**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**🙡🙜🙞🙣**

**ĐỒ ÁN CUỐI KHÓA MÔN HỌC**

**NHẬN DẠNG THỊ GIÁC VÀ ỨNG DỤNG**

**GIẢNG VIÊN HƯỚNG DẪN**

**Ts Lê Đình Duy – Ts Nguyễn Tấn Trần Minh Khang**

**HỌC VIÊN THỰC HIỆN : TRẦN VIỆT ANH – CH1502001**

**TP. HỒ CHÍ MINH, 2017**

NHẬN XÉT CỦA NGƯỜI HƯỚNG DẪN

# LỜI CẢM ƠN

Đầu tiên em chân thành cảm ơn đến Thầy Lê Đình Duy và Thầy Nguyễn Tấn Trần Minh Khang, những người đã trực tiếp hướng dẫn và tận tình chỉ bảo, truyền đạt kiến thức giúp em hoành thành đề tài đúng tiến độ. Các thầy đã cung cấp cho em những tài liệu rất quan trọng và quý báu, hướng dẫn và chỉ ra hướng đi đúng dắn cho em. Em xin được gửi tới các thầy lời cảm sâu sắc nhất.

Em cũng xin cảm ơn các anh chị, các bạn cùng lớp đã giúp đỡ, chia sẻ và góp ý cho nhóm những lời khuyên hữu ích về môn học này.

Mặc dù em đã cố gắng hết sức trong quá trình thực hiện báo cáo nhưng chắc chắc sẽ không tránh khỏi những thiếu sót. Kính mong các thầy và các bạn tận tình góp ý, chỉ bảo. Một lần nữa, em xin cảm ơn và mong nhận được tình cảm chân thành từ tất cả mọi người.

Tp.HCM ngày 05/07/2017

Trần Việt Anh

# MỤC LỤC

[NHẬN XÉT CỦA NGƯỜI HƯỚNG DẪN ii](#_Toc488072867)

[LỜI CẢM ƠN iv](#_Toc488072868)

[MỤC LỤC 1](#_Toc488072869)

[MỤC LỤC HÌNH 3](#_Toc488072870)

[MỤC LỤC BẢNG 4](#_Toc488072871)

[THUẬT NGỮ VIẾT TẮT 5](#_Toc488072872)

[CHƯƠNG 1 MỤC TIÊU CỦA ĐỒ ÁN. 6](#_Toc488072873)

[CHƯƠNG 2 THIẾT KẾ CHƯƠNG TRÌNH. 7](#_Toc488072874)

[CHƯƠNG 3 THỰC NGHIỆM 9](#_Toc488072875)

[3.1 Feature Raw (pixel intensity): 10](#_Toc488072876)

[3.1.1 KNN 10](#_Toc488072877)

[3.1.2 SVM 11](#_Toc488072878)

[3.2 LOCAL BINARY PATTERN(LBP) 11](#_Toc488072879)

[3.2.1 KNN 12](#_Toc488072880)

[3.2.2 SVM 13](#_Toc488072881)

[3.3 Histogram of oriented gradients(HOG) 14](#_Toc488072882)

[3.3.1 KNN 15](#_Toc488072883)

[3.3.2 SVM 16](#_Toc488072884)

[3.4 Bag of Words 17](#_Toc488072885)

[3.5 Kết quả đánh giá 17](#_Toc488072886)

[CHƯƠNG 4 HƯỚNG DẪN SỬ DỤNG 19](#_Toc488072887)

[TỔNG KẾT 21](#_Toc488072888)

[TÀI LIỆU THAM KHẢO 22](#_Toc488072889)

# MỤC LỤC HÌNH

[Hình 1 Kết quả mong muốn chương trình thực hiện được. 6](#_Toc488072890)

[Hình 2 Tài nguyên máy chủ khi chạy chương trình bằng Matlab. 9](#_Toc488072891)

[Hình 3 Tài nguyên máy chủ khi chạy chương trình bằng Matlab. 10](#_Toc488072892)

[Hình 4 Ví dụ tính giá trị LBP cho pixel trung tâm 12](#_Toc488072893)

[Hình 5 Thực thi chương trình 19](#_Toc488072894)

[Hình 6 Kết quả độ chính xác giữa các thuật toán nhận dạng khi đã được tối ưu 20](#_Toc488072895)

# MỤC LỤC BẢNG

[Bảng 1 Bảng so sánh kết quả. 17](#_Toc488072896)

