

# NHẬN DẠNG CHỮ SỐ VIẾT TAY CHO ỨNG DỤNG NHẬP ĐIỂM TỰ ĐỘNG

Dang Thanh Tung<sup>1</sup>, Truong Cong Duc Uy<sup>2</sup>, Nguyen Quang Hao<sup>3</sup>  
Duong Thi My Duyen<sup>4</sup>, Trinh Ha Cam Tu<sup>5</sup>, Nguyen Van An<sup>6</sup>  
Electrical and Electronic Department  
Ho Chi Minh city-University of Technology

## Tóm Tắt

Trong bài báo này, nhóm đã nghiên cứu cách nhận dạng chữ số viết tay phục vụ cho việc nhập điểm tự động lưu vào cơ sở dữ liệu. Đề tài sử dụng một bảng điểm mẫu có các thông tin cần nhận dạng là mã số sinh viên (MSSV), điểm tổng và điểm rubric. Sử dụng các đặc trưng về đường bao (contour) để cắt bảng điểm ra từ phiếu điểm, từ đó áp dụng các kỹ thuật xử lý ảnh để tách từng chữ số của mssv, điểm tổng ra và nhận dạng điểm rubric. Các chữ số sau đó được sử dụng một thuật toán machine learning là SVM (Support Vector Machine) để nhận dạng. Kết quả cho thấy tỉ lệ nhận dạng thành công đạt trên 97%.

**Từ khoá** *Contour, SVM, HOG, Adaptive Thresholding, Dilation, Erosion, MNIST*.

## I. Giới thiệu

Học máy (Machine Learning) là một lĩnh vực đang được nhắc đến khá nhiều trong thời gian gần đây bởi tính ứng dụng của nó trong thực tiễn. Machine learning (ML) là một phương pháp phân tích dữ liệu mà sẽ tự động hóa việc xây dựng mô hình phân tích. Sử dụng các thuật toán lập để học từ dữ liệu, machine learning cho phép máy tính tìm thấy những thông tin giá trị ẩn sâu mà không được lập trình một cách rõ ràng nơi để tìm. Những công nghệ áp dụng ML có thể dễ dàng thấy như: google dịch, xe ô tô tự lái, hệ thống gợi ý mua hàng, hệ thống nhận diện khuôn mặt của facebook,... Điều này cho thấy ML đang đóng vai trò quan trọng trong sự phát triển của công nghệ tương lai.

Đề tài nhóm thực hiện là nhận dạng bảng điểm, bên cạnh việc xử lý ảnh thì kết quả đầu ra phụ thuộc vào sự nhận dạng chữ số viết tay. Chữ viết tay (gồm các số 0 - 9) phụ thuộc vào cách viết mỗi người nên khó có thể nhận dạng theo cách lập trình thông thường. Tuy nhiên dù cho mỗi người viết khác nhau nhưng các chữ số vẫn có các đặc trưng khác riêng biệt nhau, do đó có thể dùng ML để nhận dạng. Nhóm sử dụng thuật toán ML là SVM để huấn luyện nhận dạng các số dựa vào đặc trưng HOG của các ảnh số. Tập dữ liệu huấn luyện được dùng là MNIST, đây là một tập dữ liệu chữ số viết tay từ 0 đến 9 bao gồm 60000 mẫu huấn luyện. Cách phân tiếp theo của bài báo gồm có phần II cho cái nhìn tổng quan về mô hình và quá trình xử lý, phần III nêu lên các phương pháp xử lý ảnh sử dụng, phần IV nói về phương pháp nhận dạng SVM, phần V trình bày kết quả đạt được và phần VI là kết luận của đề tài.

