



NHẬN DẠNG ĐIỂM VIẾT TAY TRÊN BẢNG ĐIỂM VỚI BIẾN ĐỔI HOUGH VÀ ĐẶC TRƯNG GIST

Lê Thanh Trúc¹ và Phạm Nguyên Khang²

¹ Khoa Kỹ thuật Công nghệ, Trường Đại học Tây Đô

² Khoa Công nghệ Thông tin & Truyền thông, Trường Đại học Cần Thơ

Thông tin chung:

Ngày nhận: 19/09/2015

Ngày chấp nhận: 10/10/2015

Title:

Handwritten digits recognition for school report cards using Hough transformation and Gist descriptor

Từ khóa:

Điểm viết tay, phát hiện bảng, phát hiện các ô, nhận dạng biểu mẫu, nhận dạng điểm viết tay

Keywords:

Handwriting scores, table detection, cells detection, form recognition, Handwriting scores recognition

ABSTRACT

Automatic form recognition is a problem attracting attention and investment of many researchers around the world. It plays an important role in data inputting and processing in practice. The study, "Handwriting scores recognition on transcripts" is derived from the actual need of Department of Academic Affairs and the other faculties at Tay Do University. This research is carried out through many phases such as noise reducing, de-skewing, rebuilding the table, locating columns in table, and recognizing handwriting by using Gist and SVM. Methodology for researching is to combine theory and practice for commenting and appraising the result. The research has given some highlights: a) positioning and building tables can be implemented without using any learning machines, without knowing about the logical position of the table in the form; b) locating any column as well as cell in the table without marking or color accents on the form is easily; c) the cost for form handling is low. The "Handwriting scores recognition on transcripts" system is developed successfully from the research. It allows loading the transcript from the list of transcripts image, recognizing and exporting the result to any file with the accuracy up to 97.30% on 187 transcripts.

TÓM TẮT

Nhận dạng biểu mẫu tự động đang là bài toán thu hút nhiều sự chú ý và đầu tư nghiên cứu của nhiều người trên thế giới. Sự cần thiết của nhận dạng biểu mẫu đóng vai trò quan trọng trong nhập và xử lý dữ liệu tự động cũng như hiệu quả và lợi ích nó mang lại trong thực tế. Nghiên cứu "Nhận dạng điểm số viết tay trên bảng điểm" được xuất phát từ nhu cầu thực tế của Phòng Đào tạo cũng như các khoa tại Trường Đại học Tây Đô. Bài toán này được nghiên cứu thực hiện qua nhiều công đoạn như khử nhiễu, khử nghiêng, xây dựng lại bảng, định vị cột và cuối cùng là nhận dạng điểm số viết tay trong cột định vị bằng cách sử dụng GIST và SVM. Để giải quyết bài toán này, nghiên cứu thực hiện phương pháp kết hợp lý thuyết và thực nghiệm, sau đó nhận xét và đánh giá chương trình. Phương pháp nghiên cứu đưa ra có một số điểm nổi bật như sau: thứ nhất là định vị và xây dựng bảng không cần dùng bất kỳ máy học nào cũng như không cần biết trước vị trí logic của bảng trong biểu mẫu; thứ hai là dễ dàng định vị cột cũng như ô bất kỳ trong bảng mà không cần đánh dấu hay điểm nhấn màu sắc nào trên biểu mẫu; cuối cùng là chi phí xử lý biểu mẫu là thấp. Sản phẩm cuối cùng nghiên cứu đạt được là xây dựng thành công phần mềm "Nhận dạng điểm số viết tay trên bảng điểm" với phiên bản 1.0. Phần mềm này cho phép tải bảng điểm, sau đó, nhận dạng điểm và xuất kết quả nhận dạng sang file bất kỳ với độ chính xác lên đến 97,30% trên 187 bảng điểm.

1 GIỚI THIỆU

1.1 Tình hình nhận dạng biểu mẫu và chữ viết tay

Nhận dạng chữ là lĩnh vực được quan tâm nghiên cứu và ứng dụng từ nhiều năm nay. Nhận dạng chữ gồm nhận dạng chữ in và nhận dạng chữ viết tay. Nhận dạng chữ in thường được ứng dụng phục vụ công việc tự động hóa đọc tài liệu, tăng tốc độ và hiệu quả nhập thông tin vào máy tính trực tiếp. Nhận dạng chữ viết tay thường phục vụ cho các ứng dụng đọc và xử lý chứng từ, hóa đơn, phiếu ghi, bản thảo viết tay,...

Đến thời điểm này, bài toán nhận dạng chữ in đã được giải quyết gần như trọn vẹn như sản phẩm FineReader 9.0 của hãng ABBYY có thể nhận dạng 20 ngôn ngữ khác nhau với độ chính xác hơn 99%. Phần mềm nhận dạng chữ Việt in VnDOCR 4.0 của Viện Công nghệ Thông tin Hà Nội có thể nhận dạng được các tài liệu chứa hình ảnh, bảng và văn bản tiếng Việt với độ chính xác trên 98%. Tuy nhiên, bài toán nhận dạng chữ viết tay vẫn còn là vấn đề thách thức lớn đối với các nhà nghiên cứu. Bài toán này chưa thể giải quyết trọn vẹn vì nó phụ thuộc quá nhiều vào nét chữ, sự biến đổi quá đa dạng trong cách viết và trạng thái của người viết. Bên cạnh đó, bài toán nhận dạng chữ viết tay hạn chế với các kết quả chủ yếu chỉ tập trung trên các tập dữ liệu chữ số viết tay chuẩn như USPS và MNIST.