# THUẬT NGỮ VIẾT TẮT

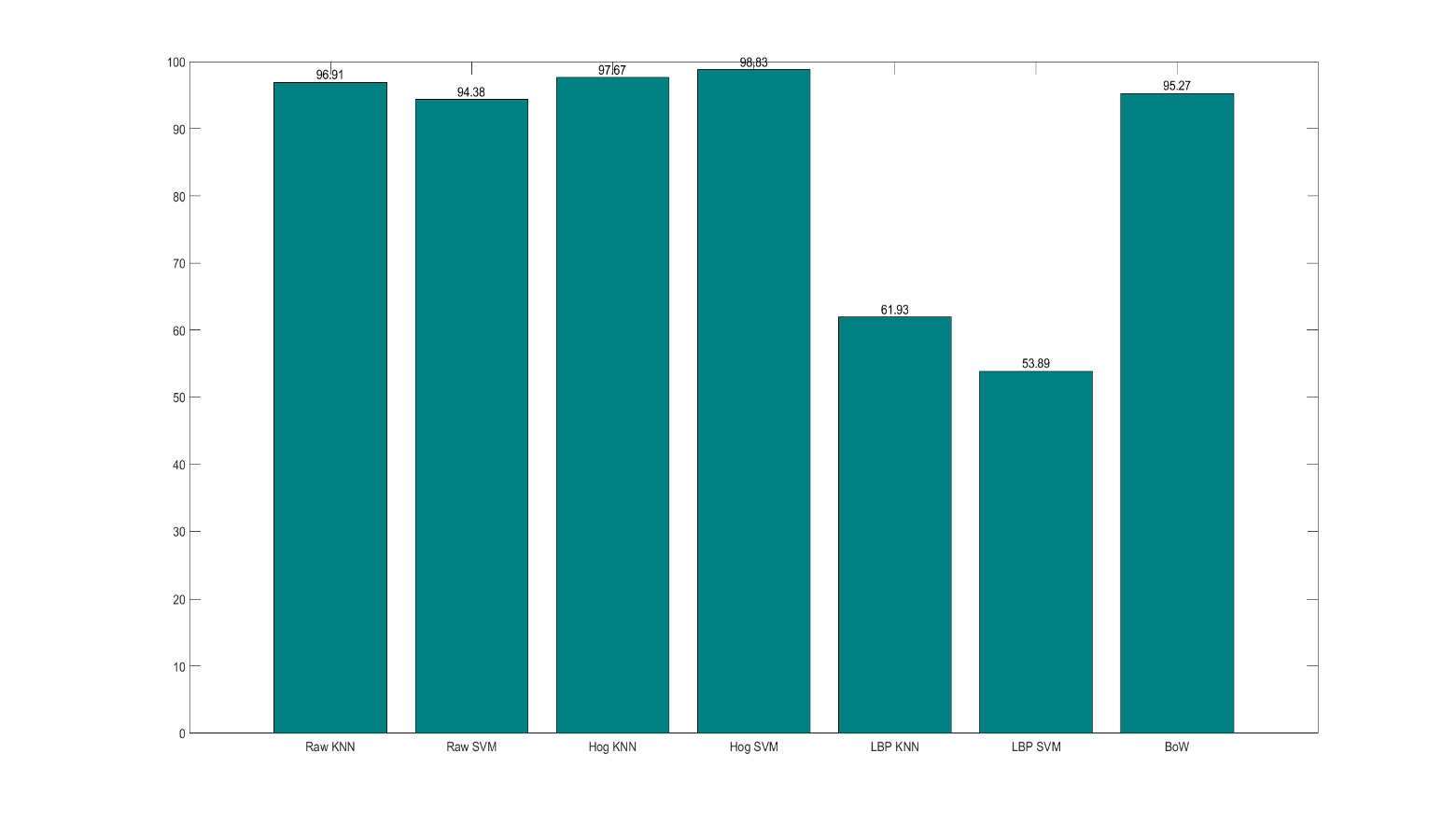
|  |
| --- |
| HOG – Histogram of Oriented Gradient |
| KNN – k-nearest neighbors |
| LDA – Linear Discriminant Analysis |
| LBP – Local Binary Pattern |
| SVM – Support Vector Machine |

# MỤC TIÊU CỦA ĐỒ ÁN.

Xây dựng chương trình so sánh mức độ chính xác của các thuật toán nhận dạng trên bộ dữ liệu chữ số viết tay với các tham số:

* Feature: raw (pixel intensity), HOG, LBP, BoW.
* Machine learning methods: KNN, SVM

Mục tiêu khác của đề tài là hiệu chỉnh các thông số đầu vào của các hàm các feature mà Matlab đã cung cấp để tăng hiệu xuất trong nhận dạng chữ số viết tay.



Hình 1 Kết quả mong muốn chương trình thực hiện được.

# THIẾT KẾ CHƯƠNG TRÌNH.

Chương trình bao gồm các hàm:

