 DOI: 10.35382/18594816.1.41.2020.646

GIẢI PHÁP NHẬP ĐIỂM DỰA VÀO ĐẶC TRƯNG GIST, KĨ THUẬT SVM VÀ TESSERACT

Nguyễn Hùng Hậu[1], Nguyễn Thái Sơn[2]

INPUTING STUDENTS' SCORE BASED ON GIST FEATURES, SUPPORT VECTOR MACHINES AND TESSERACT

Nguyen Hung Hau[1], Nguyen Thai Son[2]

Tóm tắt – *Việc quản lí điểm số trong các trường đại học và cao đẳng yêu cầu tính an toàn và tính chính xác cao bởi tính chất quan trọng của nó. Trong đó, việc nhập điểm số của học sinh và sinh viên vào hệ thống để đảm bảo công tác lưu trữ thường mất nhiều thời gian và công sức do khả năng xảy ra sai sót cao. Để giảm thiểu rủi ro và tăng tính chính xác, trong bài báo này, chúng tôi đề xuất giải pháp nhập điểm bằng kĩ thuật nhận dạng. Trong giải pháp này, chúng tôi sử dụng trích chọn đặc trưng GIST, kĩ thuật SVM và Tesseract trong nhập điểm viết tay cho Trường Cao đẳng nghề Sóc Trăng. Giải pháp đề xuất gồm hai công việc chính là nhận dạng vùng mã số học sinh, sinh viên dạng chữ in bằng Tesseract và nhận dạng vùng điểm số viết tay bằng mô hình máy học SVM với đặc trưng GIST. Trong phần kết quả thực nghiệm, giải pháp đề xuất đạt được độ chính xác cao, hơn 96% cho chữ in và hơn 93% cho điểm số viết tay. Thời gian nhận dạng trung bình cho một bảng điểm là 7,9 giây. Điểm nổi bật của trong nghiên cứu này là sự kết hợp nhận dạng chữ in và chữ số viết tay cho công tác nhập điểm số ứng dụng thực tế tại đơn vị.*

Từ khóa: GIST, MNIST, nhận dạng, OCR, SVM.

[1]Trường Cao đẳng Nghề Sóc Trăng
[2]Trường Đại học Trà Vinh
Ngày nhận bài: 17/7/2020; Ngày nhận kết quả bình duyệt: 14/10/2020; Ngày chấp nhận đăng: 10/12/2020
Email: nhhau@svc.edu.vn
[1]Soc Trang Vocational College
[2]Tra Vinh University
Received date: 17th July 2020; Revised date: 14th October 2020; Accepted date: 10th December 2020

Abstract – *Handwriting recogination plays an important role in data inputing and processing in the practice. This attracts much attention of many researchers in different fields. In this paper, a new algorithm is proposed by basing on GIST features, Support Vector Machines (SVM) and Tesseract for entering the score on students' transcript form at Soc Trang Vocational College. The algorithm consists of two main works, i.e., recognizing students'code and recogziing handwritten digit. In the proposed algorithm, all regions of interest are determined and extract their dictint features with using tesseract and GIST. Then, these features are classified by SVM mechanism. Experimental results demonstrated that the proposed algorithm obtained high performance with accuracy up to 96,57% for students' code and 93,55% for Handwritting scores. Average time was 7,9s per one transcript.*

Keywords: GIST, MNIST, OCR, recognition, SVM.

I. GIỚI THIỆU

Việc nhận dạng một lĩnh vực như giám sát giao thông, bãi giữ xe, bảng điểm, phiếu ghi hàng đang được giới khoa học quan tâm nhằm giải quyết các yêu cầu trong cuộc sống hiện nay. Trong đó, việc nhận dạng chữ viết tay hiện vẫn còn nhiều thách thức đối với những nhà nghiên cứu, bởi nó phụ thuộc vào con người, ngôn ngữ, trạng thái tâm lí khi viết. Trong khi đó, chữ viết tay xuất hiện rất nhiều trong các cơ quan, xí nghiệp. Nó được thể hiện trên các biểu mẫu. Trong trường học, công tác điểm số của người học cũng được thể

hiện dưới dạng chữ viết tay trên các biểu mẫu liên quan. Những thông tin về điểm số của người học dưới dạng viết tay cần đưa vào máy để xử lí. Phương pháp truyền thống đòi hỏi cần một quy trình nhập điểm khá phức tạp, tốn thời gian và thậm chí có thể dẫn đến sai sót. Việc áp dụng kĩ thuật nhận dạng trên bảng điểm viết tay để đưa thông tin vào máy là một trong những giải pháp hỗ trợ xử lí nhanh và hạn chế sai sót. Để làm được điều đó, nhiều phương pháp, kĩ thuật khác nhau được tiếp cận như logic mờ, giải thuật di truyền, mô hình xác suất thống kê, mô hình mạng nơ-ron nhân tạo.

Nhiều công trình nghiên cứu về nhận dạng chữ số viết tay đạt hiệu quả khả quan. Gaurav Jain, Jason Ko [1] nhận dạng chữ số viết tay, sử dụng giải thuật PCA kết hợp với 1-NN trên tập dữ liệu MNIST, kết quả cho tỉ lệ chính xác 78,4%. Đỗ Thanh Nghị, Phạm Nguyên Khang [2] đã áp dụng giải thuật máy học rừng ngẫu nhiên xiên phân (rODT) sử dụng các đặc trưng toàn cục GIST, kết quả thực nghiệm trên tập dữ liệu MNIST đạt độ chính xác 99,12%. Cả hai nghiên cứu thực hiện trên tập dữ liệu mẫu được cung cấp bởi tổ chức NIST (National Institute of Standards and Technology) và sau đó được LeCun cập nhật, chia thành hai tập riêng biệt gọi là MNIST [3]. Đây là tập dữ liệu chuẩn, chưa thể hiện được tính thực tế cao do trong thực tiễn tồn tại các dạng biểu mẫu viết tay rất đa dạng và phong phú.