Nhận dạng biểu mẫu bảng được nhiều nhà nghiên cứu trong và ngoài nước quan tâm nghiên cứu vì tính ứng dụng rộng rãi ngoài thực tiễn của nó. Hiện nay, trên thế giới có nhiều công trình nghiên cứu nhận dạng biểu mẫu một cách tự động. Ở Việt Nam, với nghiên cứu “Phân tích cấu trúc và nhận dạng biểu mẫu” do Đỗ Thanh Cao - Đại học Cần Thơ thực hiện năm 2011, nghiên cứu này nhắm đến nhận dạng các biểu mẫu trong ngành ngân hàng. Tuy nhiên, nghiên cứu không thực hiện trên biểu mẫu bảng. Công trình “Áp dụng phương pháp T-Recs vào nhận dạng bảng” do Tô Văn Khánh và Ngô Quốc Tạo nghiên cứu được trình bày ở Hội nghị khoa học Viện Công nghệ Thông Tin – 12/ 2006 và công trình “Phát hiện cấu trúc bảng trong nhận dạng văn bản” do tác giả Bùi Thị Thi – Đại học Công Nghệ Thông Tin và Truyền Thông Thái Nguyên thực hiện năm 2012, hai công trình này thực hiện nhận dạng trên biểu mẫu bảng với phương pháp T-Recs và T-Recs++. Bên cạnh đó, công trình “Nghiên cứu ứng dụng đường cong tham số B-SPLINE vào nhận dạng chữ số viết tay” do tác giả Cao Bá Thành – Đại học Đà Nẵng thực

hiện vào năm 2012. Công trình này sử dụng đường cong B-Spline vào nhận dạng ký tự in và ký tự viết tay trên bảng điểm ở Trung học Phổ thông.

1.2 Giới thiệu bài toán nghiên cứu

Sau khi sinh viên thi, bài thi được các giảng viên nhận về chấm và lên điểm bằng tay vào bảng điểm (Hình 1). Nghiên cứu cần xây dựng chương trình nhận dạng điểm viết tay trên cột điểm tổng cộng và xuất kết quả sang file bất kỳ (*.csv, *.xls, ...) theo cấu trúc mỗi dòng chứa <điểm>.

Với bài toán trên, nghiên cứu đề ra hai mục tiêu chính. Thứ nhất là định vị chính xác các ô điểm số viết tay trên cột điểm tổng cộng. Thứ hai là nhận dạng chính xác điểm số viết tay trong các ô đã được định vị với độ chính xác 80% trở lên trên mỗi bảng điểm.

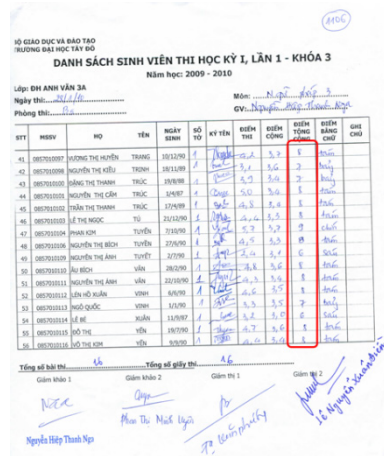
1.3 Đối tượng và phạm vi nghiên cứu

Nghiên cứu được thực hiện trên đối tượng là các file ảnh bảng điểm được thu nhận qua máy scanner là các file .pdf đã có sẵn, ảnh đầu vào có kích thước 5100 x 7017.

Phạm vi các điểm nghiên cứu nhận dạng gồm có điểm 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 là các số tròn, không chứa số lẻ. Ngoài ra, trường hợp sinh viên vắng thi thì ô điểm sẽ nhận giá trị là “v”.

1.4 Công tác chuẩn bị

Nghiên cứu thu thập 603 ảnh bảng điểm và tiến hành thống kê theo giảng viên, sau đó chia các bảng điểm theo mỗi giảng viên làm ba phần. Hai phần nghiên cứu thực hiện kiểm tra định vị ô và lấy điểm số viết tay cho máy học, nhóm này được gọi là *nhóm bảng điểm huấn luyện* (416 ảnh). Phần còn lại nghiên cứu để dành cho việc nhận dạng được gọi là *nhóm bảng điểm nhận dạng* (187 ảnh).



Hình 1: Bảng điểm

2 PHƯƠNG PHÁP NGHIÊN CỨU

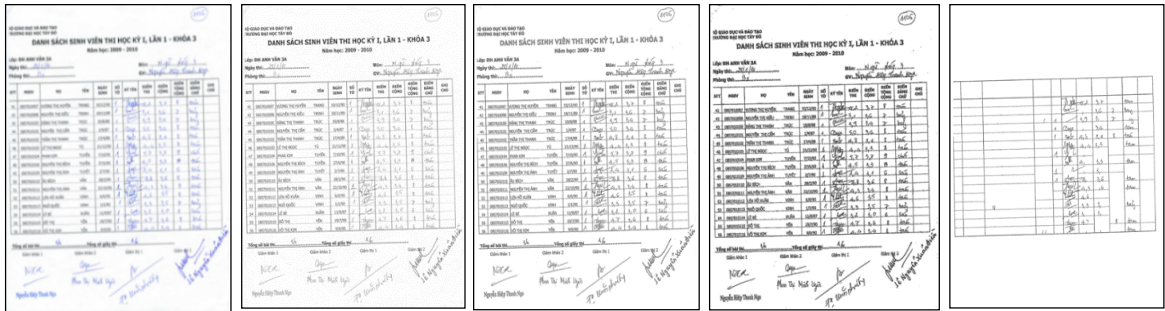
2.1 Định vị và lọc vùng băng

Để nhận dạng điểm viết tay trên cột điểm, nghiên cứu cần phải định vị các ô ở cột điểm tổng cộng. Để định vị các ô, nghiên cứu cần tiếp cận vùng băng. Vùng băng là vùng rộng nhất trên ảnh. Trước khi định vị vùng băng, ảnh băng điểm được tiền xử lý qua các bước:

- Làm mịn ảnh: Ảnh được tải ở dạng mức xám, sau đó được làm mịn bằng bộ lọc Gaussian.
- Nhị phân ảnh: Sau khi ảnh được làm mịn,

nghiên cứu đưa ảnh về ảnh nhị phân bằng cách dùng ngưỡng thích nghi. Nếu giá trị tại điểm ảnh $(i,j) >$ giá trị ngưỡng thì (i,j) có màu trắng, ngược lại có màu đen. Với giá trị ngưỡng là giá trị trung bình của vùng 3×3 với tâm là điểm (i,j) đang xét.