1. a\_detaicuoikhoa: Đóng vai trò chính để hiện thị kết quả của chương trình.
2. test: đóng vai trò gọi các hàm Recognition\_BoW\_svm, Recognition\_HOG\_KNN, Recognition\_HOG\_svm, Recognition\_LBP\_KNN, Recognition\_LBP\_svm, Recognition\_raw\_knn, Recognition\_raw\_svm nếu có yêu cầu training và predict lại chương trình.
3. load\_data: load image và label cho các hàm nhận dạng bên trên.
4. loadMNISTImages: load image(hàm được thầy cung cấp).
5. loadMNISTLabels: load label(hàm được thầy cung cấp).
6. Recognition\_raw\_knn: hàm đóng vai trò xuất và lưu các kết quả bao gồm model xây dựng bằng thuật toán học máy KNN, không trích xuất đặc trưng của ảnh và kết quả predict.
7. Recognition\_raw\_svm: : hàm đóng vai trò xuất và lưu các kết quả bao gồm model xây dựng bằng thuật toán học máy SVM, không trích xuất đặc trưng của ảnh và kết quả predict.
8. Recognition\_LBP\_KNN: hàm đóng vai trò xuất và lưu các kết quả bao gồm model xây dựng bằng thuật toán học máy KNN, thuật toán trích xuất đặc trưng của ảnh là LBP và kết quả predict.
9. Recognition\_LBP\_svm: hàm đóng vai trò xuất và lưu các kết quả bao gồm model xây dựng bằng thuật toán học máy SVM, thuật toán trích xuất đặc trưng của ảnh là LBP và kết quả predict.
10. extract\_Features\_LBP: hàm đóng vai trò trích lọc đặc trưng LBP của ảnh.
11. Recognition\_HOG\_KNN: hàm đóng vai trò xuất và lưu các kết quả bao gồm model xây dựng bằng thuật toán học máy KNN, thuật toán trích xuất đặc trưng của ảnh là HOG và kết quả predict.
12. Recognition\_HOG\_svm: hàm đóng vai trò xuất và lưu các kết quả bao gồm model xây dựng bằng thuật toán học máy SVM, thuật toán trích xuất đặc trưng của ảnh là HOG và kết quả predict.
13. extract\_HOG\_Features: hàm đóng vai trò trích lọc đặc trưng HOG của ảnh.
14. Recognition\_BoW\_svm: hàm đóng vai trò xuất và lưu các kết quả bao gồm model được xây dựng bằng thuật toán Bag of Words và kết quả predict.

Chương trình bao gồm các thư mục:

1. Data: chứa dữ liệu đầu vào của chương trình.
2. MDL: chứa model được lưu xuống sau khi training.
3. Results: lưu trữ kết quả predict.

# THỰC NGHIỆM

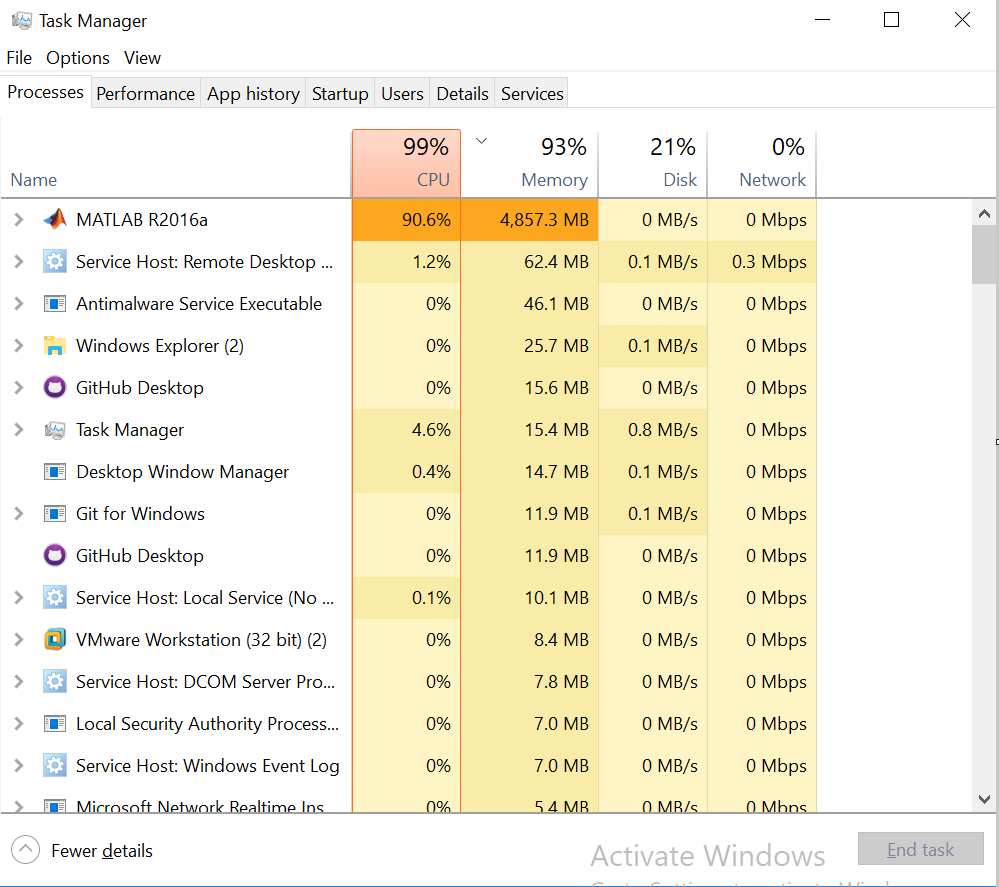
Chương trình được cài đặt trên Matlab version 2016a trên server 202.78.227.241 có cấu hình:

Hệ điều hành: Windows 10 x64.