Các hướng nghiên cứu gần đây của Lê Thanh Trúc [4], Võ Ngọc Lợi [5] đã dựa vào đặc trưng GIST và mô hình máy học SVM cho phân lớp ảnh. Lê Thanh Trúc [4] sử dụng biến đổi Hough trong tiền xử lí, kết quả nhận dạng được bảng điểm của Trường Đại học Tây Đô đạt độ chính xác 97,3%. Tuy nhiên, thời gian thực hiện xử lí lâu đến 14,51 giây. Hơn nữa, giải pháp này chỉ áp dụng cho các số tròn từ 0 đến 9 trên bảng điểm. Điều này chưa thực tế với tình hình quản lí điểm số. Nghiên cứu của Võ Ngọc Lợi [5] nhận dạng được điểm số có phần thập phân một chữ số lẻ với độ chính xác 94,4% trong thời gian 2,7 giây trên bảng điểm của Trường Đại học Bạc Liêu. Các nghiên cứu đã tập trung xử lí trên dữ liệu thực tiễn tại đơn vị. Tuy nhiên, các nghiên cứu chưa cho kết quả chính xác cao hoặc do thời gian xử lí lâu do nhiều nguyên nhân như viết thiếu nét, sửa điểm, kí nháy, viết lấn dòng, viết chữ dính với nhau, chất lượng đầu vào của ảnh kém, nhiều nhiễu.

Do cấu trúc bảng điểm của Trường Cao đẳng Nghề Sóc Trăng chứa nhiều vùng thông tin khác biệt so với bảng điểm của các nghiên cứu trên nên việc áp dụng kết quả các nghiên cứu đã có vào bảng điểm thực tế tại Trường Cao đẳng Nghề Sóc Trăng là không thể. Trong bài viết này, chúng tôi đề xuất hệ thống nhận dạng bảng điểm thực tế tại Trường Cao đẳng Nghề Sóc Trăng bằng phương pháp trích đặc GIST và giải thuật máy học SVM trên vùng dữ liệu điểm số tổng kết được viết bằng tay; đồng thời, chúng tôi kết hợp thư viện tesseract-OCR để nhận dạng trên vùng dữ liệu mã số học sinh, sinh viên (HSSV).

II. TỔNG QUAN NGHIÊN CỨU

A. Rút trích đặc trưng GIST

GIST là một đặc trưng toàn cục biểu diễn nội dung ảnh được Oliva & Torralba [6] đề xuất năm 2001. Đặc trưng GIST thể hiện dưới dạng một vector được tính toán từ kết quả của việc áp dụng các bộ lọc Gabor lên ảnh. Từ dữ liệu ảnh đầu vào, sau khi trích đặc trưng sẽ cho ra một vector có 960 chiều. Sơ lược các bước tiến hành như sau:

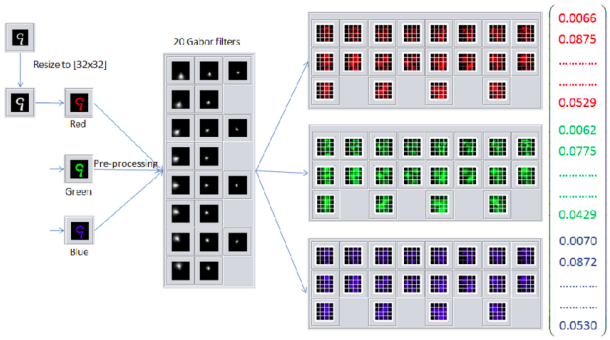
Bước 1: Ảnh đầu vào sau khi được tiền xử lí sẽ được tách ra thành ba kênh màu Red-Green-Blue riêng biệt.

Bước 2: Áp dụng phép biến đổi Fourier trên mỗi kênh màu.

Bước 3: Ứng với mỗi ảnh Fourier áp dụng lần lượt hai mươi bộ lọc Gabor lên ảnh. Bộ lọc Gabor được tạo ra ở ba scales và tám hướng. Trong đó, scale 1 và scale 2 sử dụng tám bộ lọc, scale 3 sử dụng bốn bộ lọc.

Bước 4: Cuối cùng, kết quả của mỗi bộ lọc được đưa qua phép biến đổi Fourier ngược, sau đó chia thành 16 vùng bằng nhau và trích đặc trưng. Kết quả của mỗi vùng là một đặc trưng. Như vậy số chiều của đặc trưng GIST là:

$$3 \times (8+8+4) \times 16 = 960 \qquad (1)$$

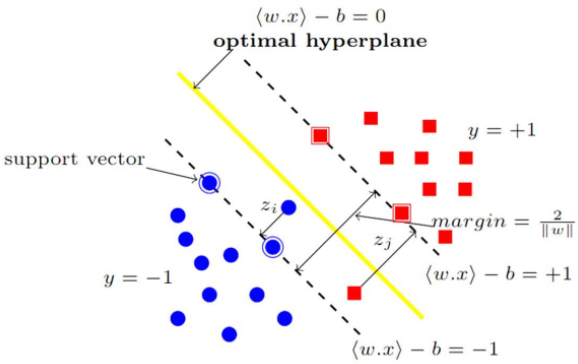


Hình 1: Minh họa trích đặc trưng GIST [7]

B. *Máy học vector hỗ trợ (SVM)*

Máy học vector hỗ trợ được Vapnik nghiên cứu từ những năm 1965, đến những năm 1990 thì giải thuật [8] được chính thức phát triển mạnh, trở thành công cụ hữu hiệu và phổ biến của lĩnh vực máy học, nhận dạng và khai mỏ dữ liệu, có thể được sử dụng cho phân loại, hồi quy hoặc các nhiệm vụ khác.