- Làm nổi các đối tượng ảnh và giãn ảnh: dùng phép toán âm bản nhằm làm nổi các đối tượng như các đường thẳng, văn bản,... Phép toán giãn ảnh nhằm làm giãn nở các đường nét bị mất ở bước nhị phân hóa, hai phép toán này được thực hiện nhằm giúp việc phát hiện vùng băng dễ dàng hơn.



a) Ảnh gốc

b) Ảnh sau khi làm mịn

c) Ảnh nhị phân (ảnh gốc)

d) Ảnh sau khi làm nổi và giãn ảnh (0: nền; 1: đối tượng)

e) Ảnh sau khi lọc vùng băng (0: nền; 1: đối tượng)

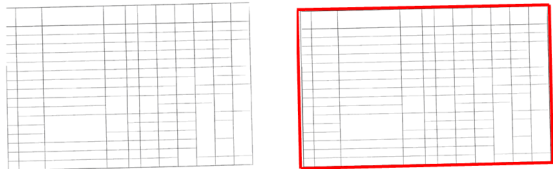
Hình 2: Các ảnh minh họa tiền xử lý và lọc vùng băng

Sau khi thực hiện các bước tiền xử lý, nghiên cứu sử dụng kỹ thuật floodFill. Hàm floodFill bắt đầu tìm kiếm từ điểm (x, y) có màu trắng, điểm (x,y) được tô màu Fcolor và các điểm láng giềng của nó cũng được tô cùng màu Fcolor, vùng (x, y) là vùng lớn nhất có diện tích S_i . Nếu vùng có điểm bắt đầu (m, n) có diện tích lớn hơn S_i thì khi đó cập nhật lại vùng có điểm bắt đầu là điểm (m, n) là vùng lớn nhất. Sau khi tìm được vùng lớn nhất, nghiên cứu tiến hành tô màu lại cho ảnh. Vùng lớn nhất được tô phủ màu trắng và các vùng còn lại được phủ màu đen. Bước này được gọi là lọc vùng lớn nhất. Vùng lớn nhất nghiên cứu nhận được là vùng băng (Hình 2e).

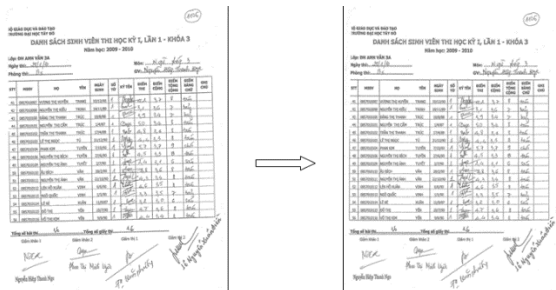
2.2 Phát hiện góc nghiêng và xoay ảnh

Để phát hiện góc nghiêng, nghiên cứu dùng biến đổi Hough nâng cao và nhận về danh sách các đoạn thẳng. Nghiên cứu vẽ lại các đoạn thẳng này, sau đó, tìm hình chữ nhật bé nhất bao quanh các đoạn thẳng. Hình chữ nhật này gọi là hình chữ nhật xoay (rotate rect), mỗi hình chữ nhật xoay có 4 thông số là tâm, độ rộng, chiều cao và góc nghiêng θ so với trục Ox (Hình 3). Góc nghiêng của bảng điểm là góc của hình chữ nhật xoay. Nghiên cứu tiến hành xoay ảnh gốc theo tâm quay

là điểm trung tâm của ảnh gốc và góc xoay là góc θ (Hình 4).



Hình 3: Hình vẽ lại các đoạn thẳng (bên trái) và hình chữ nhật xoay bao quanh (bên phải)



Hình 4: Ảnh gốc (ảnh bên trái) và ảnh gốc sau khi xoay (ảnh bên phải)

2.3 Xây dựng bảng

Nghiên cứu xoay các đoạn thẳng ở bước 2.2 theo góc θ với tâm quay là tâm của hình chữ nhật

xoay. Sau đó, tiến hành xây dựng lại bảng bằng cách vẽ lại các đoạn thẳng. Giải thuật xây dựng bảng như sau:

Đầu vào: danh sách các đoạn thẳng.

Đầu ra: ảnh grid như Hình 5.

Function mergeLine (danh sách các đoạn thẳng l){

For (i = 0; i < tổng số đoạn thẳng; i++)

 Tìm $x_{min}, x_{max}, y_{min}, y_{max}$ của l;

For (i = 0; i < tổng số đoạn thẳng - 1; i++) {

 If (i đã duyệt rồi) continue;

 If (i nằm ngang) {

 for (j = i+1; j < tổng số đoạn thẳng;

j++)

 If (j nằm ngang && cùng dòng với

i){

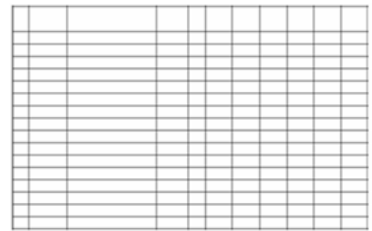
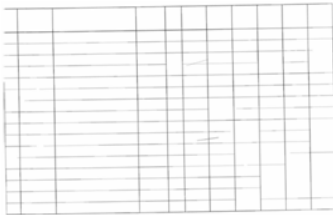
 Gom i, j lại thành 1 đoạn

thẳng;

 Đánh dấu j đã duyệt rồi;

 }

 Vẽ đoạn thẳng (x_{min}, y_1) đến (x_{max}, y_2);