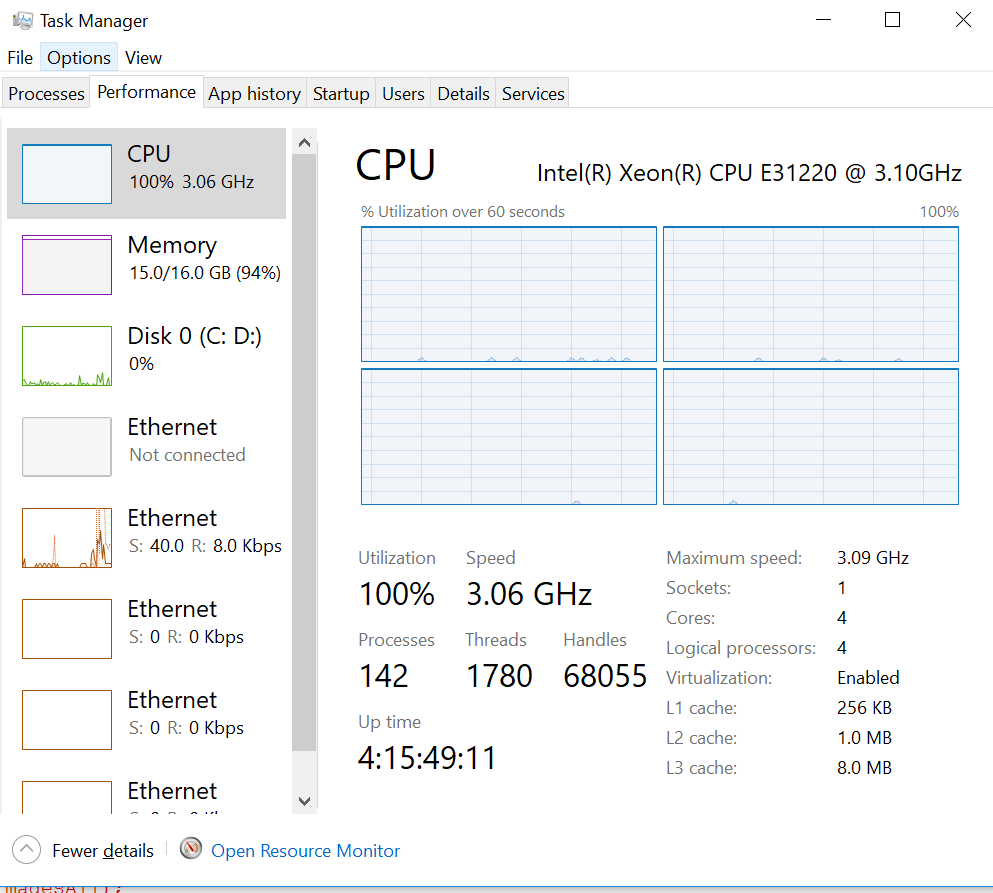
CPU: Intel Xeon E31220.

Ram: 16GB.

HDD: 500Gb.



Hình 2 Tài nguyên máy chủ khi chạy chương trình bằng Matlab.



Hình 3 Tài nguyên máy chủ khi chạy chương trình bằng Matlab.

## Feature Raw (pixel intensity):

Xây dựng model dựa trên ảnh chữ số viết tay và không rút trích đặc trưng của ảnh.

Tiến hành chạy riêng các hàm Recognition\_raw\_knn và Recognition\_raw\_svm để lấy kết quả

### KNN

Xây dựng model chỉ với tập ảnh và label, không có các tham số khác: tỉ lệ chính xác trong tiên đoán là 96.91%

Mdl\_raw\_knn = fitcknn(imgTrainImagesAll',lblTrainLabelsAll);

Xây dựng model với các tham số độ đo khoảng cách ‘minkowski’, thuật toán tìm kiếm lân cận gần nhất ‘kdtree’ và số lượng láng giềng tìm kiếm NumNeighbors=3 cho kết quả dự đoán chính xác tăng lên 97.06%

Mdl\_raw\_knn = fitcknn(imgTrainImagesAll',lblTrainLabelsAll,'Distance','minkowski','NSMethod','kdtree','NumNeighbors',3);

### SVM

Xây dựng model chỉ với tập ảnh và label, không có các thông số khác: tỉ lệ chính xác trong tiên đoán là 94.38%

Mdl\_raw\_svm = fitcecoc(imgTrainImagesAll',lblTrainLabelsAll);

Xây dựng model với thông số đầu vào như bên dưới cho tỉ lệ chính xác khi dự đoán là 94,44%.

Mdl\_raw\_svm = fitcecoc(imgTrainImagesAll',lblTrainLabelsAll,'Prior','uniform');

## LOCAL BINARY PATTERN(LBP)

Thông tin LBP của pixel tại trung tâm của mỗi khối ảnh sẽ được tính dựa trên thông tin của các pixel lận cận. Có thể tóm tắt các bước tiến hành như sau:

1. Xác định bán kính làm việc.
2. Tính giá trị LBP cho pixel ở trung tâm (xc, yc) khối ảnh dựa trên thông tin của các pixel lân cận:

Trong đó, (gp) là giá trị grayscale của các pixel lân cận, (gc) là giá trị grayscale của các trung tâm và (s) là hàm nhị phân được xác định như sau: s(z) = 1 nếu giá trị z ≥0.

