

# MỘT SỐ VẤN ĐỀ TRONG NHẬN DIỆN CHỮ SỐ VIẾT TAY

## SOME ISSUES IN HANDWRITTEN DIGIT RECOGNITION

NGUYỄN THU NGUYỆT MINH<sup>(\*)</sup>, TRẦN VĂN ĐỒNG<sup>(\*\*)</sup> và NGUYỄN ANH TUẤN<sup>(\*\*\*)</sup>

**TÓM TẮT:** Nhận diện chữ số viết tay (Handwritten Digit Recognition) là một kỹ thuật ứng dụng các thuật toán máy học dùng để nhận diện và phân lớp chữ số viết tay dưới dạng hình ảnh. Một số thuật toán máy học phổ biến hiện nay bao gồm K Nearest Neighbors (KNN), Support Vector Machine (SVM), Stochastic Gradient Descent, Artificial Neuron Network (ANN), Hidden Markov Model (HMM)... Trong phạm vi nghiên cứu, tác giả chỉ đề cập các vấn đề liên quan đến độ chính xác của một số thuật toán đã thực nghiệm.

**Từ khóa:** nhận diện chữ số viết tay, histogram các gradient định hướng, tập dữ liệu MNIST, thư viện OpenCv.

**ABSTRACT:** Handwritten digit recognition is a technique that using machine learning algorithms for recognizing and classifying handwritten digits in form of an image. Some algorithms are popularly used in such tasks that consist of K nearest neighbors (KNN), support vector machine (SVM), stochastic gradient descent, artificial neuron network (ANN), Hidden Markov Model (HMM) and so on. In this paper, we discuss some issues that affected on the precision of implemented algorithms in our application.

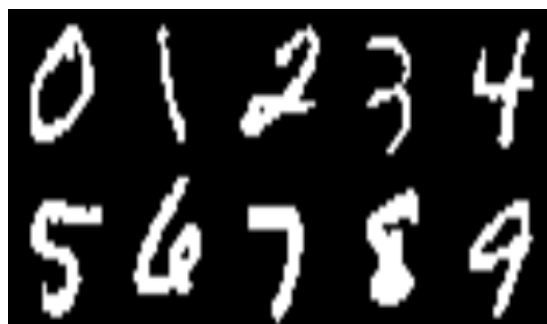
**Key words:** Handwritten digit recognition, Histogram of Oriented Gradients, MNIST, OpenCv.

## 1. DỮ LIỆU HUẤN LUYỆN

### 1.1. Mnist

Dữ liệu gồm 70.000 mẫu chữ số viết tay, mỗi mẫu là một ảnh grayscale kích thước 28x28. Gồm:

Chữ số	Số mẫu
0	6.903
1	7.877
2	6.990
3	7.141
4	6.824
5	6.313
6	6.876
7	7.293
8	6.825
9	6.958



**Hình 1.** Minh họa một số mẫu chữ số được trích ra từ tập dữ liệu MNIST

### 1.2. Opencv-digits

Dữ liệu là một ảnh grayscale chứa 5.000 mẫu chữ số viết tay, mỗi mẫu là một ảnh kích thước 20x20, mỗi chữ số viết tay gồm 500 mẫu.

<sup>(\*)</sup>ThS. Trường Đại học Văn Lang, Email: nguyenthunguyetminh@vanlanguni.edu.vn

<sup>(\*\*)</sup>ThS. Trường Trung Cấp Kinh tế Kỹ thuật Quận 12, Email: trvdong@gmail.com

<sup>(\*\*\*)</sup> ThS. Trường Nhân lực Quốc tế, Email: tuannguyenn@gmail.com

Hình 2 cho thấy, tập dữ liệu Opencv-digits là một ma trận 50x100, mỗi phần tử

trong ma trận là một ảnh 20x20, mỗi chữ số chiếm 5 dòng, 100 cột.



Hình 2. Tập dữ liệu Opencv-digits

### 1.3. Semeion (uci)

Tập dữ liệu nhị phân gồm 1593 mẫu,

mỗi mẫu gồm 256 thuộc tính, có thể được biểu diễn dưới dạng một ảnh 16x16.