SVM thuộc dạng máy học có giám sát, là mô hình xây dựng một siêu phẳng hoặc một tập hợp các siêu phẳng trong một không gian nhiều chiều hoặc vô hạn chiều. Siêu phẳng tối ưu phải là siêu phẳng tách hai lớp xa nhất có thể.



Hình 2: Mô hình SVM phân lớp tuyến tính cho bài toán hai lớp [9]

Đối với vấn đề phân lớp nhị phân tuyến tính như Hình 2, siêu phẳng tối ưu phân tập dữ liệu thành hai lớp là siêu phẳng có thể tách rời dữ liệu thành hai lớp riêng biệt với lề lớn nhất. Việc chia cắt được thực hiện nhờ vào hai siêu phẳng hỗ trợ song song. Siêu phẳng hỗ trợ lớp (+1) màu xanh lá cây nằm về bên phải của siêu phẳng này. Tương tự, siêu phẳng hỗ trợ cho lớp (-1) màu xanh dương nằm về bên trái của siêu phẳng này. Theo đó, chúng ta cần tìm siêu phẳng H: y=w.x + b=0 và hai siêu phẳng H1, H2 hỗ trợ song song với H và có cùng khoảng cách đến H với điều kiện không có phần tử nào của tập mẫu nằm giữa H1 và H2.

Khi đó:

$$w.x_i + b >= +1 \text{ với } y=+1$$
$$w.x_i - b >= -1 \text{ với } y=-1 \quad (2)$$

Kết hợp hai điều kiện trên ta có: $y_i(w.x_i + b) >= 1$

Khoảng cách giữa siêu phẳng hỗ trợ song song được gọi là lề và được tính bằng lề = $\frac{2}{\| W \|}$.

Trong đó: $\frac{2}{\| W \|}$ là độ lớn của vector w.

Đối với bài toán nhiều lớp, SVM có thể xây dựng trực tiếp mô hình cho nhiều lớp từ bài toán tối ưu cho k lớp. Các phương pháp thường dùng như: 1 – tất cả (one vs rest): mỗi mô hình phân tách một lớp từ các lớp khác, xây dựng k mô hình cho k lớp; 1 – 1 (one vs one): mỗi mô hình phân tách hai lớp, xây dựng $\frac{k \times (k-1)}{2}$ mô hình cho k lớp và phương pháp khác, như phân tách hai nhóm, mỗi nhóm có thể bao gồm nhiều lớp, xác định cách phân tách nhóm sao cho có lợi nhất.

Trong nghiên cứu này, chúng tôi sử dụng thư viện LibSVM [10] để phân lớp SVM, sử dụng phân lớp theo mô hình 1–1. LibSVM thực hiện huấn luyện mô hình và dự đoán dựa trên tập tin có cấu trúc: $label att-i:val-i....att-n:val-n$.

C. *Công cụ Tesseract OCR*

Tesseract là một công cụ nhận dạng kí tự quang học được thiết kế chạy trên các nền tảng hệ điều hành khác nhau. Đây là phần mềm miễn phí, được phát hành theo giấy phép Apache, phiên bản 2.0. Kể từ năm 2006, nó được cải thiện rộng rãi bởi Google.

Với những cải tiến về khả năng nhận dạng kí tự quang học [11], Tesseract phiên bản 4 [12] sử dụng mạng bộ nhớ dài-ngắn (LSTM), một dạng đặc biệt của mạng nơ-ron hồi quy (RNN) cho kết quả nhận dạng kí tự quang học khá tốt, trong đó có cả tiếng Việt.

III. GIẢI PHÁP ĐỀ XUẤT

A. Lưu đồ xử lí bảng điểm

Ảnh sau khi scan sẽ được tiến hành các bước tiền xử lí để thu được ảnh tốt hơn cho các bước tiếp sau, bước tiền xử lí sử dụng các hàm làm mịn ảnh, hàm chuyển sang ảnh xám, nhị phân ảnh với ngưỡng thích nghi.

Sau bước tiền xử lí, thực hiện định vị các vị trí để lọc lấy thông tin, cụ thể: lọc chọn vùng bảng chứa mã số, điểm số của HSSV; lọc vùng chứa mã số HSSV (dạng chữ in) và vùng chứa điểm số thập phân (dạng chữ viết tay); vùng chứa điểm số, tiếp tục thực hiện tách các kí số của chữ số thập phân. Khi lấy được vùng thông tin cần thiết, tiếp theo nghiên cứu huấn luyện mô hình và nhận dạng. Nội dung này được chia thành hai công việc cần phải thực hiện:

- Công việc 1: Nhận dạng vùng chữ in sẽ được thực hiện bằng Tesseract OCR;

- Công việc 2: Nhận dạng đối với chữ viết tay. Để thực hiện việc này, việc đầu tiên là sưu tập dữ liệu, chuyển đổi định dạng ảnh cho phù hợp, tiếp theo trích chọn đặc trưng để đưa vào huấn luyện mô hình. Tương tự với việc huấn luyện, ảnh cần nhận dạng cũng phải trải qua nội dung tiền xử lí, trích chọn đặc trưng sau đó nhận dạng dựa trên mô hình đã huấn luyện.

Cuối cùng, dữ liệu của kết quả nhận dạng sẽ được trích xuất, lưu trữ vào file Excel.

B. Tiền xử lí

Đầu tiên, ảnh đầu vào sẽ được chuyển đổi sang dạng ảnh đa mức xám bằng cách sử dụng hàm cvtColor(src,gray,CV_BGR2GRAY).

Tiếp theo, nghiên cứu tiến hành làm mịn bằng bộ lọc trung bình với hàm nhân ma trận 5×5. Sau đó, cố gắng giữ lại những nét cần thiết, nội dung được tiến hành bằng các phép toán hình thái như làm dày và ăn mòn dựa trên hàm cấu trúc hình chữ nhật kích thước 3×3 (MORPH_RECT,size (3×3)).