Ví dụ:



Hình 4 Ví dụ tính giá trị LBP cho pixel trung tâm

1\*20 + 1\*21 + 1\*22 + 1\*23 + 0\*24 + 0\*25 + 0\*26 + 0\*27 = 15.

Xây dựng model dựa trên ảnh chữ số viết tay và rút trích đặc trưng của ảnh dựa vào thuật toán LBP.

Tiến hành chạy riêng các hàm Recognition\_LBP\_KNN và Recognition\_LBP\_svm để lấy kết quả.

### KNN

Với thuật toán rút trích đăc trưng LBP không có thông số đầu vào được cài đặt trong hàm “extract\_Features\_LBP” như bên dưới:

for i = 1:nData

img1D=img\_Data(:,i);

img2D=reshape(img1D,28,28); features\_Data(:,i)=extractLBPFeatures(img2D);

end

Xây dựng model chỉ với đặc trưng của ảnh lấy được khi sử dụng thuật toán rút trích đặc trưng LBP và label, không có các tham số khác: tỉ lệ chính xác trong tiên đoán là 61.93%.

Mdl\_LBP\_KNN=fitcknn(features\_data\_train',lblTrainLabelsAll)

Xây dựng model với các tham số độ đo khoảng cách ‘minkowski’, thuật toán tìm kiếm lân cận gần nhất ‘kdtree’ và số lượng láng giềng tìm kiếm NumNeighbors=3 cho kết quả dự đoán chính xác giảm xuống còn 57,12%

Ta thêm một số thông số đầu vào để cải thiện thuật toán rút trích đặc trung LBP với các thông số 'NumNeighbors',4,'Radius',2,'CellSize',[14 14]:

for i = 1:nData

img1D=img\_Data(:,i);

img2D=reshape(img1D,28,28); features\_Data(:,i)=extractLBPFeatures(img2D,'NumNeighbors',4,'Radius',2,'CellSize',[14 14]);

end

Kết quả đã tăng được tỉ lệ dự đoán chính xác lên 90,2%

### SVM

Với thuật toán rút trích đăc trưng LBP không có thông số đầu vào được cài đặt trong hàm “extract\_Features\_LBP” như bên dưới:

for i = 1:nData

img1D=img\_Data(:,i);

img2D=reshape(img1D,28,28); features\_Data(:,i)=extractLBPFeatures(img2D);

end

Xây dựng model chỉ với đặc trưng của ảnh lấy được khi sử dụng thuật toán rút trích đặc trưng LBP và label: tỉ lệ chính xác trong tiên đoán là 53,89%.

Mdl\_LBP\_SVM=fitcecoc(features\_data\_train',lblTrainLabelsAll)

Xây dựng model với thông số đầu vào như bên dưới cho tỉ lệ chính xác khi dự đoán tăng lên không đáng kể 53,96%

Mdl\_LBP\_SVM=fitcecoc(features\_data\_train',lblTrainLabelsAll,'Prior','uniform');

Ta thêm một số thông số đầu vào để cải thiện thuật toán rút trích đặc trung LBP với các thông số 'NumNeighbors',4,'Radius',2,'CellSize',[14 14]:

for i = 1:nData

img1D=img\_Data(:,i);

img2D=reshape(img1D,28,28); features\_Data(:,i)=extractLBPFeatures(img2D,'NumNeighbors',4,'Radius',2,'CellSize',[14 14]);

end

Kết quả đã tăng được tỉ lệ dự đoán chính xác lên 90,94%

## Histogram of oriented gradients(HOG)

Phương pháp HOG dựa trên việc đếm số lần xuất hiện của các hướng đạo hàm (gradient orientation) trong các vùng cục bộ của ảnh.

Bản chất của phương pháp HOG là các thông tin về hình dáng và vẻ bề ngoài của các đối tượng cục bộ trong ảnh có thể được mô tả bằng cách sử dụng thông tin về sự phân bố của các cường độ gradient (intensity gradients) hoặc của các hướng biên (edge directions). Các toán tử HOG được cài đặt bằng cách chia nhỏ một bức ảnh thành các vùng con được gọi là các “tế bào” (cells) và với mỗi cell đó một histogram về các hướng của gradients sẽ được tính cho các điểm nằm trong cell. Ghép các histogram này lại ta sẽ có một biểu diễn cho bức ảnh ban đầu. Để tăng cường hiệu năng nhận dạng, các histogram cục bộ có thể được chuẩn hóa về độ tương phản bằng cách tính một ngưỡng cường độ trong một vùng lớn hơn của ảnh, gọi là các khối (blocks) và sử dụng giá trị ngưỡng đó để chuẩn hóa tất cả các cell trong khối.