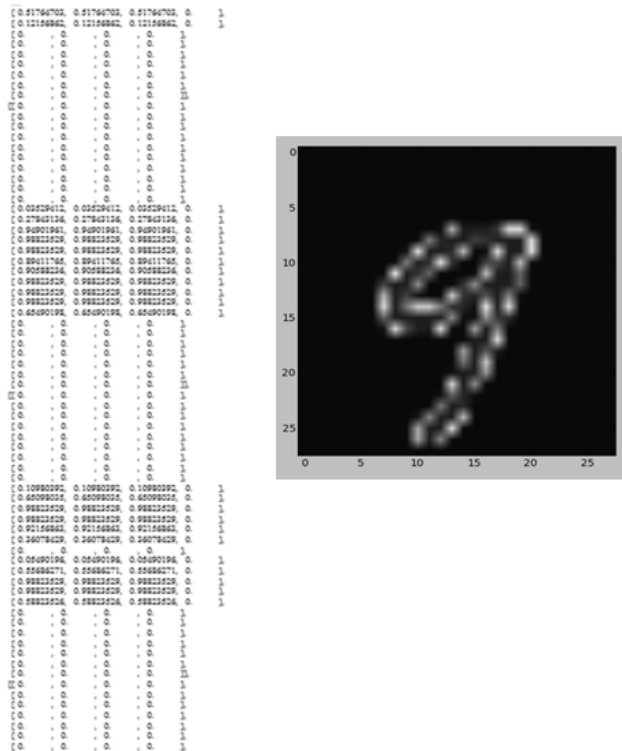
Hình 3. một số mẫu ảnh chữ số viết tay trong tập dữ liệu Semeion (UCI)

## 2. ĐẶC TRƯNG

Đặc trưng là các thuộc tính của đối tượng được sử dụng trong máy học để huấn luyện, nhận diện, phân lớp hoặc dự báo. Trong nhận diện chữ số viết tay, các đặc trưng thường được sử dụng bao gồm:

### 2.1. Đặc trưng thô (Raw features)

Là giá trị của các pixels trong ảnh. Đặc trưng thô được biểu diễn dưới dạng một mảng, các phần tử trong mảng là giá trị RGB của mỗi pixel ảnh. Hình 4 minh họa cho mảng đặc trưng thô của một ảnh chữ số viết tay.

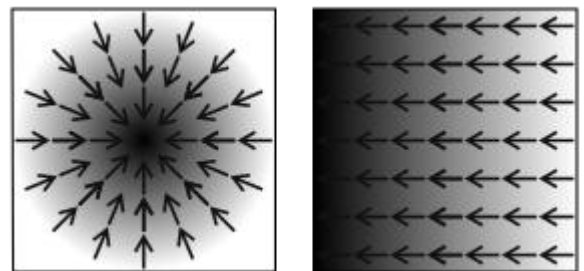


Hình 4. Một phần mảng đặc trưng thô (trái) của ảnh chữ số 9 (phải)

## 2.2. Đặc trưng hog (Histogram of Oriented Gradients features)

Là một vector mô tả đặc trưng của một đối tượng. Hog chỉ biểu diễn những thông tin hữu ích (đặc trưng) của một hình ảnh bằng cách loại bỏ những thông tin thừa có thể gây nhiễu. Vấn đề là thế nào là “thông tin hữu ích”? Giả sử chúng ta muốn tìm nút áo trong một ảnh. Nút áo có hình tròn, đôi khi do góc chụp có thể cho nút áo có hình ellipse. Nút áo thường có một số lỗ để khâu vào áo. Nếu chỉ đơn thuần chỉ dựa vào hình học để phát hiện nút áo trong một ảnh có thể nhầm lẫn với các đối tượng khác có cùng dạng hình học với nút áo. Các lỗ trong nút áo sẽ là “thông tin hữu ích” để phân biệt giữa một cái nút áo với các đối tượng khác có cùng dạng hình học. Trong Hog, đặc trưng được rút ra từ ảnh là sự phân bố các gradient của các pixels.