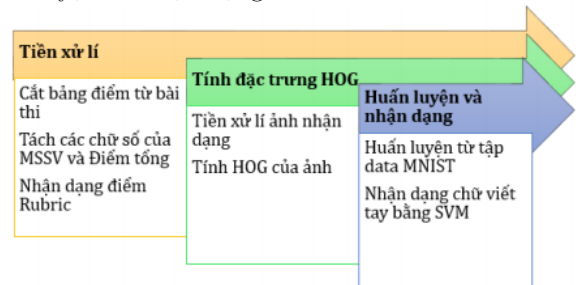
## II. Mô hình đề tài

Đề tài được thực hiện dưới dạng một phần mềm (software) với giao diện được viết bằng PyQt5. Phần mềm sẽ kết nối với một IP camera trong mạng cục bộ, điều khiển camera truyền hình ảnh về. Khi nào muốn xử lý thì ta nhấn nút để lấy hình ảnh về, xử lý và xuất dữ liệu nhận dạng ra bảng trên giao diện và dùng nó để lưu ra một file excel mới hoặc cập nhật vào file excel cũ. Bảng điểm để nhận dạng có dạng như sau:

Họ và tên SV:	MSSV:				
Cán bộ coi thi	GV tổng hợp đề				
					BM Viễn thông
Câu 1 (1,5đ)					
Câu 2 (1,5đ)					
Câu 3 (3đ)	1	2	3	4	5
Câu 4 (2đ)	1	2	3	4	5
Câu 5 (2đ)	1	2	3	4	5
					Điểm Tổng

Hình 1: Bảng điểm theo yêu cầu nhận dạng của đề tài

Gồm 3 giai đoạn: tiền xử lý, trích đặc trưng HOG, huấn luyện và nhận dạng.



Hình 2: Mô hình đề tài

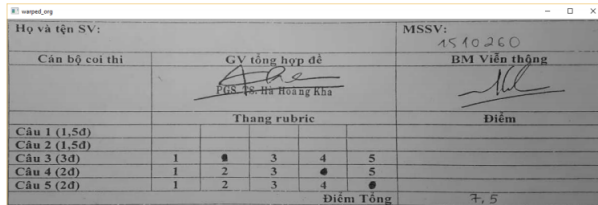
## III. Phương pháp tiền xử lý

### 1. Tìm bảng điểm trên ảnh đầu vào

Ảnh đầu vào sẽ là ảnh phiếu điểm được chụp lại với những góc nghiêng khác nhau. Do những thông tin ta cần lấy đều nằm trên bảng điểm nên ta phải xác định nó và cắt nó ra khỏi ảnh đầu vào để xử lý. Các bước thực hiện:

- Tìm contour (đường bao) cho ảnh đầu vào.
- Xấp xỉ contour và chọn contour lớn nhất trong ảnh mà có hình dạng là hình chữ nhật—đó chính là bảng cần tìm.

- Tìm tọa độ 4 góc của bảng vừa tìm được.
- Dựa vào 4 góc, ta dùng phép biến đổi góc nhìn (Perspective Warping) để biến đổi góc nghiêng của bảng thành góc thẳng đứng (Bird's-eye view), đây là ảnh dùng để xử lý tiếp theo.



Hình 3: Ảnh bảng điểm được cắt ra và biến đổi góc nhìn "bird's-eye view"

## 2. Tách MSSV và điểm tổng từ bảng điểm

Sau khi có ảnh bảng điểm phẳng, ta sẽ tách các thông tin từ trong đó. Cách thực hiện là khi biến đổi ảnh phẳng ta sẽ có được thông tin chiều dài rộng (pixel) của bức ảnh và dựa vào độ dài rộng (cm) thực của bảng, ta tính tỉ lệ để xác định tọa độ của MSSV và Điểm tổng, từ đó cắt chúng ra thành 2 ảnh con. Sử dụng các phép hình thái học và lấy contour ta sẽ lọc bỏ được các phần không mong muốn và nhiều trong ảnh cắt được.

Từ 2 ảnh con sau khi lọc, ta sẽ tìm contour để tách các chữ số ra riêng, mỗi chữ số sẽ là một contour. Dùng các điều kiện lọc để chỉ giữ lại các chữ số, dùng phương pháp sort contour để sắp xếp các chữ số được cắt ra theo đúng thứ tự của nó. Các chữ số của MSSV và Điểm tổng được tách ra sử dụng cho quá trình nhận dạng tiếp theo.