Hình 5: Ảnh đoạn thẳng vẽ lại (trái); Ảnh grid sau khi xây dựng bảng (giữa); Ảnh rõ ảnh grid (phải)

2.4 Định vị ô

Các ô chứa điểm cần nhận dạng cùng nằm trong một cột, đều có dạng hình chữ nhật. Vì vậy, để định vị các ô này nghiên cứu đưa về bài toán phát hiện các hình chữ nhật trong bảng. Mỗi hình chữ nhật được phát hiện có 3 thuộc tính là tọa độ điểm góc trên bên trái (x, y), chiều cao và chiều

```

}
If (i thẳng đứng) {
for (j = i+1; j < tổng số đoạn thẳng; j++)
If (j thẳng đứng && cùng dòng với i) {
Gom i, j lại thành 1 đoạn thẳng;
Đánh dấu j đã duyệt rồi;
}
}
If (đoạn thẳng đứng không là trái nhất)
Vẽ đoạn thẳng ( $x_1, y_{min}$ ) đến ( $x_2, y_{max}$ );
}
}
Vẽ đoạn thẳng ( $x_{min}, y_{min}$ ) đến ( $x_{min}, y_{max}$ );
}
- Với i, j: là đoạn thẳng
- i, j cùng dòng khi  $abs(x_i - x_j) \leq 80$  và  $abs(y_i - y_j) \leq 80$ .
- i, j được gom lại thành một đoạn thẳng có  $x = (x_i + x_j)/2$ ,  $y = (y_i + y_j)/2$ .
- Đoạn thẳng đứng trái nhất là đoạn thẳng có  $abs(x_{min} - x_1) \leq 40$ .
    
```

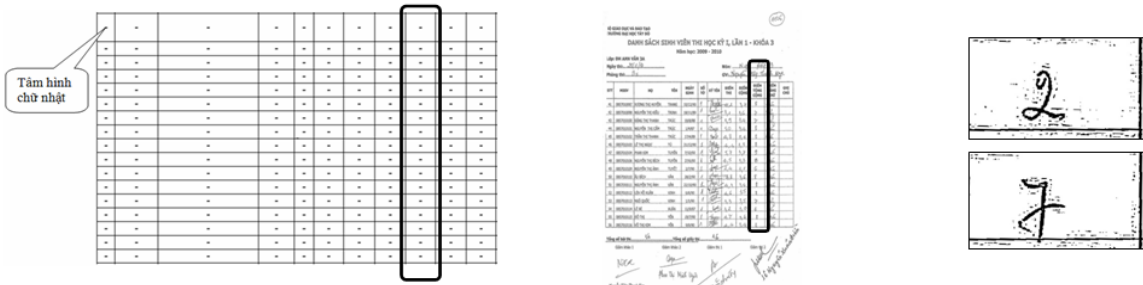
rộng. Nghiên cứu dùng mảng lưu tọa độ x (khoảng 20 giá trị x), sau đó, sắp xếp x tăng dần và loại bỏ phần tử trùng (Bảng 1, Bảng 2). Kết quả nhận được là mảng tăng dần theo x tương ứng với các hình chữ nhật trên các cột của bảng điểm. Nghiên cứu tiến hành cắt các hình chữ nhật theo vị trí cột cần tìm và thực hiện cắt trên bảng điểm gốc sau khi xoay (Hình 6).

Bảng 1: Mảng lưu trữ tọa độ x của điểm góc trên bên trái các hình chữ nhật

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
150	4452	4098	3745	3386	3027	2675	2445	2060	858	353	150	4452	4098	3745	3386	3027	2675	2445	2060

Bảng 2: Mảng sau khi sắp xếp tăng dần và xóa phần tử trùng

0	1	2	3	4	5	6	7	8	9	10
150	353	858	2060	2445	2675	3027	3386	3745	4098	4452



Hình 6: Ảnh sau khi phát hiện hình chữ nhật (trái); Ảnh gốc sau khi xoay (giữa); Ảnh các ô điểm sau khi cắt (phải)

2.5 Nhận dạng điểm số viết tay

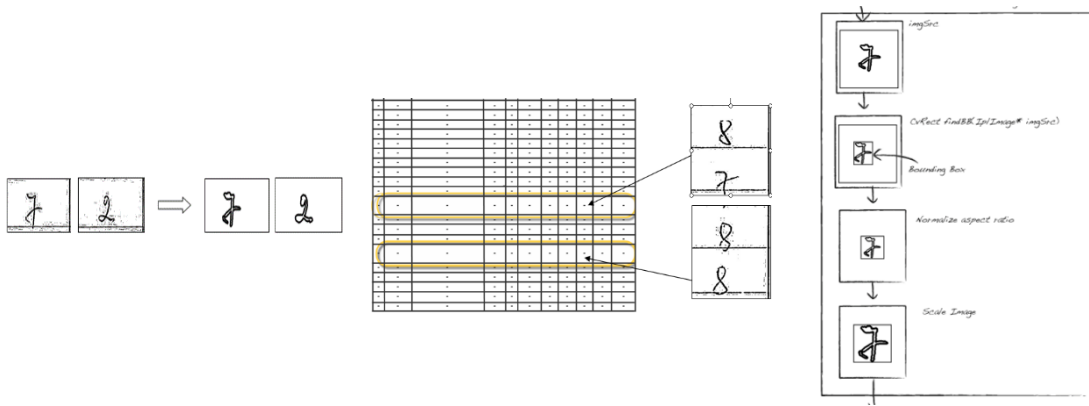
Để nhận dạng điểm số viết tay nghiên cứu thực hiện qua 3 giai đoạn là tiền xử lý các ô điểm, trích xuất đặc trưng với GIST và phân lớp với máy học SVM.

a. Tiền xử lý các ô điểm

Các ô sau khi được cắt ra từ bảng điểm chứa nhiều nhiễu như đường thẳng đứng, đường thẳng ngang, nhiễu đốm lớn, nét điểm của ô bên trên viết lẫn xuống ô bên dưới,... Nghiên cứu tiến hành tiền xử lý xóa các nhiễu đó (Hình 7a). Ngoài ra, do áp dụng biến đổi Hough phụ thuộc vào việc chọn ngưỡng M quy định chiều dài đoạn thẳng được phát hiện, với M bằng 20 nghiên cứu làm mất khá

nhều đoạn thẳng nên có dòng trong bảng nghiên cứu không phát hiện được bất cứ đoạn thẳng nào. Khi đó, một hình chữ nhật được phát hiện sẽ bao gồm nhiều hình chữ nhật thật (Hình 7b). Để giải quyết trường hợp này, nghiên cứu đề ra thuật toán chia một hình chữ nhật thành các hình chữ nhật con nếu nó có chiều cao lớn hơn chiều cao trung bình các hình chữ nhật.