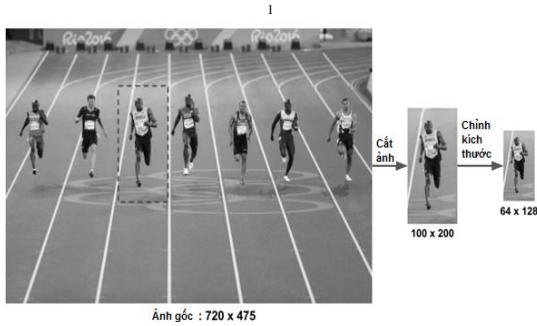
Gradient của mỗi pixel gồm 2 đại lượng: mật độ màu (cường độ gradient) và hướng gradient. Đối với ảnh màu có nhiều channel, cường độ gradient của mỗi pixel là giá trị cường độ gradient lớn nhất trong số các channel tại pixel đó. Hướng gradient tại mỗi pixel là hướng có cường độ gradient tăng lớn nhất. Hình 5 minh họa gradient tại mỗi pixel, trong đó mũi tên xanh biểu diễn hướng gradient, màu sáng tối tượng trưng cho cường độ gradient.



Hình 5. Hai đại lượng gradient của mỗi pixel trong ảnh

### Các bước tính Hog:

**Bước 1: Tiền xử lý ảnh:** Xác định vùng cần tính Hog; Cắt vùng ảnh cần quan tâm; Điều chỉnh kích thước.



Hình 6. Tiền xử lý ảnh: cắt vùng quan tâm và điều chỉnh kích thước

### Bước 2: Tính toán ảnh gradient

Dùng bộ lọc Sobel (Sobel filter) với 2 kernel  $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$  và  $\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$  để tính xấp xỉ gradient theo chiều  $x$  ( $g_x$ ) và chiều  $y$  ( $g_y$ ) ở mỗi pixel. Sau đó, tính cường độ gradient và hướng gradient (quy ra góc) theo các công thức sau:

$$g = \sqrt{g_x^2 + g_y^2} \quad (\text{Công thức 2.1})$$

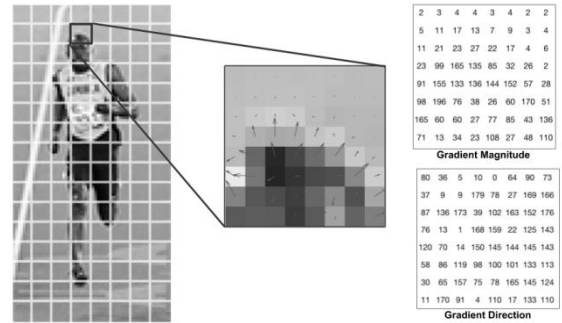
$$\theta = \arctan \frac{g_y}{g_x} \quad (\text{Công thức 2.2})$$



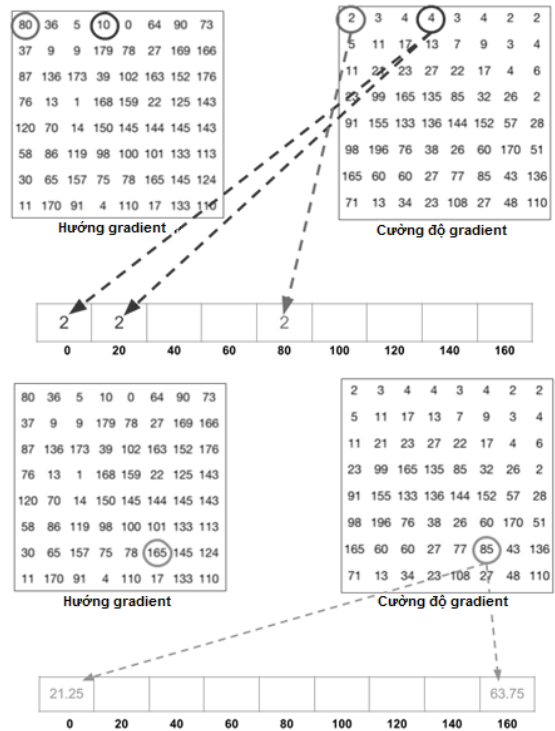
Hình 7. Ảnh trái: ảnh gradient theo chiều  $x$ . Ảnh giữa: ảnh gradient theo chiều  $y$ . Ảnh phải: ảnh cường độ gradient

**Bước 3: Tính Histogram các gradient trong các cell 8x8:** Chia ảnh thành các cell 8x8. Đối với ảnh màu, nếu mỗi pixel có 3 giá trị màu, thì mỗi cell 8x8 pixel sẽ có  $8 \times 8 \times 3 = 192$  giá trị. Nếu dùng gradient,

mỗi cell 8x8 chỉ có  $8 \times 8 \times 2 = 128$  giá trị. 128 giá trị này lại được biểu diễn bằng một vector 9 bins (9 số) tương ứng với 9 góc 0, 20, 40, 60, ... 160 tượng trưng cho hướng gradient. Như vậy nếu biểu diễn một ảnh bằng gradient sẽ giảm được số dữ liệu cần tính toán.