Hình 4: Quá trình xử lý tách các chữ số của mã sinh viên

## 3. Nhận dạng điểm Rubic

Điểm Rubric được nhận dạng dựa vào đặc điểm ô nào được tô điểm rubric thì ô đó sẽ có diện tích contour lớn nhất.

Đầu tiên dựa vào ảnh bảng phẳng, tỉ lệ cắt ra từng ô điểm rubric. Sau đó tìm contour cho từng ô và tính diện tích của chúng. Ô nào có diện tích contour lớn nhất là ô đó được tô điểm.

## IV. Nhận dạng chữ số viết tay

### 1. Đặc trưng HOG (Histogram of Oriented Gradients)

Histogram of Oriented Gradients (HOG) là một phương pháp trích đặc trưng của ảnh được ứng dụng trong thị giác máy và xử lý ảnh cho mục đích nhận dạng đối tượng. Ảnh được chia thành các cửa sổ

nhỏ hơn gọi là các cell, HOG được tính ở mỗi pixel trong cell, các cell được gộp lại thành các block có thể chồng lấn lên nhau (overlapping). Phương pháp HOG tương tự các phương pháp như edge orientation histograms, SIFT, và shape contexts nhưng HOG được tính toán trên một lưới dày đặc các cell và sử dụng cách chuẩn hóa overlapping local contrast normalizations giữa các block để nâng cao hiệu quả. Trong phương pháp này, đặc trưng được trích ra từ ảnh là sự phân bố hướng của các gradient. Ý tưởng cơ bản của phương pháp là sự hiện diện và hình dạng của vật thể có thể được mô tả bằng hướng và độ lớn của gradient. Các bước thực hiện phương pháp HOG:

- Chuẩn hóa màu của ảnh : Navneet Dalal và Bill Triggs sử dụng power law (gamma) equalization để thực hiện chuẩn hóa và nhận thấy rằng bước này ảnh hưởng nhỏ đến hiệu suất
- Tính độ lớn và hướng của các gradients
  - Tính các gradient theo chiều ngang  $g_x$  và chiều dọc  $g_y$  bằng cách filter ảnh với kernel  $[-1,0,1]$  và  $[1,0,-1]^T$
  - Tính độ lớn và hướng của các gradient theo công thức:

$$g = \sqrt{g_x^2 + g_y^2}$$

$$\theta = \arctan \frac{g_y}{g_x}$$

- Thực hiện Orientation binning: tính HOG của từng cell, chia đều thành 9 histogram channel từ  $0^\circ$  đến  $180^\circ$  (unsigned gradient). Mỗi pixel trong cell sẽ bổ sung trọng số vào channel của histogram tùy theo hướng gradient của pixel đó. Trọng số ở đây có thể là độ lớn gradient, bình phương độ lớn gradient hoặc căn bậc hai độ lớn gradient và thực nghiệm cho thấy rằng sử dụng trọng số là độ lớn gradient cho kết quả tốt nhất.
- Chuẩn hóa các block: Navneet Dalal và Bill Triggs đưa ra 4 cách chuẩn hóa block. Đặt  $v$  là vector chưa được chuẩn hóa chứa tất cả các histogram của một block,  $\|v\|_k$  là chuẩn của  $v$  với  $k = 1, 2$ .

$$L2 - norm : f = \frac{v}{\sqrt{e^2 + \|v\|_2^2}}$$

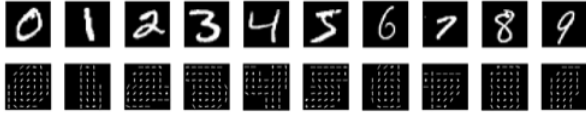
$$L1 - norm : f = \frac{v}{e + \|v\|_1}$$

$$L1 - sqrt : f = \sqrt{\frac{v}{e + \|v\|_1}}$$

L2-Hys: là L2-norm bị xén (giới hạn giá trị lớn nhất của  $v$  là 0.2) rồi chuẩn hóa lại.