### KNN

Với thuật toán rút trích đăc trưng HOG không có thông số đầu vào được cài đặt trong hàm “extract\_HOG\_Features” như bên dưới:

for i = 1:nData

img1D=img\_Data(:,i);

img2D=reshape(img1D,28,28);

features\_Data(:,i)=extractHOGFeatures(img2D

end

Xây dựng model chỉ với đặc trưng của ảnh lấy được khi sử dụng thuật toán rút trích đặc trưng HOG và label: tỉ lệ chính xác trong tiên đoán là 97,09%

Mdl\_HOG\_KNN=fitcknn(features\_data\_train',lblTrainLabelsAll);

Ta tiến hành hiệu chỉnh hàm “extract\_HOG\_Features” bằng cách thêm các tham số vào hàm có sẵn của Matlab là “extractHOGFeatures” để cải thiện hiệu suất nhận dạng chữ số viết tay 'CellSize',[2 2],'BlockSize',[4 4],'NumBins',16:

for i = 1:nData

img1D=img\_Data(:,i);

img2D=reshape(img1D,28,28);

features\_Data(:,i)=extractHOGFeatures(img2D,'CellSize',[2 2],'BlockSize',[4 4],'NumBins',16);

features\_Data(:,i)=extractHOGFeatures(img2D);

end

Hiệu chỉnh hàm training tạo model bằng cách thêm các tham số 'Distance', 'cityblock', 'NumNeighbors',3:

Mdl\_HOG\_KNN=fitcknn(features\_data\_train',lblTrainLabelsAll,'Distance','cityblock','NumNeighbors',3);

### SVM

Với thuật toán rút trích đăc trưng HOG không có thông số đầu vào được cài đặt trong hàm “extract\_HOG\_Features” như bên dưới:

for i = 1:nData

img1D=img\_Data(:,i);

img2D=reshape(img1D,28,28);

features\_Data(:,i)=extractHOGFeatures(img2D

end

Xây dựng model chỉ với đặc trưng của ảnh lấy được khi sử dụng thuật toán rút trích đặc trưng HOG và label: tỉ lệ chính xác trong tiên đoán là 97,94%

Mdl\_HOG\_svm=fitcecoc(features\_data\_train',lblTrainLabelsAll);

Sử dụng lai hàm “extract\_HOG\_Features” đã được hiệu chỉnh bên trên kết hợp với việc hiệu chỉnh hàm xây dựng model với thông số đầu vào như bên dưới:

Mdl\_HOG\_svm=fitcecoc(features\_data\_train',lblTrainLabelsAll,'Prior','uniform');

Ta cải thiện được hiệu suất nhận dạng lên

## Bag of Words

Xây dựng túi từ chỉ với thông số đầu vào là bộ dữ liệu training image cho tỉ lệ chính xác khi tiên đoán là 95,27%

bag = bagOfFeatures(imgSets);

Xây dựng túi từ với các thông số đầu vào như bên dưới tỉ lệ chính xác là 27,15%

bag=bagOfFeatures(imgSets, 'VocabularySize',1000,'Verbose',false,'PointSelection','Detector');

Xây dựng túi từ với các thông số như bên dưới cho tỉ lệ chính xác 96.68%.

bag=bagOfFeatures(imgSets, 'VocabularySize',1000,'Verbose',false);

## Kết quả đánh giá

Khi đã tối ưu tất cả các thông số của các hàm ta được bảng kết quả sau

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | RAW(%) | LBP(%) | HOG(%) | BOW(%) |
| KNN | 97.06 | 90.2 | 97.32 |  |
| SVM | 94.44 | 90.94 | 98.82 | 96.6806 |

Bảng 1 Bảng so sánh kết quả.

Đánh giá các thuật toán:

LBP: Thời gian chạy training nhanh nhất trong các thuật toán tuy nhiên khi chưa được điều chỉnh các thông số trong hàm rút trích đặc trưng cũng như hàm xây dựng model cho tỉ lệ nhận dạng đúng chữ số viết tay chỉ hơn 50%.

HOG: Thời gian training rất lâu nhưng cho cho tỉ lệ nhận dạng đúng chữ số viết tay cao nhất. Khi chạy rất tốn tài nguyên của máy chủ

BOW: Thời gian training lâu tốn tài nguyên máy chủ, nhưng cho t ỉ lệ nhận dạng đúng chữ số viết tay cao.

RAW: Không rút trích đặc trưng của ảnh, thời gian training và predict nhanh.

# HƯỚNG PHÁT TRIỂN ĐỀ TÀI

Xây dựng được chương trình có giao diện web cho phép người dùng vẽ số và lựa chon loại feature và thuật toán machine learning để nhận dạng chữ số.

Cài đặt được deep learning với alexnet để đánh giá được mức độ chính xác khi nhận dạng chữ số viết tay.