Sau khi xóa các nhiễu nghiên cứu tiến hành chuẩn hóa kích thước các ảnh. Việc chuẩn hóa kích thước được tiến hành qua các bước gồm tìm hình chữ nhật bé nhất chứa ảnh, cân bằng kích thước chiều rộng và chiều cao ảnh và sau cùng là chuẩn hóa ảnh về kích thước 64 x 64 (Hình 7c).



a) Minh họa xóa nhiễu b) Minh họa cắt hình chữ nhật lớn c) Ảnh minh họa chuẩn hóa kích thước ảnh

Hình 7: Ảnh minh họa xóa nhiễu và cắt hình chữ nhật lớn

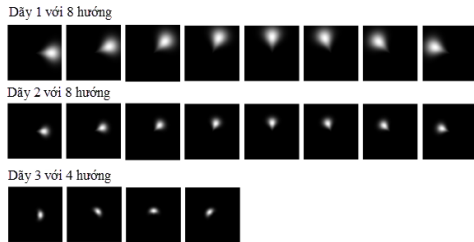
b. Trích xuất đặc trưng GIST

Đặc trưng gist do Aude Oliver đề xuất vào năm 1999. Gist thường được sử dụng trong ứng dụng phân tích, tìm kiếm ảnh, nhận dạng mặt người, phân loại kết cấu và phân loại biểu hiện trên khuôn mặt,... Ảnh sau khi được tiền xử lý sẽ được rút

trích đặc trưng. Quá trình rút trích đặc trưng gist được thực hiện với các bước sau đây.

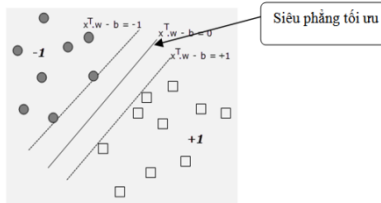
- Đầu tiên, ảnh đầu vào là ảnh mức xám được chuẩn hóa là ảnh vuông và có kích thước là lũy thừa của 2.
- Tiếp theo, ảnh được tách ra thành 3 ảnh tương ứng với 3 kênh màu R, G, B.

– Áp dụng biến đổi Fourier cho mỗi ảnh R, G, B. Các ảnh sau khi được biến đổi được đi qua ba dãy của 20 bộ lọc Gabor, mỗi dãy lại có tần số khác nhau. Dãy thứ nhất và dãy thứ hai mỗi dãy sử dụng 8 bộ lọc 8 hướng, dãy thứ 3 sử dụng 4 bộ lọc 4 hướng. Như vậy, với mỗi ảnh R, G, B qua 20 bộ lọc Gabor có tổng cộng 60 ảnh (Hình 8).

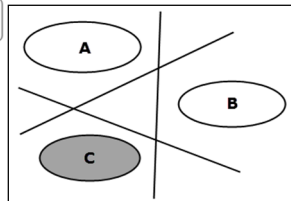


Hình 8: Hình minh họa ảnh qua 20 bộ lọc Gabor

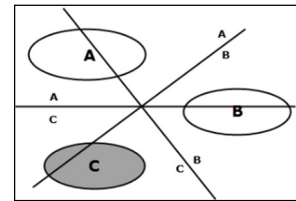
– Cuối cùng, áp dụng phép biến đổi Fourier ngược trên mỗi ảnh đi qua bộ lọc Gabor. Mỗi ảnh khi qua 20 bộ lọc Gabor sẽ cho 20 ảnh tương ứng với 20 bộ lọc. Các ảnh này được chia thành $n \times n$ vùng con bằng nhau. Mỗi vùng tương ứng với một đặc trưng. Trong chương trình nhận dạng bảng điểm, nghiên cứu chọn $n = 4$, như vậy, mỗi ảnh sẽ được chia thành 16 vùng ảnh con. Tổng số đặc trưng nhận được là $16 \times 60 = 960$ chiều. Đặc trưng trên mỗi vùng con bằng tổng số điểm ảnh trên một



a) Phân lớp SVM hai lớp



b) SVM đa lớp với 1- tất cả



c) SVM đa lớp với 1- 1

Hình 10: Hình minh họa siêu phẳng tối ưu SVM

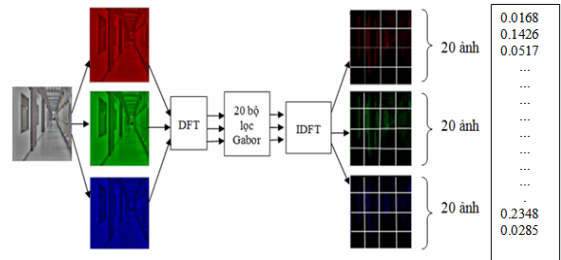
Mô hình máy học SVM được xây dựng trên nền tảng lý thuyết học thống kê, cho kết quả rất tốt trong thực tế, mô hình có độ chính xác cao, chịu đựng nhiễu, hiệu quả khi xử lý dữ liệu có số thuộc tính lớn. Chỉ cần thay đổi hàm nhân dẫn đến các mô hình mới. Máy học SVM thành công hầu hết trong các lĩnh vực nhận dạng các ký tự số viết tay, phân lớp dữ liệu văn bản, lọc thư rác, phân lớp dữ liệu gen,...