Chuyển sang ảnh nhị phân là cần thiết, ảnh nhị phân với lớp nền và lớp đối tượng đối lập nhau, làm cơ sở cho việc phát hiện đối tượng cần thiết để lấy sẽ được thuận lợi hơn. Chúng tôi chọn phân ngưỡng thích nghi (adaptiveThreshold) để thực hiện chuyển sang ảnh nhị phân, với mong muốn giữ được nhiều nét ảnh của bảng điểm để quá trình huấn luyện, nhận dạng chính xác hơn. Cụ thể, nghiên cứu đã sử dụng phương thức phân ngưỡng thích nghi với hàm gausian giúp khử nhiễu trong quá trình xử lí (ADAPTIVE_THRESH_GAUSSIAN_C).

C. Trích lấy vùng thông tin cần thiết trên ảnh

1) Trích vùng bảng điểm

Vùng cần quan tâm trên ảnh bảng điểm là vùng chứa mã số HSSV và vùng chứa điểm tổng kết. Tất cả các nội dung này nằm trên lưới bảng điểm của ảnh bảng điểm. Vì thế, chúng ta cần phải xác định được vùng bảng điểm này, Hình 4 minh họa vùng bảng điểm được cắt ra từ ảnh bảng điểm.

Để xác định vùng bảng điểm, đầu tiên, chúng ta chuyển ảnh bảng điểm sang ảnh nhị phân. Tiếp theo, giãn nở đường biên trên ảnh nhị phân bằng các phép toán hình thái ăn mòn (erode) và làm dày (dilate) dựa trên hàm cấu trúc hình chữ nhật kích thước 3×3 (MORPH_RECT,size (3×3)). Sau đó, sử dụng findContours để tìm đường biên trên ảnh nhị phân sử dụng kiểu CV_RETR_EXTERNAL để lấy những đường biên bên ngoài bao bọc vùng bảng điểm và cần tọa độ của bốn đỉnh hình chữ nhật bảng điểm nên nghiên cứu sử dụng phương pháp CV_CHAIN_APPROX_SIMPLE trong quá trình tìm biên. Tiếp theo, kiểm tra đường biên có diện tích phù hợp để lấy. Cuối cùng, sử dụng approxPolyDP để tạo đa giác từ đường biên; thực hiện cắt để tạo ảnh chứa vùng bảng điểm được cắt ra.

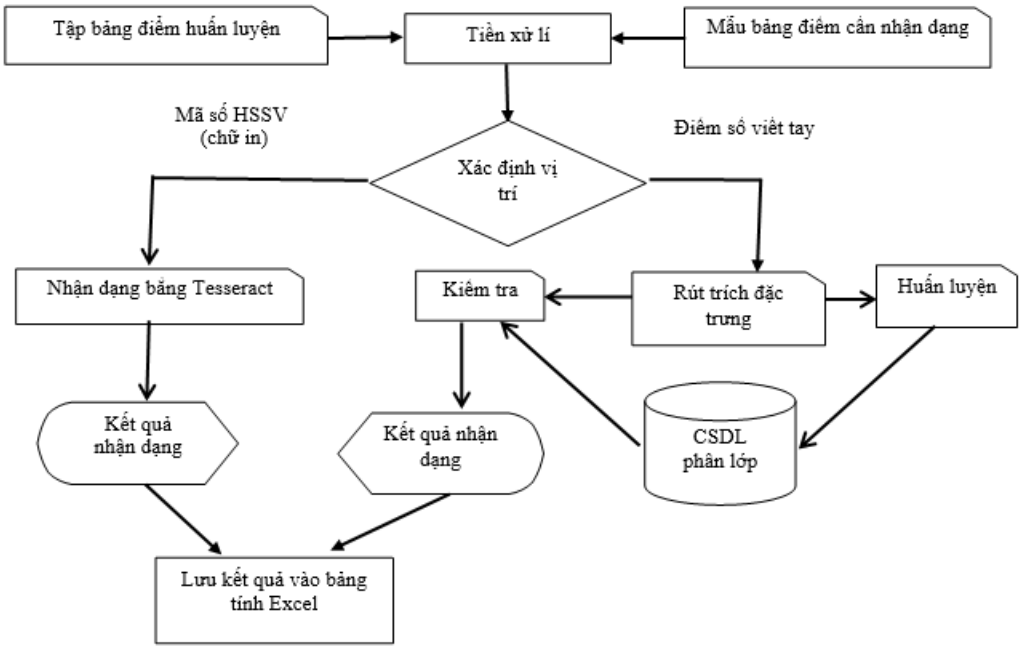
Kết quả nghiên cứu cho thấy, việc xác định tứ giác của vùng bảng điểm là 100% trừ các trường hợp bị lỗi mất đường biên hình chữ nhật trong quá trình scan.

2) Lấy thông tin vùng mã số HSSV và vùng điểm tổng kết

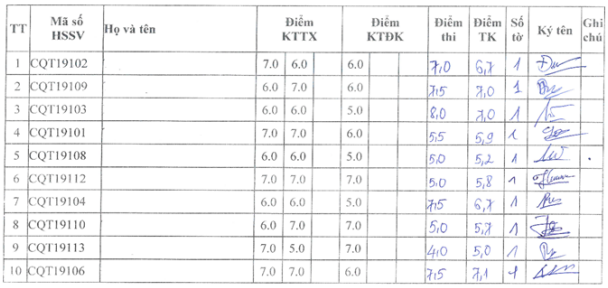
Trên cơ sở vùng bảng điểm đã được trích ra, chúng ta tiếp tục lấy các thông tin ở cột mã số HSSV (vùng chứa chữ in) và điểm tổng kết (vùng chữ viết tay).