- Gộp lại thành một feature vector chứa các HOG đã tính.  
Ứng dụng HOG để trích đặc trưng hình ảnh chữ số của đề tài: trích đặc trưng và hiển thị ảnh HOG của 10 chữ số trong

dataset của MNIST với các thông số như sau: Cell Size 4x4, Block Size 2x2, Block Overlap 50%, Number of Bin 9, Unsigned Orientation.



Hình 5: Đặc trưng HOG của các ảnh chữ số viết tay

## 2. Thuật toán SVM (Support Vector Machine)

Support vector machine (SVM) là các mô hình học được giám sát với thuật toán học tập liên quan phân tích dữ liệu được sử dụng để phân tích và phân tích hồi quy. SVM dạng chuẩn nhận dữ liệu vào và phân loại chúng vào hai lớp khác nhau.

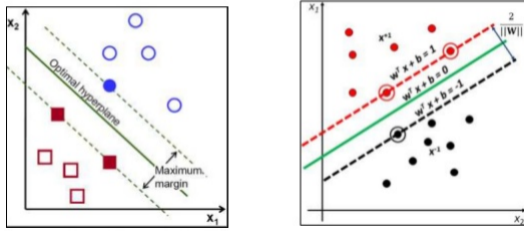
⇒ SVM là 1 loại thuật toán dạng nhị phân.

Trong yêu cầu nhận dạng chữ viết tay thì có 10 nhóm cần được nhận dạng (số 0, 1,...9).

⇒ Sử dụng Multiclass SVM One Vs One.

Ta tìm hiểu dạng chuẩn SVM nhị phân : Kernel Linear

### SVM



Hình 6: Siêu phẳng tuyến tính và siêu phẳng với lề cực đại.

Như ta thấy nếu khoảng cách tối thiểu của cả 2 nhóm đến siêu phẳng  $z$  càng lớn thì xác suất phân loại nhóm càng chính xác.

⇒ Xác định  $(z) = \omega^T \cdot x + b = 0$  sao cho khoảng cách từ  $x^{+1}$  đến  $x^{-1}$  đến  $(z)$  lớn nhất.

Chọn siêu phẳng tối ưu sao cho khoảng cách tối thiểu đến 2 nhóm là bằng nhau.

⇒ Chọn điểm thuộc nhóm 1 thỏa  $(z_1) : \omega^T \cdot x + b \geq 1$

Chọn điểm thuộc nhóm 2 thỏa  $(z_2) : \omega^T \cdot x + b \leq -1$

Hay  $y_i(\omega^T \cdot x_i + b) \geq 1$  with  $y_i = \pm 1$

Khoảng cách từ điểm đến siêu phẳng được xác định bằng công thức sau:

$$d = \frac{|\omega^T \cdot x + b|}{\sqrt{(\omega^T)^2}} = \frac{|\omega^T \cdot x + b|}{\|\omega\|}$$

$$\Rightarrow \begin{cases} d_{x^{+1}} = \frac{|\omega^T \cdot x^{+1} + b|}{\|\omega\|} = \frac{|1 - b + b|}{\|\omega\|} = \frac{1}{\|\omega\|} \\ d_{x^{-1}} = \frac{|\omega^T \cdot x^{-1} + b|}{\|\omega\|} = \frac{|-1 - b + b|}{\|\omega\|} = \frac{1}{\|\omega\|} \end{cases}$$

$$\Rightarrow \text{margin} = d_{x^{+1}} + d_{x^{-1}} = \frac{1}{\|\omega\|} + \frac{1}{\|\omega\|} = \frac{2}{\|\omega\|}$$

$$\Rightarrow \text{marginmax} \Leftrightarrow \frac{2}{\|\omega\|} \text{max} \Leftrightarrow \|\omega\| \text{min} \Leftrightarrow$$

$$\frac{1}{2} \|\omega\|^2 \text{min}$$

Dùng phương pháp nhân tử Lagrange cực tiểu hàm

$$L(\omega, b, \lambda) = \frac{1}{2} \|\omega\|^2 + \sum_{i=1}^N \lambda_i (1 - y_i(\omega^T \cdot x_i + b))$$