3 KẾT QUẢ VÀ THẢO LUẬN

3.1 Định vị ô

Nghiên cứu tiến hành chạy chương trình kiểm tra tính đúng đắn của chương trình trên nhóm bảng điểm huấn luyện. Kết quả cho thấy việc định vị là

vùng chia cho $n \times n$ (Hình 9).



Hình 9: Hình minh họa các bước rút trích đặc trưng GIST

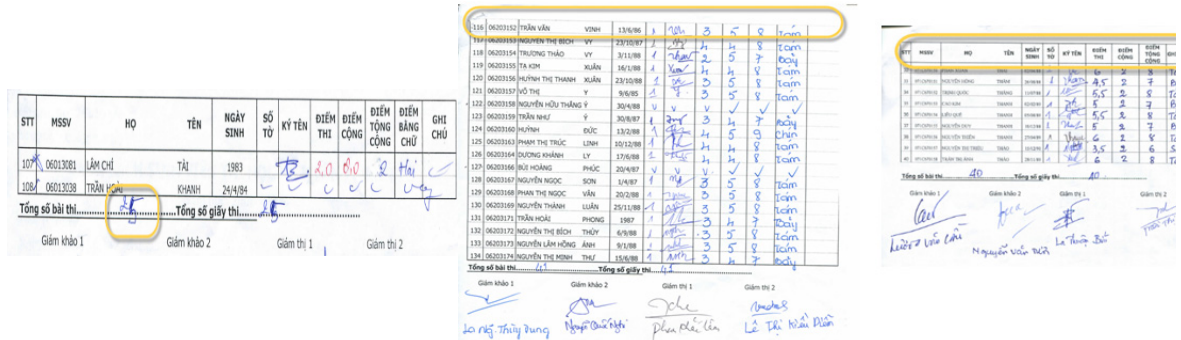
c. Phân lớp

Phân lớp dữ liệu là một nhiệm vụ chung trong máy học. Trong mô hình học có giám sát, thuật toán được cho trước một số điểm dữ liệu cùng với nhãn của chúng thuộc một trong hai lớp cho trước. Mục tiêu của thuật toán là xác định xem một điểm dữ liệu mới sẽ được thuộc về lớp nào và ta muốn biết liệu có thể chia tách hai lớp dữ liệu bằng một siêu phẳng (hyperplane) tối ưu hay không? Siêu phẳng tối ưu phải là siêu phẳng tách 2 lớp xa nhất có thể. Đây là giải thuật học SVM.

Đối với bài toán nhiều lớp chúng ta có thể đưa về bài toán 2 lớp bằng kỹ thuật 1 – tất cả (one vs all) hoặc 1 – 1 (one vs one).

hoàn toàn chính xác 100%. Mỗi bảng điểm có số sinh viên là x thì nghiên cứu nhận được đúng x ô điểm có thứ tự đúng như trên bảng điểm.

Trên nhóm bảng điểm nhận dạng, nghiên cứu nhận thấy rằng có 2/187 ảnh bảng điểm thiếu ô và 4/187 ảnh thừa ô. Nguyên nhân dẫn đến thừa ô là do nghiên cứu lọc vùng lớn nhất bằng kỹ thuật floodFill nên những bảng điểm nào có vùng lân cận vùng bảng chứa đường thẳng liên thông với vùng bảng thì sẽ tạo thêm 1 dòng. Điều này dẫn đến thừa ô (Hình 11a). Tương tự như thừa ô, các bảng điểm thiếu ô thiếu đúng một ô là do dòng đầu của bảng không có dòng tiêu đề (các bảng điểm gồm 2 trang, trường hợp nhắc đến là bảng ở trang 2 - Hình 11b).



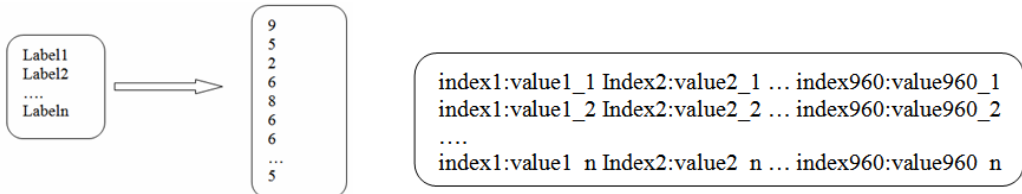
a) Minh họa bảng điểm bị thừa ô b) Minh họa bảng điểm bị thiếu ô c) Minh họa bảng điểm không thừa/ thiếu ô

Hình 11: Các bảng điểm trong trường hợp thừa và thiếu ô

3.2 Nhận dạng điểm viết tay

Sau bước định vị ô trên *nhóm bảng điểm huấn luyện*, nghiên cứu nhận được 8319 ô chứa điểm (*ảnh ô điểm*), trong đó, điểm 0 có 19 ảnh, điểm 1 có 50 ảnh, điểm 2 có 112 ảnh, điểm 3 có 262 ảnh, điểm 4 có 414 ảnh, điểm 5 có 1061 ảnh, điểm 6 có 1503 ảnh, điểm 7 có 1790 ảnh, điểm 8 có 1621 ảnh, điểm 9 có 911 ảnh, điểm 10 có 181 ảnh và cuối cùng là “v” có 395 ảnh.

Dựa vào số lượng điểm theo lớp, nghiên cứu tiến hành lấy 2/3 trong tổng số ảnh trên mỗi lớp làm *tập huấn luyện* (5546 mẫu), 1/3 còn lại làm *tập kiểm tra* (2223 mẫu). Sau đó, nghiên cứu trộn ngẫu nhiên các ảnh trong *tập huấn luyện* và *tập kiểm tra*, ghi nhận lại các nhãn các ảnh trong mỗi tập ra filelabel_test.txt và label_train.txt. Cấu trúc file nhãn như Hình 12a.