Định vị tọa độ các vùng lấy thông tin

Với ảnh đầu vào là vùng bảng điểm đã được cắt ra từ file ảnh bảng điểm gốc, tiến hành nhị phân ảnh đầu vào, tiếp theo sử dụng các phép toán hình thái giãn nở ảnh, lọc nhiễu, bitwise_not để làm nổi các đường biên dọc và đường biên ngang



Hình 3: Lưu đồ xử lí bảng điểm



TT	Mã số HSSV	Họ và tên	Điểm KTTX		Điểm KTĐK		Điểm thi	Điểm TK	Số tờ	Ký tên	Ghi chú
1	CQT19102		7.0	6.0	6.0		7,0	6,7	1		
2	CQT19109		6.0	7.0	6.0		7,5	7,0	1		
3	CQT19103		6.0	6.0	5.0		8,0	7,0	1		
4	CQT19101		7.0	7.0	6.0		5,5	6,9	1		
5	CQT19108		6.0	6.0	5.0		5,0	5,2	1		
6	CQT19112		7.0	7.0	7.0		5,0	5,8	1		
7	CQT19104		6.0	6.0	5.0		7,5	6,7	1		
8	CQT19110		6.0	7.0	7.0		5,0	5,7	1		
9	CQT19113		7.0	5.0	7.0		4,0	5,0	1		
10	CQT19106		7.0	7.0	6.0		7,5	7,1	1		

Hình 4: Vùng bảng điểm được cắt ra từ ảnh bảng điểm

của vùng bảng điểm. Cuối cùng, sử dụng phép chiếu trên các đường biên với thứ tự cột phù hợp để lưu lại giá trị tọa độ.

Tách dòng thông tin mã số và điểm số

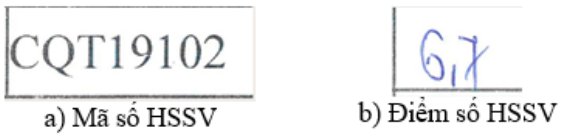
Thuật toán lấy thông tin mã số và điểm số

Đầu vào: Ảnh biên ngang Ih

Đầu ra: Ảnh từng dòng mã số (Icode) và điểm số (Imark) Xử lí:

Bước 1: Xác định tổng số dòng của vùng bảng điểm từ ảnh biên ngang Ih.

Bước 2: Duyệt qua các dòng: từ dòng 3 đến hết số dòng được xác định, thực hiện cắt ảnh của từng dòng (mã số, điểm số), trường hợp ảnh được cắt ra tại số dòng có chỉ số < 10 thì thêm vào trước đó số 0 để đảm bảo thứ tự phục vụ cho việc trích đặc trưng sau này. Thông tin cắt được lưu trữ riêng biệt tương ứng cho từng loại mã số (Icode) và điểm số (Imark). Áp dụng thư viện "jpg2pgm" trên các ảnh điểm số để có cấu trúc phù hợp với trích chọn đặc trưng GIST.



Hình 5: Mã số và điểm số HSSV được cắt ra

Tách ảnh số thập phân (phần điểm số)

Thuật toán tách ảnh số thập phân:

Đầu vào: Ảnh ô điểm số (Imark)

Đầu ra: Ảnh điểm số phần nguyên (Iint), ảnh điểm số phần thập phân (Idec).

Xử lí:

Bước 1: Xác định thư mục chứa các ảnh điểm số (Imark).

Bước 2: Lần lượt duyệt qua các tập tin trong thư mục vừa xác định.

Bước 3: Tương ứng với mỗi tập tin, thực hiện:

Bước 3.1: Nhị phân ảnh theo ngưỡng

Bước 3.2: Áp dụng giải thuật xóa nhiễu. Do trong quá trình cắt phần điểm số có thể tồn các đường nhiễu ngang – dọc, điều này cần loại bỏ để đảm bảo độ chính xác khi trích đặc trưng. Để xóa nhiễu đường đen ngang – dọc này, ta thực hiện: đối với đường đen ngang, tính tổng số điểm đen theo dòng, nếu tổng điểm đếm được theo dòng > 0,4 * tổng số cột thì vẽ đường thẳng với điểm ảnh trắng trên dòng; đối với đường thẳng đứng, tính tổng số điểm đen theo cột, nếu tổng điểm đếm được theo cột > 0,6 * tổng số dòng thì vẽ đường thẳng với điểm ảnh trắng trên cột.

Bước 3.3: Áp dụng các phép giãn, nở ảnh để hiện rõ các đường nét cần thiết.

Bước 3.4: Sử dụng biểu đồ chiếu để xác định vùng điểm số phần nguyên và vùng điểm số phần thập phân, làm cơ sở để cắt ảnh.

Bước 3.5: Thực hiện cắt ảnh dựa trên ảnh biểu đồ chiếu, phần vùng đen bên trái của biểu đồ được cắt lưu lại với tên ảnh ban đầu +1 (các ảnh đầu ra thuộc Iint), phần vùng đen bên phải của biểu đồ được cắt lưu lại với tên ảnh ban đầu +2 (các ảnh đầu ra thuộc Idec).

a) Xóa nhiễu b) Biểu đồ chiếu c) Kết quả tách số

Hình 6: Ảnh điểm số phần nguyên và phần thập phân được cắt ra

D. Trích đặc trưng GIST

Trích đặc trưng GIST được áp dụng trên dữ liệu chữ số viết tay. Dựa trên phương pháp này, mỗi ảnh kí tự số được rút trích tương ứng vector 960 chiều. Sau bước này, tập dữ liệu ảnh đưa về dạng bảng hoặc ma trận mà ở đó mỗi ảnh là một dòng có 960 cột (chiều), mỗi kí số được gán nhãn (lớp tương ứng là 0,1,...9).