Hình 8. Chia ảnh thành nhiều cell 8x8 và tính các đại lượng gradient cho các cell này

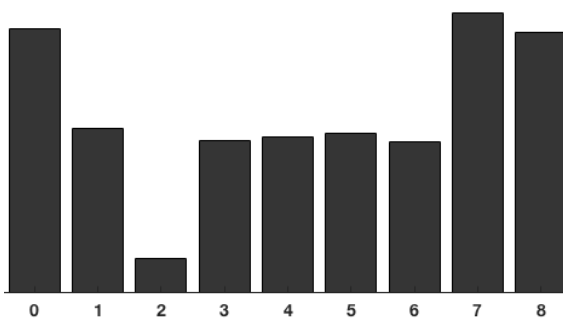


Hình 9. Histogram gradients

Hình 9 minh họa cách biểu diễn histogram các gradients dựa trên cường độ và hướng gradient các pixels trong một cell 8x8. Pixel ở vị trí  $a_{11}$  (khoanh tròn xanh

ương) có giá trị góc  $\theta = 80$  (hướng gradient), giá trị cường độ gradient  $g = 2$ , ta đưa giá trị  $g = 2$  vào bin80 trong vector histogram. Pixel ở vị trí  $a_{14}$  (khoanh tròn đỏ) có giá trị góc  $\theta = 10$ , giá trị cường độ gradient  $g = 4$ , 10 nằm giữa bin0 và bin 20, khoảng cách từ  $0 \rightarrow 10 = 10$ , tỉ lệ  $10/20 = 0.5$ , khoảng cách từ  $10 \rightarrow 20 = 10$ , tỉ lệ  $10/20 = 0.5$  nên giá trị cường độ gradient  $g = 4$  phân bố đều cho hai bin 0 và bin 20, mỗi bin có giá trị  $4 \times 0.5 = 2$ . Pixel ở vị trí  $a_{76}$  (khoanh tròn xanh lá) có giá trị góc  $\theta = 165$ , cường độ gradient  $g = 85$ . Trong histogram 9 bins góc  $0^\circ \sim 180^\circ$  nên góc  $165^\circ$  nằm giữa bin160 và bin0, khoảng cách từ  $160 \rightarrow 165 = 5$ , tỉ lệ  $5/20 = 0.25$ , khoảng cách từ  $165 \rightarrow 180 (\sim 0) = 15$ , tỉ lệ  $15/20 = 0.75$ . Do khoảng cách từ  $165 \rightarrow 160$  gần hơn nên trọng số phân bố vào bin160 sẽ là 0.75, trọng số phân bố vào bin0 sẽ là 0.25. Bin160 sẽ nhận giá trị  $85 \times 0.75 = 63.75$ , bin0 nhận giá trị  $85 \times 0.25 = 21.25$ . Ở bước trước bin0 đã có giá trị là 2, nên bin0 sẽ có giá trị là  $2 + 21.25 = 23.25$ .

Lần lượt tính cho tất cả các ô trong cell trên, chúng ta được biểu đồ histogram như sau:



Hình 10. Biểu đồ histogram

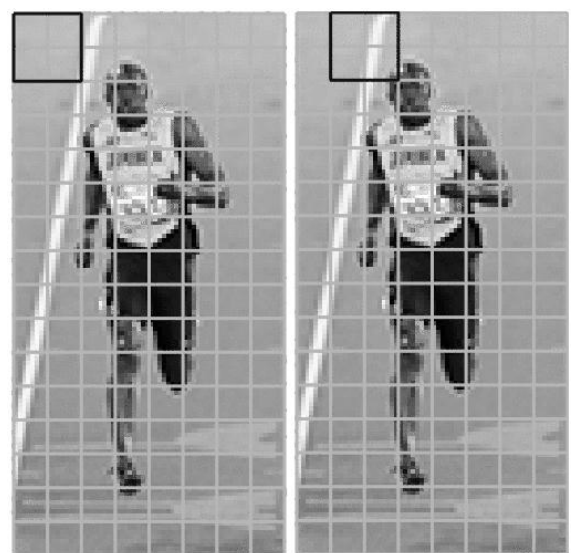
**Bước 4: Chuẩn hóa khối 16x16:** Gradient của một ảnh rất nhạy cảm với các vùng có