Với điều kiện ràng buộc :  $y_i(\omega^T \cdot x_i + b) \geq 1$  with  $y_i = \pm 1$

$$\frac{\partial L}{\partial \omega} = \omega - \sum_{i=1}^N \lambda_i y_i x_i = 0 \Leftrightarrow \omega = \sum_{i=1}^N \lambda_i y_i x_i$$

$$\frac{\partial L}{\partial b} = - \sum_{i=1}^N \lambda_i y_i = 0 \Leftrightarrow \sum_{i=1}^N \lambda_i y_i = 0$$

Khi thỏa mãn điều kiện trên thì :  $y_i(\omega^T \cdot x_i + b) = 1 \Rightarrow b$

Từ các điều kiện trên ta được hàm Lagrange:

$$L(\omega, b, \lambda) = \frac{1}{2} \|\omega\|^2 + \sum_{i=1}^N \lambda_i - \sum_{i=1}^N \lambda_i y_i \omega^T x_i + \sum_{i=1}^N \lambda_i y_i b$$

$$= \frac{1}{2} \|\omega\|^2 + \sum_{i=1}^N \lambda_i - \omega \omega^T = \sum_{i=1}^N \lambda_i - \frac{1}{2} \|\omega\|^2$$

$$= \sum_{i=1}^N \lambda_i - \sum_{i=1}^N \lambda_i y_i x_i \omega^T \lambda_j y_j x_j \omega^T$$

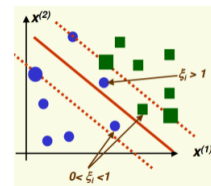
Cực tiểu hàm này với điều kiện ràng buộc :

$$\begin{cases} \lambda_i \geq 0 \\ \sum_{i=1}^N \lambda_i y_i = 0 \end{cases}$$

Từ đó tìm được các  $\lambda_{i,j}$  tương ứng  $\Rightarrow \begin{cases} \omega \\ b \end{cases} \Rightarrow$

Siêu phẳng cần tìm.

Đối với trường hợp các dữ liệu xâm lấn lẫn nhau, ta có thể phân lớp bằng hàm tuyến tính bằng cách hi sinh vài điểm.



Hình 7: Siêu phẳng cho dữ liệu xâm lấn lẫn nhau.

Trong đó :

- $\xi_i = 0$  tương ứng phân lớp đúng.
- $0 < \xi_i < 1$  tương ứng phân lớp đúng nhưng điểm dữ liệu nằm gần đường phân lớp.
- $\xi_i > 1$  tương ứng phân lớp sai.

Khi đó hàm Lagrange :  $L(\omega, b, \lambda) = \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^N \xi_i$

Với ràng buộc :  $y_i(\omega^T \cdot x_i + b) \geq 1 - \xi_i$  with  $y_i = \pm 1$ ,  $\xi_i \geq 0$ , C là hằng số dương nhỏ phù hợp.

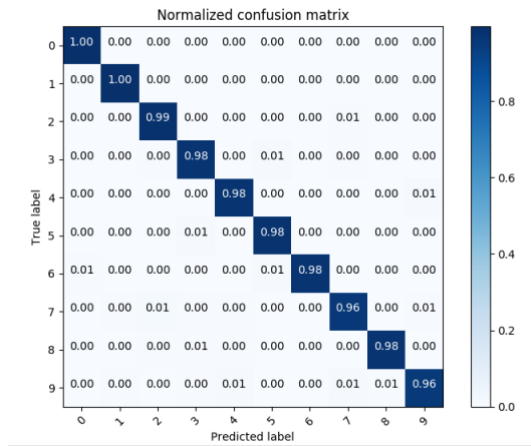
Đối với phân bố không tuyến tính ta dùng phương pháp Kernel Trick để chuyển đổi qua không

gian khác. Sau đó áp dụng SVM tuyến tính.