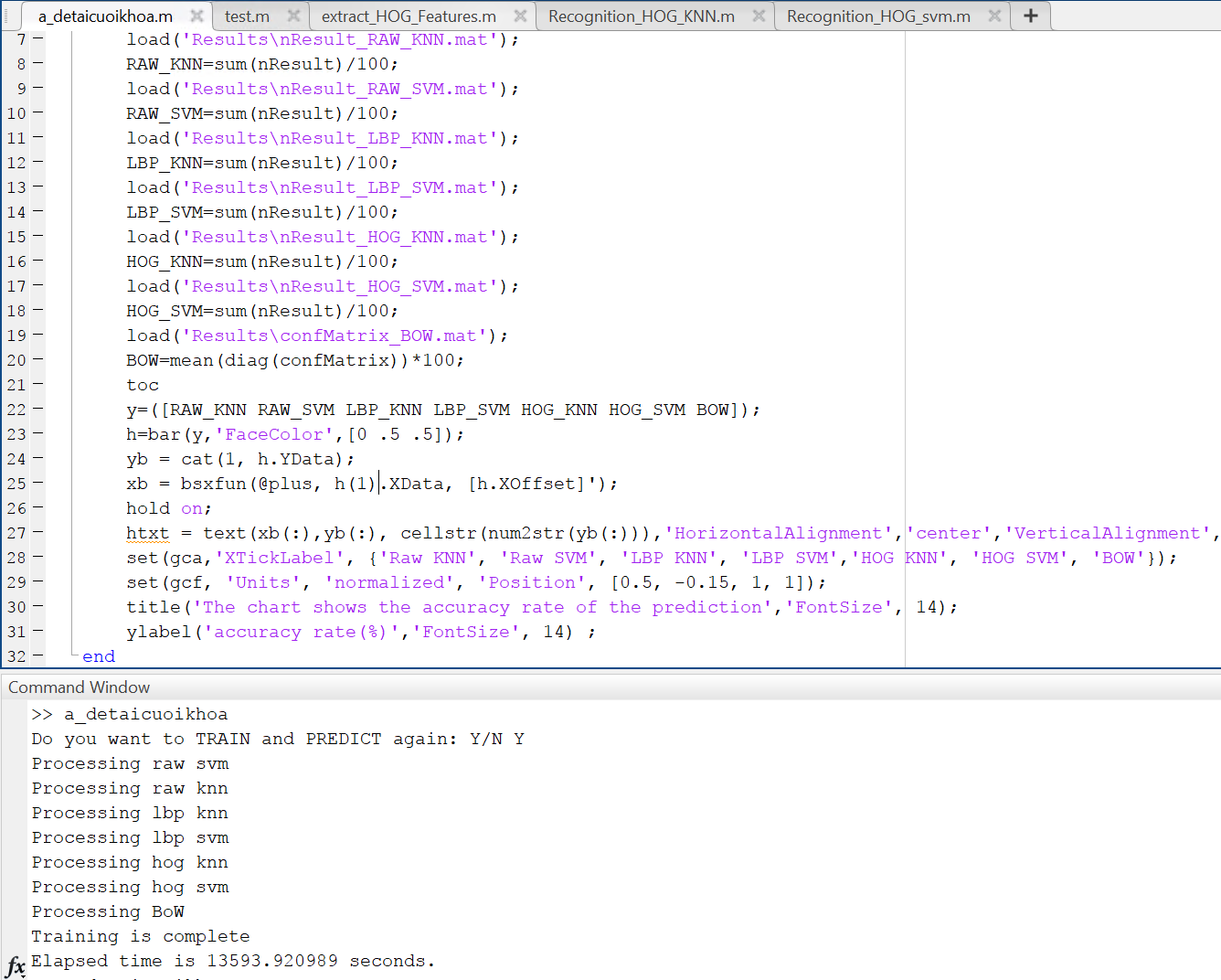
# HƯỚNG DẪN SỬ DỤNG

Source code của chương trình được đặt tại Github với đường link dưới đây: <https://github.com/vietanhtran88/VRA-TranVietAnh-CH1502001>