ánh sáng chồng lấn nhất là ảnh thiếu sáng, do đó để giảm độ ảnh hưởng của yếu tố ánh sáng lên gradient (đồng nghĩa với giảm nhiễu) ta cần chuẩn hóa gradient. Xét một ví dụ về chuẩn hóa như sau: giả sử một pixel có giá trị màu RGB là  $[128, 64, 32]$ . Chiều dài vector trên sẽ là:  $\sqrt{128^2 + 64^2 + 32^2} = 146.64$ , vector được chuẩn hóa L2 sẽ là  $\left[\frac{128}{146.64}, \frac{64}{146.64}, \frac{32}{146.64}\right] = [0.87, 0.43, 0.22]$

Trong bước này, thay vì chuẩn hóa vector histogram của từng cell 8x8, chúng ta sẽ chuẩn hóa vector histogram cho từng khối 16x16, mỗi khối sẽ có 4 cell 8x8, tức sẽ có sự chồng lấn giữa các khối. Một cell 8x8 có vector histogram kích thước 9x1, do đó một khối 16x16 sẽ có vector histogram kích thước 36x1.

Lần lượt trượt cửa sổ 16x16 và tính vector histogram cho từng khối từ trái sang phải và từ trên xuống dưới, mỗi bước trượt 8 pixel cho đến khi hết hình.

Hình 11 minh họa một bước trượt cửa sổ khối 16x16.



Hình 11. Cửa sổ trượt 16x16

### Tính vector đặc trưng HOG

Gộp tất cả vector 36x1 vào thành một vector lớn hơn có kích thước: Có 7 vị trí của số trượt theo chiều ngang và 15 vị trí của số trượt theo chiều dọc, tổng cộng có  $7 \times 15 = 105$  vị trí; Mỗi khối 16x16 được biểu diễn bằng một vector 36x1, nếu gộp 105 vector 36x1 vào thành một vector lớn hơn sẽ được vector có kích thước  $36 \times 105 = 3780 \times 1$ .



Hình 12. Biểu diễn vector Hog của các cell 8x8

Hình 12 minh họa biểu diễn vector đặc trưng Hog 9x1 đã được chuẩn hóa của các cell 8x8. Nhận thấy quanh vùng thân và chân của vận động viên, hướng chủ đạo của các vector xoay bao quanh vùng này.

## 3. KẾT QUẢ MỘT SỐ THỰC NGHIỆM

### 3.1. Thực nghiệm kiểm tra độ chính xác thuật toán

**Thực nghiệm 1:** Thuật toán: KNN; Dữ liệu: Opencv-digits. 50% dữ liệu được dùng làm tập huấn luyện, 50% dữ liệu được dùng làm tập test; Đặc trưng: đặc trưng thô; Kết quả: độ chính xác 91.76%.

**Thực nghiệm 2:** Thuật toán: KNN; Dữ liệu: Opencv-digits. 50% dữ liệu được dùng làm tập huấn luyện, 50% dữ liệu được dùng

làm tập test; Đặc trưng: đặc trưng Hog; Kết quả: độ chính xác 92.88%.

**Thực nghiệm 3:** Thuật toán: SVM; Dữ liệu: Opencv-digits. 50% dữ liệu được dùng làm tập huấn luyện, 50% dữ liệu được dùng làm tập test; Đặc trưng: đặc trưng thô; Kết quả: độ chính xác 90.44%.

**Thực nghiệm 4:** Thuật toán: SVM; Dữ liệu: Opencv-digits. 50% dữ liệu được dùng làm tập huấn luyện, 50% dữ liệu được dùng làm tập test; Đặc trưng: đặc trưng Hog; Kết quả: độ chính xác 93.8%.

### 3.2. Ứng dụng nhận diện số báo danh trên phiếu trả lời trắc nghiệm

Trên Phiếu trả lời trắc nghiệm, phần ghi thông tin số báo danh của thí sinh gồm 2 phần: phần đánh số báo danh bằng cách tô các chấm tròn trên các chữ số và phần ghi số báo danh bằng tay. Hình 13 minh họa một mẫu Phiếu trả lời trắc nghiệm.