Bài toán nhận dạng chữ viết gồm 10 nhóm cần được phân loại là các chữ số 0, 1, 2, 3, ..., 9. Vì vậy ta sử dụng thuật toán MultiClass SVM ( cụ thể dùng One Vs One).

Ta sẽ chọn 2 nhóm trong trường hợp k lớp cần phân nhóm :  $C_k^2 = \frac{k(k-1)}{2}$

Trong bài toán ứng với k=10  $\Rightarrow$  cần 45 bộ phân lớp One Vs One. Sau đó áp dụng SVM nhị phân cho 45 bộ phân lớp One Vs One cho 10 nhóm dữ liệu.



Hình 9: Dữ liệu xuất ra file excel để lưu trữ.

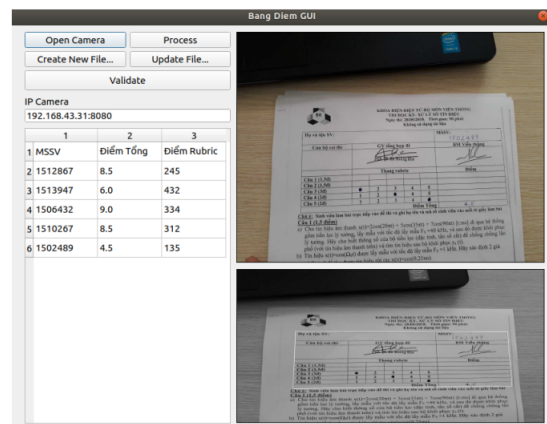
### 3. Tập dữ liệu chữ số viết tay MNIST

Tập dữ liệu chữ số viết tay MNIST bao gồm tập huấn luyện có 60000 mẫu và tập kiểm tra có 10000 mẫu. Đây là tập dữ liệu con của một tập dữ liệu lớn hơn của NIST. Tất cả các chữ số đã được chuẩn hóa kích thước và đưa về trọng tâm.

Tất cả ảnh trắng đen của NIST đều được chuẩn hóa cho vừa khung 28x28 pixel và vẫn đảm bảo giữ nguyên tỉ lệ khung hình. Các ảnh kết quả có mức xám do sử dụng kĩ thuật anti-aliasing trong giải thuật chuẩn hóa. Các ảnh được đưa về trọng tâm trong ảnh 28x28 pixel bằng cách tính trọng tâm và tịnh tiến.

### 2. Đánh giá mô hình HOG kết hợp SVM

Nhóm sử dụng phương pháp đánh giá mô hình là xây dựng Confusion matrix. Về cơ bản, confusion matrix thể hiện có bao nhiêu điểm dữ liệu thực sự thuộc một lớp, và được dự đoán rơi vào lớp. Cụ thể đối với đề tài, mô hình được đánh giá như sau: gồm 10 lớp (0 – 9), đường chéo màu xanh thể hiện tỉ lệ phần trăm phân loại đúng của mỗi lớp dữ liệu, phần khác đường chéo là tỉ lệ phần trăm phân loại sai của lớp dữ liệu này thành lớp dữ liệu khác.



Hình 10: Normalized confusion matrix được tính từ tập kiểm tra của MNIST

## V. Kết quả

### 1. Kết quả đạt được

Đã thiết kế được một phần mềm (software) để chạy tất cả các thuật toán. Nhận dạng thành công mã số sinh viên, điểm tổng và điểm rubric. Tỉ lệ nhận dạng đạt trên 97%. Dữ liệu sau khi nhận dạng được xuất ra file excel (\*.xls) để lưu lại.

	A	B	C	D	E	F	G
1	ST	MSSV	Điểm Tổng	Điểm Rubric			
2	1	1512867	8.5	245			
3	2	1513947	6	432			
4	3	1506432	9	334			
5	4	1510267	8.5	312			
6	5	1502489	4.5	135			

Hình 8: Giao diện phần mềm và kết quả nhận dạng.