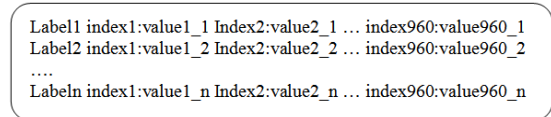
a) Cấu trúc file nhãn b) Cấu trúc file đặc trưng

Hình 12: Cấu trúc file nhãn và file đặc trưng

Nghiên cứu cũng tiến hành rút trích đặc trưng các ảnh trong *tập huấn luyện* và *tập kiểm tra* đã được trộn ngẫu nhiên, ghi nhận các đặc trưng ra file đặc trưng (feature_test.txt và feature_train.txt) với cấu trúc file như Hình12b. Cuối cùng, nghiên cứu ghép file nhãn và file đặc trưng thành *file huấn luyện (train.txt)* và *file kiểm tra(test.txt)* với cấu trúc như Hình 13:

là RBF, tham số $c = 100$ và $\gamma = 0.1$, hệ thống đạt độ chính xác là 97.11%.

Tiếp đến, trên nhóm *bảng điểm nhận dạng*, nghiên cứu cũng tiến hành rút trích đặc trưng các ô đã được định vị và ghi nhận vào *file đặc trưng nhận dạng* với cấu trúc như Hình 13, trong đó, các labeli được tạo ngẫu nhiên từ 0 đến 11. Với đầu vào là mô hình huấn luyện thu được ở bước huấn luyện, nghiên cứu trả về kết quả nhận dạng là file .xls có cấu trúc mỗi dòng là điểm, có độ chính xác là 97,30%.



Hình 13: Cấu trúc file train và file test

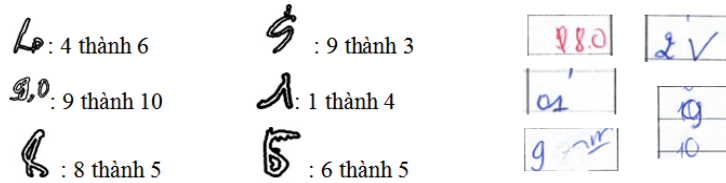
Nghiên cứu đưa file *train.txt* vào máy học và nhận được mô hình, nghiên cứu sử dụng mô hình này để kiểm tra độ chính xác trên file *test.txt*. Với các tham số gồm kiểu SVM là c-SVM, hàm nhân

Ngoài ra, nghiên cứu tiến hành thử nghiệm trên nhóm bảng điểm mới (nhóm bảng điểm của các giảng viên chưa có mẫu trong tập huấn luyện), nghiên cứu đạt kết quả rất khả quan với độ chính xác 86,34%.

Tóm lại, cách tiếp cận của nghiên cứu đạt độ chính xác 97,30% trên nhóm bảng điểm có mẫu

trong tập huấn luyện và đạt 86,34% trên nhóm bảng điểm mới. Kết quả nhận dạng bảng điểm chưa đạt 100% do nhiều nguyên nhân khách quan và chủ quan. Nguyên nhân khách quan do chất lượng ảnh bảng điểm kém, các điểm viết tay chứa nhiều

hiều, các điểm viết tay bị viết lẩn xuống các ô điểm khác. Nguyên nhân chủ quan là do chữ viết tay phụ thuộc nhiều vào nét chữ của người viết, cán bộ lên điểm chưa chuẩn (điểm 0 thì ghi là 00, chỉnh sửa điểm, ký tên nháy kể bên điểm,...) (Hình 14).



Hình 14: Ảnh minh họa kết quả sai điểm viết tay do nguyên nhân chủ quan và khách quan

3.3 Thời gian thực hiện

Thời gian thực hiện chương trình nhận dạng phụ thuộc khá nhiều vào chất lượng bảng điểm đầu vào, số lượng các ô điểm trên mỗi bảng điểm, cấu hình máy,... Nhìn chung, bảng điểm với 30 ô điểm, nghiên cứu nhận dạng mất khoảng 14 đến 17 giây. Thời gian thực hiện chương trình tính từ lúc tải một ảnh bảng điểm cho đến khi định vị và cắt các ô

điểm trong cột điểm được trình bày ở Bảng 3. Nghiên cứu đo thời gian thực hiện trên *nhóm bảng điểm huấn luyện*, kết quả thời gian trung bình thực hiện định vị ô một ảnh *bảng điểm huấn luyện* là 13.40 giây. Thời gian trung bình thực hiện một ảnh *bảng điểm nhận dạng* tính từ thời điểm tải ảnh bảng điểm đến lúc trả về nhãn dự đoán là 14.51 giây.

Bảng 3: Bảng thống kê thời gian thực hiện trên một bảng điểm

STT	Các bước xử lý	Chi tiết các bước xử lý	Thời gian thực hiện (giây)	Tổng cộng (giây)
1.	Tải ảnh		0.22	0.22
2.	Tiền xử lý	Làm mịn ảnh	0.17	0.6
		Chuyển ảnh về ảnh nhị phân	0.27	
		Làm nổi các đối tượng	0.04	
		Lưu ảnh gốc	0.06	
		Giãn ảnh	0.06	
3.	Định vị vùng bảng	Tìm vùng lớn nhất	2.28	4.61
		Lọc vùng lớn nhất	2.28	
		Cơ ảnh	0.05	
4.	Phát hiện và hiệu chỉnh góc nghiêng	Dùng biến đổi Hough để phát hiện đường thẳng	4.6	7.6
		Vẽ grid từ Hough để phát hiện góc nghiêng	0.08	
		Tính góc nghiêng	1.02	
		Giảm góc nghiêng	1.9 + 1.87 (góc + grid)	
5.	Xây dựng bảng	Vẽ lại grid	0.54	0.54
6.	Định vị ô	Phát hiện hình chữ nhật	0.09	0.11
		Cắt các ô điểm trong cột điểm tổng cộng	0.01	
		Hậu xử lý các ô điểm	0.01	
Tổng cộng			13.68	13.68

4 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Qua thời gian dài thực hiện, nghiên cứu đã xây dựng thành công chương trình nhận dạng bảng điểm đáp ứng được nhu cầu thực tiễn tại Trường Đại học Tây Đô. Chương trình bao gồm định vị ô và nhận dạng điểm viết tay. Về định vị ô, chương trình này có thể định vị ô khá tổng quát và có thể định dạng các ô trong cột theo nhu cầu của người sử dụng và đạt độ chính xác 99,9%. Về nhận dạng

điểm viết tay, chương trình đạt được độ chính xác đến 97.30%.