**PHIẾU TRẢ LỜI TRẮC NGHIỆM**

Giám thị 1	1. Tên trường: <b>Trường THPT Lê Quý Đôn</b>	9. Số báo danh: <b>0169954</b>	10. Mã đề thi: <b>22A</b>		
Họ tên: .....	2. Điểm thi: .....	<div style="display: flex; flex-wrap: wrap;"> <div style="width: 50%;"> <p>3. Phòng thi: <b>Phòng 1</b></p> <p>4. Họ và tên thí sinh: <b>Nguyễn Hoàng Phúc</b></p> <p>5. Ngày sinh: <b>20/10/2000</b></p> <p>6. Chữ ký: <b>Có chữ ký</b></p> <p>7. Môn thi: <b>Có số đề thi</b></p> <p>8. Ngày thi: <b>5/10/2016</b></p> </div> <div style="width: 50%;"> <p>1. <input type="radio"/> 2. <input type="radio"/> 3. <input type="radio"/> 4. <input type="radio"/> 5. <input type="radio"/> 6. <input type="radio"/> 7. <input type="radio"/> 8. <input type="radio"/> 9. <input type="radio"/> 10. <input type="radio"/> 11. <input type="radio"/> 12. <input type="radio"/> 13. <input type="radio"/> 14. <input type="radio"/> 15. <input type="radio"/> 16. <input type="radio"/> 17. <input type="radio"/> 18. <input type="radio"/> 19. <input type="radio"/> 20. <input type="radio"/> 21. <input type="radio"/> 22. <input type="radio"/> 23. <input type="radio"/> 24. <input type="radio"/> 25. <input type="radio"/> 26. <input type="radio"/> 27. <input type="radio"/> 28. <input type="radio"/> 29. <input type="radio"/> 30. <input type="radio"/> 31. <input type="radio"/> 32. <input type="radio"/> 33. <input type="radio"/> 34. <input type="radio"/> 35. <input type="radio"/> 36. <input type="radio"/> 37. <input type="radio"/> 38. <input type="radio"/> 39. <input type="radio"/> 40. <input type="radio"/> 41. <input type="radio"/> 42. <input type="radio"/> 43. <input type="radio"/> 44. <input type="radio"/> 45. <input type="radio"/> 46. <input type="radio"/> 47. <input type="radio"/> 48. <input type="radio"/> 49. <input type="radio"/> 50. <input type="radio"/> 51. <input type="radio"/> 52. <input type="radio"/> 53. <input type="radio"/> 54. <input type="radio"/> 55. <input type="radio"/> 56. <input type="radio"/> 57. <input type="radio"/> 58. <input type="radio"/> 59. <input type="radio"/> 60. <input type="radio"/></p> </div> </div>			
Giám thị 2					
Họ tên: .....					
Chữ ký: .....					

**Thí sinh lưu ý:**

- Giữ cho phiếu phẳng, không bôi bẩn, làm rách.
- Phải ghi đầy đủ các mục theo hướng dẫn.
- Dùng bút chì đen tô kín các ô trong mục: Số báo danh, mã đề thi.
- Ghi mã đề thi vào phần bài làm để chấm thủ công.

**Phần trả lời:** Số thứ tự câu trả lời dưới đây ứng với số thứ tự câu trắc nghiệm trong đề thi. Đốt với mỗi câu trắc nghiệm thí sinh chọn và tô kín một ô tròn tương ứng với phương án trả lời đúng.