## VI. Kết luận

Trong bài báo này, nhóm đã trình bày các phương pháp để nhận dạng thành công một bảng điểm theo mẫu được cho trước. Cụ thể là phải cắt được bảng điểm ra khỏi phiếu điểm trước, rồi dựa vào đó để xử lí nhận dạng điểm rubric và tách các chữ số viết tay ra để đưa vào bộ SVM. Tại đây, bộ SVM sau khi được huấn luyện sẽ nhận dạng chữ số viết tay với một tỉ lệ chính xác cao, đáp ứng được yêu cầu đề tài. Tuy nhiên nếu đối với những chữ viết tay có cách viết quá khác biệt, ta cần phải thu thập lại dữ liệu huấn luyện để ra kết quả chính xác hơn.

## Tham khảo

- [1] OpenCV dev team. “OpenCV-Python Tutorials”. Internet:  
[https://docs.opencv.org/3.0-beta/doc/py\\_tutorials/py\\_tutorials.html](https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_tutorials.html), Nov 10, 2014
- [2] Adrian Rosebrock. “4 Point OpenCV getPerspective Transform Example”. Internet:  
<https://www.pyimagesearch.com/2014/08/25/4-point-opencv-getperspective-transform-example>, August 25, 2014
- [3] Adrian Rosebrock. “Sorting Contours using Python and OpenCV”. Internet:  
<https://www.pyimagesearch.com/2015/04/20/sorting-contours-using-python-and-opencv/>, April 20, 2015
- [4] Dalal, Navneet, and Bill Triggs. “Histograms of oriented gradients for human detection.” Computer Vision and Pattern Recognition (CVPR), 2005.
- [5] R. Ebrahimzadeh and M. Jampour, "Efficient Handwritten Digit Recognition based on Histogram of Oriented Gradients and SVM," International Journal of Computer Applications (0975 – 8887), vol. 104, no. 9, pp. 10-13, Oct. 2014.
- [6] MathWorks, Inc. “Digit Classification Using HOG Features”. Internet:  
<https://www.mathworks.com/help/vision/examples/digit-classification-using-hog-features.html#d120e775>
- [7] Yann Lecun et al. “The MNIST database of handwritten digits”. Internet:  
<http://yann.lecun.com/exdb/mnist/>
- [8] Ole Kröger “Tensorflow, MNIST and your own handwritten digits”. Internet:  
<https://medium.com/@o.kroeger/tensorflow-mnist-and-your-own-handwritten-digits-4d1cd32bbab4>
- [9] B.S. Hanzra “Digit Recognition using OpenCV, sklearn and Python”. Internet:  
<http://hanzratech.in/2015/02/24/handwritten-digit-recognition-using-opencv-sklearn-and-python.html>, Feb 24, 2015
- [10] Pedregosa et al., “Scikit-learn: Machine Learning in Python,” JMLR 12, pp. 2825-2830, 2011
- [11] scikit-image dev team. “Module: feature”. Internet:  
<http://scikit-image.org/docs/dev/api/skimimage.feature.html>
- [12] Tieg Vu Huu “Bài 33: Các phương pháp đánh giá một hệ thống phân lớp”. Internet:  
<https://machinelearningcoban.com/2017/08/31/evaluation/-confusion-matrix>, Jan 3, 2018
- [13] William H et al (2007) The Art of Scientific Computing (3rd edition). Internet:  
<http://apps.nrbook.com/empanel/index.htmlpg=883>.
- [14] Patrick Winston “16. Learning: Support Vector Machines”. Internet:  
[https://www.youtube.com/watch?v=PwhiWxHK8ot=1566sfbclid=IwAR1qg9r\\_e-n-i9aqiRchEbJtZlSMtyuDn6a0SH2KJe-p-smqDYacJvEs2E](https://www.youtube.com/watch?v=PwhiWxHK8ot=1566sfbclid=IwAR1qg9r_e-n-i9aqiRchEbJtZlSMtyuDn6a0SH2KJe-p-smqDYacJvEs2E) , Jan 10, 2014.