Tập dữ liệu được lưu trữ dưới dạng đặc trưng GIST, có cấu trúc như sau:

```
Label1 index1:value1_1 index2:value2_1 ..... index960:value960_1
Label2 index1:value1_2 index2:value2_2 ..... index960:value960_2
...
Labeln index1:value1_n index2:value2_n ..... index960:value960_n
```

Trong đó:

- Label là giá trị nhãn của tập huấn luyện/tập kiểm tra. Đối với việc phân lớp, nó xác định một lớp. Đối với hồi quy, nó là một số thực bất kì.
- Index là số nguyên bắt đầu từ 1, các giá trị index sau sẽ tăng dần đến 960 (số chiều GIST).
- Value là một số thực.

Riêng phần chữ in sẽ không thực hiện việc rút trích đặc GIST, nội dung này sẽ sử dụng thư viện Tesseract OCR để nhận dạng và cho kết quả tương ứng.

Kết quả bảng điểm sau khi nhận dạng được có thể ghi dữ liệu ra tập tin để lưu trữ hoặc đưa vào hệ thống quản lí điểm số tại trường.

IV. KẾT QUẢ THỰC NGHIỆM

Thực nghiệm được tạo ra dựa trên bộ công cụ QtCreator bằng ngôn ngữ C++ cùng với một số thư viện mã nguồn mở như Opencv 2.4 [13]; libsvm 3.24 [10]; Tesseract OCR 4.1.0 [12] và một số hàm nguồn mở như jpeg2pgm [14], Leargist [15], QtXlsxWriter-master. Các bộ công cụ, ngôn ngữ, thư viện và hàm được dùng để thực nghiệm chạy trên nền tảng hệ điều hành mã nguồn mở Ubuntu. Các thí nghiệm được chạy trên máy tính cá nhân, bộ vi xử lí Intel Core i5-8256U, 1.60Ghz, 8 nhân và bộ nhớ RAM 4GB.

A. Chuẩn bị dữ liệu

Dữ liệu là các bảng điểm được sưu tập thông qua scan dưới dạng file *.jpg. Các file scan được thực hiện trên máy Sharp MX-M315N, tiêu chuẩn scan: 600dpi, Jpeg, full color. Số lượng bảng điểm thu thập là 210, trong đó, chúng tôi chọn ra các bảng điểm phù hợp, có 114 bảng điểm được xem là khả thi. Các bảng điểm không được chọn do giảng viên đã tạo nên vùng liên thông với vùng chứa điểm, viết kí số dính liền với nhau, viết lấn dòng trong quá trình viết tay.

Sau khi lọc các bảng điểm có tính khả thi, nghiên cứu tiến hành tách các số ở ô điểm số viết tay để khảo sát chọn ra những kí số phù hợp. Quá trình này đã trích xuất được 2819 ảnh

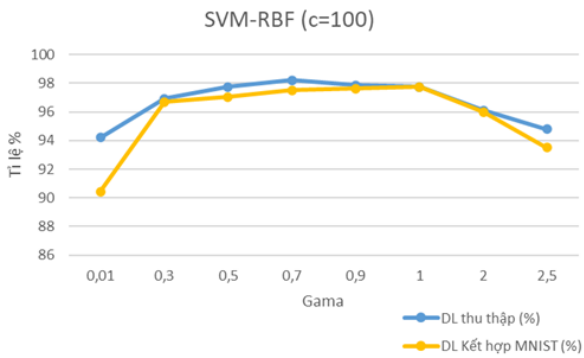
kí số viết tay trên 114 ảnh bảng điểm, tiến hành phân bổ vào các thư mục từ 0 đến 9 tương ứng với các kí số đã thu thập. Tiếp theo, thực hiện phân bổ thành tập dữ liệu huấn luyện (70%) có 1972 ảnh kí số và tập dữ liệu kiểm tra (30%) có 847 ảnh kí số, kết quả tập dữ liệu được phân bổ như Bảng 1.

Nghiên cứu cũng sử dụng bộ dữ liệu MNIST do Yan Lecun et al. phát triển [3], bộ dữ liệu gồm 60.000 dữ liệu cho tập huấn luyện và 10.000 dữ liệu cho tập kiểm tra.

B. Huấn luyện

Huấn luyện nhằm tạo ra mô hình phục vụ cho quá trình nhận dạng tại vùng điểm số được viết bằng tay trên bảng điểm. Nghiên cứu thực hiện huấn luyện hai mô hình: mô hình thứ nhất, dựa trên tập dữ liệu thu thập; mô hình thứ hai, dựa trên tập dữ liệu kết hợp giữa dữ liệu thu thập với bộ dữ liệu MNIST.

Để huấn luyện mô hình, đầu tiên tiến hành trích đặc trưng GIST với tập dữ liệu huấn luyện, tiếp theo chuyển đổi tập tin đặc trưng GIST thành dạng để sử dụng được với SVM, cuối cùng tập tin mô hình được tạo dựa trên thư viện LibSVM. Khảo sát độ chính xác của hai trường hợp trên khi thực hiện điều chỉnh tham số gamma trên SVM (c = 100), kết quả khảo sát như Bảng 2.



Hình 7: So sánh độ chính xác của các mô hình sử dụng SVM với hàm nhân RBF, c = 100

Kết thúc quá trình khảo sát, chúng tôi nhận thấy SVM với hàm nhân RBF (c = 100) có trường hợp cho độ chính xác cao nhất 98,23% tại giá trị gamma = 0,7. Vì thế, chúng tôi chọn mô hình với tham số gamma = 0,7 sử dụng trong quá trình thực nghiệm.