Mã đề thi: .....	1. <input type="radio"/> 2. <input type="radio"/> 3. <input type="radio"/> 4. <input type="radio"/> 5. <input type="radio"/> 6. <input type="radio"/> 7. <input type="radio"/> 8. <input type="radio"/> 9. <input type="radio"/> 10. <input type="radio"/> 11. <input type="radio"/> 12. <input type="radio"/> 13. <input type="radio"/> 14. <input type="radio"/> 15. <input type="radio"/> 16. <input type="radio"/> 17. <input type="radio"/> 18. <input type="radio"/> 19. <input type="radio"/> 20. <input type="radio"/> 21. <input type="radio"/> 22. <input type="radio"/> 23. <input type="radio"/> 24. <input type="radio"/> 25. <input type="radio"/> 26. <input type="radio"/> 27. <input type="radio"/> 28. <input type="radio"/> 29. <input type="radio"/> 30. <input type="radio"/> 31. <input type="radio"/> 32. <input type="radio"/> 33. <input type="radio"/> 34. <input type="radio"/> 35. <input type="radio"/> 36. <input type="radio"/> 37. <input type="radio"/> 38. <input type="radio"/> 39. <input type="radio"/> 40. <input type="radio"/> 41. <input type="radio"/> 42. <input type="radio"/> 43. <input type="radio"/> 44. <input type="radio"/> 45. <input type="radio"/> 46. <input type="radio"/> 47. <input type="radio"/> 48. <input type="radio"/> 49. <input type="radio"/> 50. <input type="radio"/> 51. <input type="radio"/> 52. <input type="radio"/> 53. <input type="radio"/> 54. <input type="radio"/> 55. <input type="radio"/> 56. <input type="radio"/> 57. <input type="radio"/> 58. <input type="radio"/> 59. <input type="radio"/> 60. <input type="radio"/>
------------------	--

Hình 13. Mẫu Phiếu trả lời trắc nghiệm

Mục đích xây dựng ứng dụng là nhằm loại bỏ các Phiếu trả lời trắc nghiệm không hợp lệ do số báo danh được ghi bằng phương pháp chấm tròn không khớp với số báo danh được ghi bằng cách viết tay.

**Dữ liệu:** Tập dữ liệu MNIST; **Đặc trưng:** Đặc trưng HOG; **Thuật toán:** Thuật toán máy học: SVM, thuật toán triển khai ứng dụng: Thuật toán 3.1

**Thuật toán 3.1:** *Input:* ảnh scan Phiếu trả lời trắc nghiệm; *Output:* tập các số báo danh đã được nhận diện: 1). rois = các chữ số báo danh viết tay trên Phiếu trả lời trắc nghiệm; 2). for each roi in rois; 3). Chuyển roi sang ảnh grayscale; 4). Xác định các vùng contour trong roi; 5). Loại các contour nhiễu, chỉ lấy contour có diện tích lớn nhất; 6). Scaling contour dựa trên kích thước contour; 7). Resize contour 28x28; 8). Tính vector Hog cho contour; 9). Nhận diện contour bằng thuật toán SVM; 10). end for.

### 3.2.4. Một số kết quả



Hình 14. Một số kết quả thực nghiệm

### 3.2.5. Phân tích một số nguyên nhân ảnh hưởng đến kết quả.

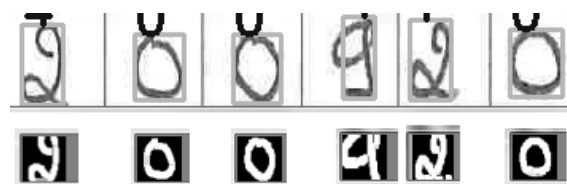
*Sự khác biệt về hình dáng chữ viết của người Việt Nam và người phương Tây*

Ví dụ, trong dữ liệu Opencv, đa số số 1 thường được viết chỉ bằng một nét đi xuống **1**, người Việt Nam thường có cách viết số 1 có chân và một nét đi lên trước khi bắt đầu nét đi xuống **1**. Theo thống kê trong bộ dữ liệu Opencv, số **1** chiếm tỉ lệ 479/500, số **1** chiếm tỉ lệ 21/500. Một trường hợp khác là số 7, trong khi số **7** chiếm 435/500, số **7** chỉ chiếm 65/500.

*Chữ viết nguệch ngoạc, cách điệu*

Ví dụ: **8**, **3**

*Quá trình xác định contour chưa chính xác:* Contour hiểu một cách đơn giản là đường nối dọc theo biên tất cả các điểm liên tục cùng màu và cùng cường độ. Hình 15 minh họa kết quả xác định contour số báo danh của một Phiếu trả lời trắc nghiệm. Đường viền xanh trong ảnh trên là contour các chữ số xác định được, ảnh bên dưới là contour đã được cắt ra và chuyển sang ảnh grayscale. Một số chữ số sau khi được lấy ra từ kết quả xác định contour đã bị mất một số nét ở biên.