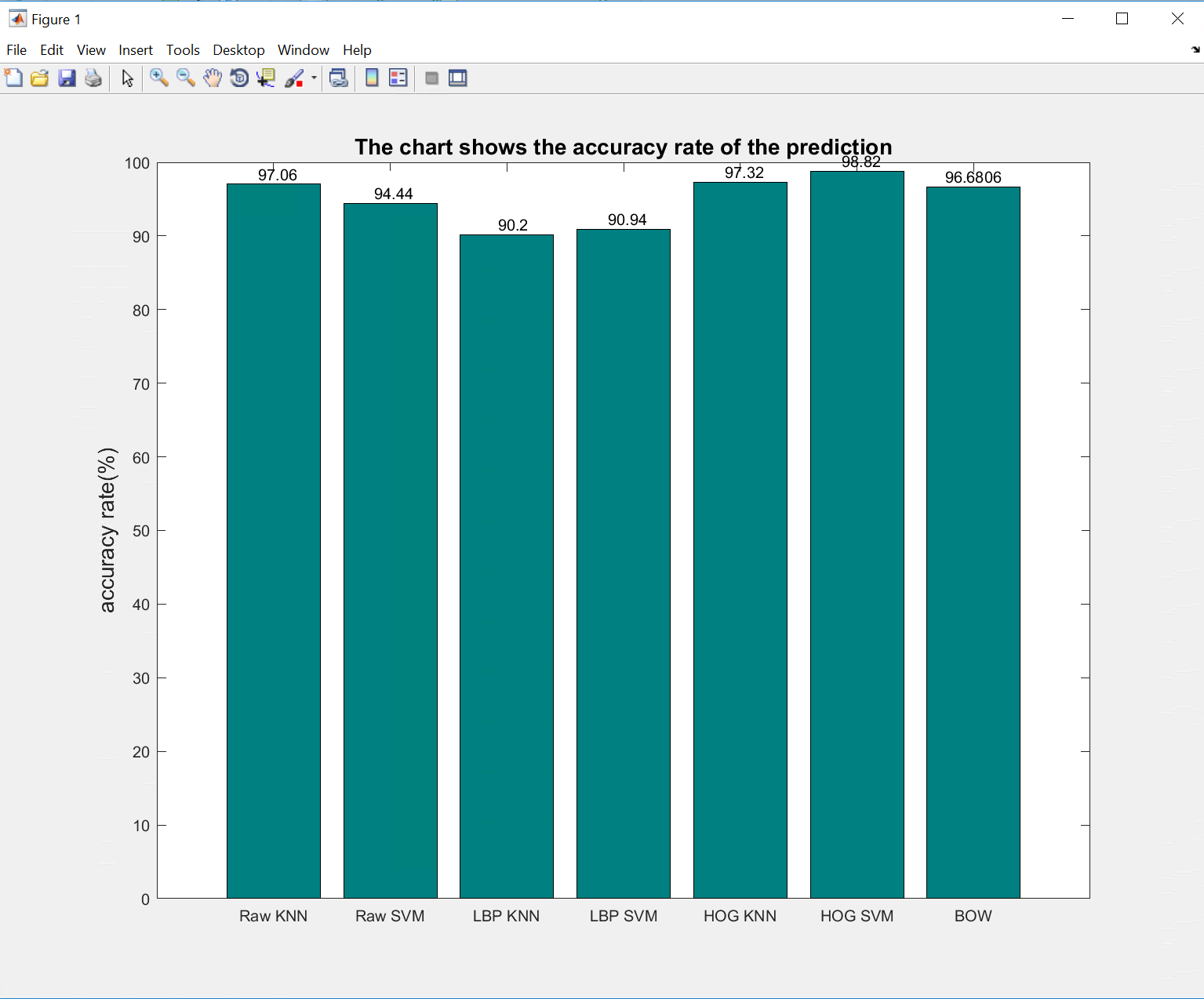
Video demo chương trình: <https://youtu.be/gnixEm80jDs>

Chạy hàm a\_detaicuoikhoa để thực thi chương trình:

* Nhập ‘Y’ nếu muốn training lại toàn bộ các model
* Nhấn phím bất kỳ để xuất kết quả so sánh mà không phải training lại các model.



Hình 5 Thực thi chương trình



Hình 6 Kết quả độ chính xác giữa các thuật toán nhận dạng khi đã được tối ưu

Thời gian thực thi chương trình nếu có bước training lại toàn bộ các model là 13593,9 giây tương đương với 3,7 giờ.

# TỔNG KẾT

Đề tài đã trình bày được cách thức cài đạt các thuật toán nhận dạng cơ bản dựa trên các hàm mà Matlab đã cung cấp. Ngoài ra đề tài cũng đã đưa ra được một số thông số nhằm tối ưu để tăng hiệu suất nhận dạng chữ số viết tay.

Đề tài có rất nhiều hạn chế cần phải khắc phục và đó cũng là hướng phát triển nếu có cơ hội:

* Hạn chế do bộ công cụ Matlab version 2016a chưa hỗ trợ deep learning nên chưa hiện thực cài đặt được cũng như đánh giá được hiệu suất nhận dạng chữ số viết tay.
* Hạn chế cho kiến thức về lập trình: chưa có kiến thức sâu về lập trình hiện tại chỉ viết được các chương trình cơ bản, chưa viết được các chương trình có giao diện để dễ dàng hỗ trợ người dùng

Nếu có cơ hội hướng phát triển mong muốn khi kết thúc môn học này là xây dựng một chương trình nhận dạng khuôn mặt người dùng hỗ trợ cho việc chấm công của nhân viên, do bản thân quản trị hệ thống của công ty nên có bộ dữ liệu từ của hệ camera, đây là đầu vào cần thiết cho quá trình training của chương trình.

Việc nghiên cứu đề tài đã giúp em thu được một số kết quả nhất định. Tuy nhiên do vấn đề thời gian cũng như sự hiểu biết có hạn nên đề tài này không thể tránh khỏi những thiếu sót nhất định. Em rất mong nhận được sự góp ý của các quý Thầy cũng như các bạn để hoàn thiện hơn trong việc tìm hiểu công nghệ này.

Trần Việt Anh

# TÀI LIỆU THAM KHẢO

[1] Bài giảng môn Nhận Dạng Thị Giác Và Ứng Dụng – TS.Lê Đình Duy, TS.Nguyễn Tấn Trần Minh Khang.