì số ứng cử viên tối đa sinh ra là: $F_{10} = F_9 + F_8 + F_7 = 504$ từ ($F_0 = 1, F_1 = 2, F_2 = 4, F_3 = 7, F_4 = 13, F_5 = 24, F_6 = 44, F_7 = 81, F_8 = 149, F_9 = 274, F_{10} = 504$).

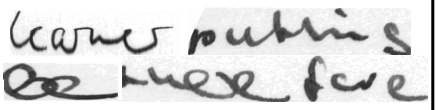
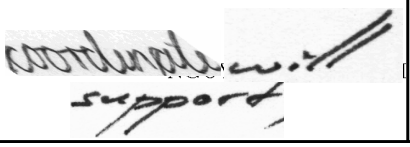
Kết quả thực nghiệm

Môi trường thử nghiệm

Chúng tôi đã cài đặt hệ thống trên ngôn ngữ Visual C++ 6.0, sử dụng thư viện CxImage cho việc hiển thị và thao tác trên ảnh. Chương trình được thử nghiệm trên hệ điều hành Window XP, máy tính PC tốc độ 2,4 GHz, bộ nhớ 512 MB RAM.

Dữ liệu và kết quả thử nghiệm

Mục đích ứng dụng của phương pháp này là nhận dạng chữ viết tay tiếng Việt. Tuy nhiên, do hiện tại chưa có được một cơ sở dữ liệu chữ viết tay tiếng Việt chuẩn để đánh giá nên hiện tại chúng tôi đã tiến hành thử nghiệm trên tập cơ sở dữ liệu từ viết tay tiếng Anh IAM [13], bao gồm 556 trang ảnh (form) được viết bởi 250 người viết khác nhau. Tập chữ cái để huấn luyện mạng nơron gồm có 55000 chữ (cắt ra từ 350 form được chọn một cách ngẫu nhiên). Tập từ để thử nghiệm gồm 13589 từ. Kết quả thử nghiệm như sau:

Chữ viết ngoáy, khó nhận biết		2411	~33.8 phút	46.8% (1128 từ đúng)	24.5 phút	52.4% (1263 từ đúng)
Chữ nghiêng quá $>15^0$		525	~7.6 phút	52.2% (273 từ đúng)	5.2 phút	57% (299 từ đúng)

Kết quả thực nghiệm cho thấy:

• Phương pháp này đạt chất lượng tốt khi chữ viết tay được viết rời nhau hoặc chữ viết dính nhau nhưng dễ nhận biết (không bị chồng lên nhau, không bị mất nét). Trong trường hợp chữ viết quá ngoáy hoặc nghiêng quá 15^0 thì chất lượng nhận dạng của phương pháp này không tốt. Tuy nhiên, trong thực tế, với những kiểu chữ viết quá xấu như vậy thì các cách tiếp cận khác cũng không cho chất lượng cao. Vì vậy, để giải quyết bài toán nhận dạng chữ viết, người ta thường phải đưa ra một số ràng buộc với tập dữ liệu đầu vào: chữ viết tương đối dễ nhận biết (đầy đủ nét và không xoắn hay lồng vào nhau).

• Việc sử dụng thông tin từ vựng ngay trong bước nhận dạng đã giúp loại bỏ được các vị trí cắt sai ngay khi nó được phát hiện, điều này làm giảm khả năng sai sót, giảm thời gian tính toán cũng như làm giảm số ứng cử viên cần đánh giá ở phần thống kê dẫn đến làm tăng chất lượng nhận dạng. Với kiểu chữ viết không bị dính nhau thì độ chênh lệch về thời gian nhận và tỷ lệ nhận dạng đúng giữa việc sử dụng từ điển từ vựng và không sử dụng từ điển là không nhiều (thời gian trung bình để nhận dạng một từ trong trường hợp không sử dụng từ điển nhanh hơn khoảng 0,007s). Tuy nhiên, đối với các kiểu chữ viết dính nhau (dễ xảy ra nhập nhằng) thì thời gian trung bình để nhận dạng một từ trong trường hợp sử dụng từ điển sẽ nhỏ hơn so với trường hợp không sử dụng từ điển khoảng từ 0,2s đến 0,3s và tỷ lệ nhận dạng đúng trong trường hợp này cũng tăng lên khoảng từ 3% đến 5%.