C. Nhận dạng

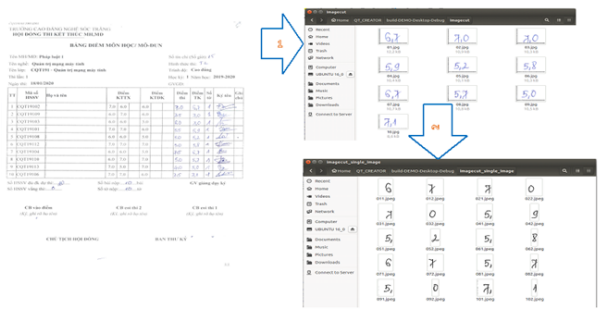
Việc nhận dạng được thực hiện trên 100 bảng điểm thu thập. Đầu tiên, nghiên cứu tiến hành ghi nhận việc cắt vùng chứa bảng điểm, kết quả cho thấy việc xác định vùng chứa điểm là hoàn toàn chính xác trên các bảng điểm thử nghiệm, chỉ trừ trường hợp ảnh đầu vào với chất lượng quá kém như ảnh độ phân giải thấp, ảnh với đường nét quá mờ.

Tiếp theo, ghi nhận việc xác định vị trí các ô chứa thông tin cần lấy gồm: mã số HSSV dạng chữ in, điểm tổng kết dạng viết tay. Việc định vị, tách lấy các ô thông tin cho kết quả chính xác 100%. Các ô chứa mã số HSSV sẽ được nhận dạng trực tiếp với Tesseract OCR. Riêng các ô chứa điểm tổng kết được viết bằng tay, tiếp tục thực hiện tách các kí số trước khi đưa vào nhận dạng, quá trình tách lấy điểm số bằng phép chiếu (projection) chưa đạt độ chính xác 100% do nhiều lí do, điển hình như chữ số viết dính với nhau, thiếu khoảng cách giữa hai phần trong số thập phân, nét viết không đồng đều.

Cuối cùng, kết quả nhận dạng mã số và điểm số của HSSV sẽ được trích xuất lưu trữ vào file dữ liệu dưới dạng tập tin Excel.

1) Nhận dạng vùng chứa điểm viết tay: Nhận dạng vùng chứa điểm dạng viết tay dựa trên mô hình SVM (hàm nhân RBF, c = 100, gamma = 0,7) đã huấn luyện:

- Giai đoạn 1: Thực hiện tách ô điểm, tách từng kí số như Hình 8.



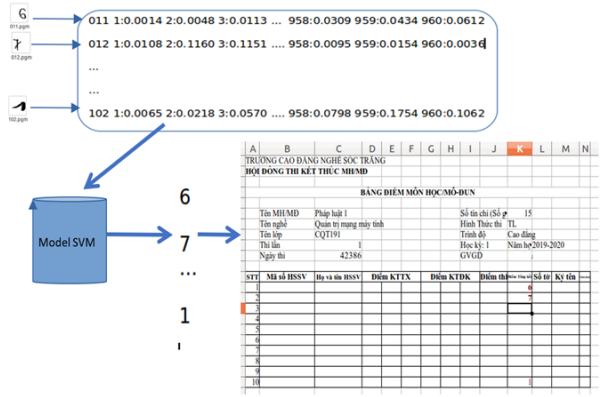
Hình 8: Quá trình định vị, tách ô, tách kí số

Bảng 1: Phân bổ dữ liệu huấn luyện

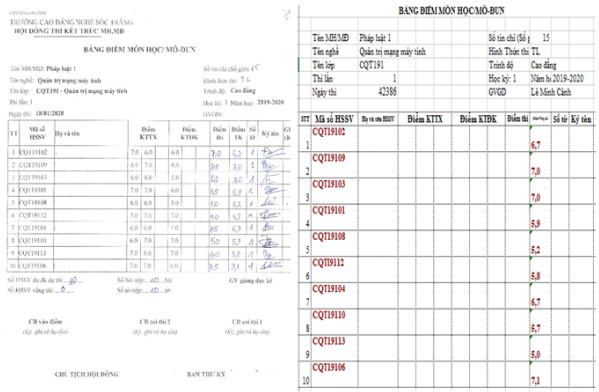
Phân bố dữ liệu/Nhãn	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Tổng**
Thu thập	**286**	**181**	**215**	**235**	**326**	**342**	**320**	**334**	**352**	**228**	**2819**
Tập Train (70%)	200	127	150	165	228	239	224	233	246	160	**1972**
Tập Test (30%)	86	54	65	70	98	103	96	101	106	68	**847**

Bảng 2: Thống kê khảo sát độ chính xác các mô hình

Bảng so sánh phân lớp dữ liệu với SVM (hàm nhân RBF, c = 100)		
Gamma	**Dữ liệu thu thập (%)**	**Dữ liệu kết hợp với MNIST (%)**
0,01	94,2149	90,4368
0,3	96,9303	96,6942
0,5	97,7568	97,0484
0,7	98,2290	97,5207
0,9	97,8749	97,6387
1	97,7568	97,7568
2	96,1039	95,9858
2,5	94,8052	93,5065



Hình 9: Quá trình trích đặc trưng, nhận dạng và ghi kết quả



a) Bảng điểm cần nhận dạng b) File kết quả nhận dạng (dạng Excel)

Hình 10: Kết quả nhận dạng và lưu trữ bảng điểm

- Giai đoạn 2: Trích đặc trưng ảnh được cắt, dựa vào mô hình đã huấn luyện để nhận dạng, đưa kết quả nhận dạng vào file Excel như Hình 9.

2) Nhận dạng vùng chứa mã số và ghi kết quả vào file Excel: Các ô chứa mã số HSSV được tách ra từ bảng điểm sẽ được thực hiện nhận dạng bằng thư viện Tesseract OCR. Kết quả nhận dạng sẽ được lưu trữ vào file dạng text. Sau đó, dữ liệu này sẽ được trích xuất vào file Excel, quá trình trích xuất này được thực hiện bởi thư viện QtXlsxWriter-master.

3) Đánh giá độ chính xác: Nghiên cứu đã tiến hành thực nghiệm nhận dạng trên 100 bảng điểm cho độ chính xác trung bình 96,57% đối với dạng chữ in và 93,55% đối với dạng điểm số viết tay. Chúng tôi thực hiện đối sánh trên hai nhóm bảng điểm (nhóm A – bảng điểm chứa dữ liệu trong tập huấn luyện; nhóm B – bảng điểm mới), kết quả ghi nhận và thống kê như Bảng 3.