Về hướng phát triển, chương trình nhận dạng bảng điểm có thể kết hợp nhận dạng điểm viết tay với nhận dạng chữ in gồm mã số sinh viên, mã lớp, mã môn học làm nền tảng cho việc nhận dạng tự động bảng điểm. Ngoài ra, chương trình cần mở rộng nhận dạng nhiều cột cũng như nhận dạng biểu mẫu khác, đây là hướng cần đạt đến của nghiên cứu.

LỜI CẢM ƠN

Xin chân thành cảm ơn sự hỗ trợ kinh phí của Trường Đại học Tây Đô thông qua đề tài cấp trường. Chân thành cảm ơn Thầy Lý Mẫn Tệp (Trường Đại học Tây Đô) đã cung cấp bảng điểm. Chân thành cảm ơn Thầy Trịnh Huệ, Thầy Nguyễn Chí Thắng (Khoa Kỹ thuật Công nghệ - Trường Đại học Tây Đô) đã tạo điều kiện về mọi mặt. Cuối cùng, chân thành cảm ơn sự hỗ trợ anh Nguyễn Việt Vinh.

TÀI LIỆU THAM KHẢO

1. Đỗ Thanh Nghị: *Khai mở dữ liệu – Minh họa bằng ngôn ngữ R*, NXB Đại học Cần Thơ, trang 44-52 (2011).
2. Phạm Anh Phương, *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, (2008), 109-118.
3. Cao Bá Thành: *Nghiên cứu ứng dụng đường cong tham số B-SPLINE vào nhận dạng chữ số viết tay*. Luận văn thạc sĩ Trường Đại học Đà Nẵng (2012).
4. António R.: *Image representation, sampling and quantization*, ECE 6962-Fall (2008).
5. Bradski G. and Kaehler A.: *Learning OpenCV*. Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA95472, (2008).
6. Cesarini S., Maraini S., Sardi L. and Sorda G.: *Trainable table location in document images*, Proc. Intl. Conf. Patt. Recog., pp. 236-240 (2002).
7. Chaudhuri D., Samal A.: *A simple method for fitting of bounding rectangle to close regions*, Pattern Recognition 40, pp. 1981–1989 (2007).
8. Chih-Chung C. and Chil-Jen L., *LIBSVM: a Library for Support Vector Machines*, NationalTaiwanUniversity (2004).
9. Cristianini N. and Shawe-Taylor J. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. CambridgeUniversity Press (2000).
10. Dennis S., John P.: *A Linear Time Algorithm for the Minimum Area Rectangle Enclosing a Convex Polygon*. Computer Science Technical Reports, page 382 (1983).
11. Gatos B., Danatsas D., Pratikakis I. and Perantonis S.: *Automatic table detection in document images*, Proc. Intl. Conf. Advances Patt. Recog., pp. 609-618 (2005).
12. Ghassan H., Karin A. and Rafeef . – Gharbieh: *Automatic Line Detection Using Hough Transform* (1999).
13. Illingworth J. and Kittler J.: *A survey of the Hough Transform*. Computer Vision, Graphics and Image Processing, 44: pp. 87–116 (1988).
14. Kieninger T. and Dengel A.: *Applying the T-Recs Table Recognition System to the Business Letter Domain*. Proc. of the 6th International Conference on Document Analysis & Recognition, Seattle, pp. 518-522, (2001).
15. Neves L. and Facon J.: *Methodology of Automatic extraction of Table-Form Cells*. IEEE Proc. of the XIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAP'00), pp. 15-21 (2000).
16. Olivier B.: *Digital Image Processing*, Prentice Hall. (3th Ed.). IMESI–2009.
17. Ramesh J., Rangachar K. and Brian S.: *Machine Vision*. Published by McGraw-Hill, Inc., ISBN 0-07-032018-7 (1995).
18. Ranjitha S.: *Enhancing the Autonomous Robotic Person Detection and Following Using Modified Hough Transforms* (2013).
19. Reza S. and Shahram K.: *Document skew detection using minimum area bounding rectangle*, Proc. Intl. Conf. Coding and Computer, pp. 253-258 (2000).
20. Richard O. and Peter E.: *Use of the Hough Transformation To Detect Lines and Curves in Pictures*. Communication of the ACM, 15, 1, January 1972, pp. 11-15.
21. Tai-Sin Lee, *Image representation using 2D Gabor wavelets*, IEEE Trans. Pattern Analysis and Machine Intelligence, VOL. 18, NO. 10 (1996).
22. Trucco and Jain: *Using Hough Transform to detect lines*. Chapter 5.
23. Wei-lun C.: *Gabor wavelet transform and its application*. Unpublished report.
24. Wu X. and Kumar V.: *Top 10 Algorithms in Data Mining*. Chapman & Hall/CRC (2009).
25. Zanibbi R., Blostein D. and Cordy J.: *A survey of table recognition*. International Journal of Document Analysis and Recognition (IJ DAR), vol. 7, pp. 1-16, (2004).
26. Zheng Y., Liu C., Ding X., Pan S.: *Form Frame Line Detection with Directional Single-Connected Chain*. Proc. of the 6th Int. Conf. on Doc. Anal. & Recognition, pp. 699-703 (2001).