6. KẾT LUẬN

Trong bài báo này chúng tôi đã đề xuất một hướng tiếp cận mới để nhận dạng từ viết tay dựa trên mạng nơron kết hợp với thống kê từ vựng. Từ kết quả thử nghiệm đã có cũng như các đặc tính của phương pháp này cho thấy đây là một cách tiếp cận khả thi để nhận dạng chữ viết tay (đặc biệt là kiểu chữ viết bị dính nhau). Với phương pháp này, tất cả các vị trí cắt có thể có trên ảnh đầu vào đều được xem xét do vậy nếu chúng ta xây dựng được tập các đường cong đặc trưng đầy đủ thì các vị trí cắt đúng chắc chắn sẽ không bị bỏ qua, có nghĩa là kết quả đúng chắc chắn sẽ nằm trong tập ứng cử viên đã nhận dạng được. Bên cạnh đó, việc sử dụng độ đo *xác suất xuất hiện của một từ* để đánh giá khả năng lựa chọn của một từ là một giải pháp đủ tốt để chọn ra được kết quả đúng. Ngoài ra việc sử dụng các thông tin về từ vựng ngay trong bước nhận dạng sẽ giúp phương pháp này tránh được khả năng bùng nổ tổ hợp do đã phát hiện và loại bỏ được các vị trí cắt không hợp lý (các vị trí cắt sai) ngay trong khi nhận dạng.

Một số đề xuất cho hướng phát triển tiếp theo

Để hướng đến mục tiêu ứng dụng cho tiếng Việt, trong thời gian tới, chúng tôi sẽ tiếp tục tập trung vào các hướng nghiên cứu sau đây:

- Bổ sung thêm các tiêu chuẩn xác định vị trí cắt.
- Thu thập tập cơ sở dữ liệu mẫu chữ viết tay tiếng Việt.
- Bổ sung thêm một số lớp mạng để nhận dạng các cặp từ đôi trong tiếng Việt.
- Bổ sung thêm phần phân tích cú pháp và ngữ nghĩa tiếng Việt vào trong qúa trình nhận dạng để việc hiệu chỉnh và khử nhập nhằng chính xác hơn đồng thời để kết quả nhận dạng không bị phụ thuộc nhiều vào dữ liệu thống kê đầu vào.

TÀI LIỆU THAM KHẢO

[1] Alessandro Vinciarelli., A survey on off-line cursive word recognition, *Pattern Recognition* **35** (2002) 1433–1446.

[2] Andreas Schlapbach, Horst Bunke, Off-line handwriting identification using HMM based recognizers, *ICPR* (2) (2004) 654–658.

[3] G. Nicchiotti, S. Rimassa, C. Scagliola, A Simple and Effective Cursive Word Segmentation Method, *7th IWFHR* (2000).

[4] V. Lavrenko, T. Rath, and R. Manmatha, Holistic word recognition for handwritten historical documents, *Proceedings of Document Image Analysis for Libraries (DIAL)* (2004) 278–287.

[5] M. Maragoudakis, E. Kavallieratou, N. Fakotakis, Improving handwritten character segmentation by incorporating Bayesian knowledge with support vector machines, *IEEE Proc. of Int. Conf. Audio Speech & Signal Processing*, Orlando-Florida, (2002).

[6] M. E. Morita, L. S. Oliveira, and R. Sabourin, Unsupervised feature selection for ensemble of classifiers, *9th International Workshop on Frontiers in Handwriting Recognition (IWFHR-9)*, Kokubunji, Tokyo, Japan, October, 2004 (81–86).

[7] Nguyễn Thị Thanh Tân, Ngô Quốc Tạo, Một cấu trúc mạng nơron thích hợp cho việc nhận dạng chữ số viết tay, Nghiên cứu cơ bản và ứng dụng công nghệ thông tin (FAIR), Hà Nội (2003).

[8] Nguyễn Thị Thanh Tân, Ngô Quốc Tạo, Một phương pháp giải quyết vấn đề dính chữ trong nhận dạng chữ viết tay hạn chế, *Hội thảo Quốc Gia một số vấn đề chọn lọc của CNTT*, Đà Nẵng (2004).

[9] L. S. Oliveira, M. Morita, and R. Sabourin, Feature selection for ensembles applied to handwriting recognition, *International Journal on Document Analysis and Recognition* (2006).

[10] Radtke, V. W. Paulo, Sabourin, Robert, Wong, Tony, Intelligent feature extraction for ensemble of classifiers, *8th International Conference on Document Analysis and Recognition* (2005).

[11] Simone Marinai, Marco Gori, Giovanni Soda, Artificial neural networks for document analysis and recognition, *IEEE Trans. Pattern Anal. Mach. Intell* (2005) 23–35.

[12] S. Sethu Selvi and K. Indira, A novel character segmentation algorithm for offline handwritten character recognition, *Proc. of Int. Conf on Cognition and Recognition PES College of Engineering Mandya* (2005).

[13] U. Marti and H. Bunke, The IAM-database: an English sentence database for off-line handwriting recognition, *Journal of Document Analysis and Recognition* (2002).

[14] J. Wang, P. Neskovic, and L. N. Cooper, A probabilistic model for cursive handwriting recognition using spatial context, *ICASSP* (2005).

[15] http://hwr.nici.kun.nl/

[16] http://www.webopedia.com/TERM/H/handwriting_recognition.html

Nhận bài ngày 29 - 7 - 2005
Nhận lại sau sửa ngày 20 - 4 - 2006