Hình 15. Kết quả cắt chữ số từ contour

*Nét chữ không đều:* Nét chữ khi viết không đều nhất là ở các nét cuối. Nét không đều dẫn đến kết quả chuyển sang ảnh grayscale bị mất một số nét hoặc nét

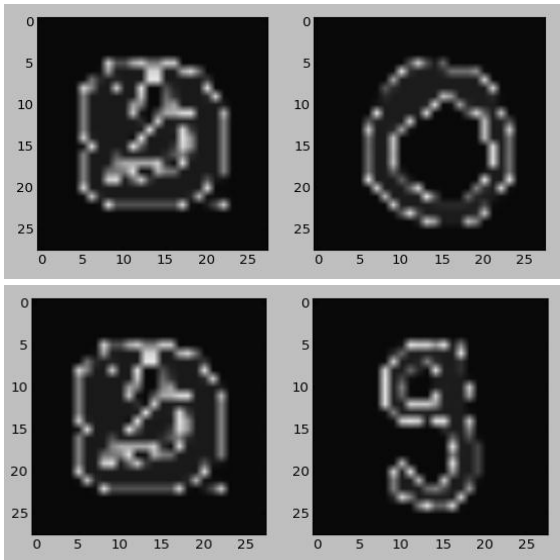


đậm nhạt khác nhau. Hình 16 minh họa kết quả chuyển sang ảnh grayscale của một số chữ số có nét không đều.



**Hình 16.** Ảnh grayscale nét chữ không đều

Hình 17 cho thấy, số 9 có nét không đều ở trên dễ nhận diện thành số 0 hơn là nhận diện số 9.



**Hình 17.** Ảnh phóng to so sánh chữ số 9 nhận được từ contour (trái) và một mẫu chữ số 0 và 9 từ bộ dữ liệu MNIST (phải)

#### 4. KẾT LUẬN

Nhận diện chữ số viết tay là một tiến trình phụ thuộc vào nhiều nhóm yếu tố như:

*Nhóm yếu tố kỹ thuật:* Thuật toán máy học; Thuật toán xử lý ảnh.

*Nhóm yếu tố phi kỹ thuật:* Chất lượng dữ liệu phục vụ cho máy học; Chất lượng mẫu cần nhận diện.

Ứng dụng nhận diện chữ số báo danh đòi hỏi độ chính xác 100% là một bài toán khó. Có thể nâng cao độ chính xác ứng dụng thông qua nhóm yếu tố kỹ thuật như: điều chỉnh các tham số thuật toán, cải tiến thuật toán hoặc phát triển các thuật toán mới.

Đối với nhóm yếu tố phi kỹ thuật, có thể cải thiện như: Dữ liệu phục vụ máy học: ngoài các bộ dữ liệu đã nêu ở trên, có thể bổ sung hoặc xây dựng bộ dữ liệu riêng phù hợp với hình dáng chữ số viết tay của người Việt Nam; Đối với việc cải thiện chất lượng mẫu cần nhận diện có thể: sử dụng viết chì chất lượng cao dùng để viết số báo danh; Phiếu trả lời trắc nghiệm không được nhẵn để làm mất nét ở các nếp nhăn,...

#### TÀI LIỆU THAM KHẢO

1. Reza Ebrahimzadeh, Mahdi Jampour (2014), *Efficient Handwritten Digit Recognition based on Histogram of Oriented Gradients and SVM*, International Journal of Computer Applications (0975 – 8887) Volume 104 – No.9.
2. Grace Tsai (2010), *Histogram of Oriented Gradients*, University of Michigan.
3. Navneet Dalal, Bill Triggs (2005), *Histograms of Oriented Gradients for Human Detection*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition.
4. MNIST, *Handwritten Digits database of New York University*, <http://www.cs.nyu.edu/~roweis/data.html>.



5. *Histogram of Oriented Gradients*, <https://www.learnopencv.com/histogram-of-oriented-gradients>.
6. *OCR of Hand-written Data using kNN*, [http://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_ml/py\\_knn/py\\_knn\\_opencv/py\\_knn\\_opencv.html](http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_ml/py_knn/py_knn_opencv/py_knn_opencv.html).
7. *OCR of Hand-written Data using SVM*, [http://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_ml/py\\_svm/py\\_svm\\_opencv/py\\_svm\\_opencv.html#svm-opencv](http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_ml/py_svm/py_svm_opencv/py_svm_opencv.html#svm-opencv).

Ngày nhận bài: 25/04/2017. Ngày biên tập xong: 8/8/2017. Duyệt đăng: 20/8/2017