Việc nhận dạng điểm số viết tay cho kết quả độ chính xác chưa đạt 100% do chất lượng ảnh đầu vào của bảng điểm chưa đồng đều về độ sáng,

Bảng 3: Thống kê kết quả nhận dạng mã số HSSV và điểm số viết tay

Nhóm bảng điểm	A	B
Độ chính xác trung bình đối với điểm số viết tay	95,52%	92,89%
Độ chính xác trung bình đối với mã số HSSV (dạng chữ in)	93,01%	97,76%

cách viết kí số thập phân thiếu nhất quán và đồng đều. Kết quả quá trình tách kí số thập phân chưa đạt hiệu quả cao đã làm ảnh hưởng đến quá trình máy nhận dạng sai.

Đối với việc nhận dạng mã số HSSV (dạng chữ in), nghiên cứu sử dụng thư viện Tesseract để nhận dạng, cho kết quả độ chính xác cũng chưa đạt 100%, do quá trình nhận dạng còn phụ thuộc vào một số yếu tố như dạng phông chữ thể hiện của dữ liệu trên từng bảng điểm, độ phân giải DPI trên ảnh, độ nhiễu của ảnh đầu vào.

V. KẾT LUẬN

Bài báo đã trình bày những vấn đề liên quan đến các bài toán cơ bản như phát hiện vùng chứa bảng điểm, kí tự, trích đặc trưng kí tự, mô hình huấn luyện. Giải pháp đề xuất kết hợp với sự hỗ trợ của máy học đã tạo một hệ thống nhận dạng chữ số viết tay kết hợp với nhận dạng chữ in để phục vụ cho nhập điểm. Kết quả thực nghiệm trên 100 bảng điểm cho độ chính xác trung bình 96,57% đối với dạng chữ in và 93,55% đối với dạng điểm số viết tay. Thời gian nhận dạng trung bình cho một bảng điểm là 7,9 giây. Điểm nổi bật của nghiên cứu này là sự kết hợp nhận dạng chữ in và chữ số viết tay trên một bảng điểm thực tế tại Trường Cao đẳng Nghề Sóc Trăng.

TÀI LIỆU THAM KHẢO

[1] Jain G, Ko J. *Handwritten Digits Recognition ECE462*. Multimedia Systems, Project Report, University of Toronto. 2008; 1–3. Available from: http://individual.utoronto.ca/gauravjain/ECE462-HandwritingRecognition.pdf [Accessed 25th June 2020]

[2] Đỗ Thanh Nghị, Phạm Nguyên Khang. Nhận dạng ký tự số viết tay bằng giải thuật máy học. *Tạp chí Khoa học Trường Đại học Cần Thơ*. 2013; 27:64–71.

[3] Yann Lecun, Corinna Cortes, Christopher J.C. Burges. *The MNIST database of handwritten digits*. 1998. Truy cập từ: http://yann.lecun.com/exdb/mnist/ [Ngày truy cập: 25/6/2020].

[4] Lê Thanh Trúc. Nhận dạng điểm số viết tay phục vụ công tác lên điểm Phòng Đào tạo Trường Đại học Tây Đô. *Tạp chí Khoa học Trường Đại học Cần Thơ*. 2015; 15:79–87.

[5] Võ Ngọc Lợi, Trần Cao Đệ. Nghiên cứu nhận dạng điểm số viết tay có phần thập phân. Trong *Kỉ yếu Hội thảo toàn quốc về Công nghệ Thông tin*. Thành phố Cần Thơ: NXB Đại học Cần Thơ. 2017.

[6] Oliva A, Torralba A. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *International Journal of Computer Vision*. 2001; 42(3):145–175.

[7] Do T. N, Pham N. K. Handwritten digit recognition using GIST descriptors and random oblique decision trees. In *The National Foundation for Science and Technology Development (NAFOSTED) Conference on Information and Computer Science*. Springer, Cham. March, 2014: pp. 1–15.

[8] Vapnik V. N. *The Nature of Statistical Learning Theory*. Springer-Verlag. 1995.

[9] Phạm Nguyên Khang, Trần Nguyễn Minh Thư, Đỗ Thanh Nghị. Điểm danh bằng mặt người với đặc trưng GIST và máy học véc-tơ hỗ trợ. Trong *Kỷ yếu Hội nghị Quốc gia lần thứ X về Nghiên cứu cơ bản và ứng dụng Công nghệ Thông tin (FAIR)*. Đà Nẵng. 2017. DOI: 10.15625/vap.2017.00019. .

[10] Chang C. C, Lin C. J. LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*. 2011; 2(3):1–27.

[11] Tanvir S. H, Khan T. A, Yamin A. B. Evaluation of Optocal Character Recognition Algorithms and Feature Extraction Techniques. In *The Sixth International Conference on Innovative Computing Technology*. 2016: pp.326–331.

[12] Lihang Li. *Tesseract Open Source OCR Engine (main repository)*. 2019. Truy cập từ: https://github.com/tesseract-ocr/tesseract [Ngày truy cập: 30/6/2020].

[13] OpenCV 2.4.13.7. *Tài liệu về thư viện nguồn mở OpenCV*. 2019. Truy cập từ: https://docs.opencv.org/2.4/ [Ngày truy cập: 30/6/2020].

[14] Lihang Li. *A simple utility to convert JPEG to PGM*. 2013. Truy cập từ: https://github.com/hustcalm/jpg2pgm [Ngày truy cập: 30/6/2020].

[15] Tiangang Song. *A simple C++ Wrapper of Lear's GIST implementation using OpenCV*. 2014. Truy cập từ: https://github.com/whu-tgsong/LibGIST [Ngày truy cập: 30/6/2020].