

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN MÔN HỌC
NHẬN DẠNG THỊ GIÁC VÀ ỨNG DỤNG

GVHD:

1. PGS.TS.Lê Đình Duy
2. TS.Nguyễn Tấn Trần Minh Khang

HVTH:

Khuất Bá Duy Lâm

TP Hồ Chí Minh, tháng 12 năm 2017

1. Tài nguyên tham khảo :

➤ Nguồn tài nguyên em đã sử dụng để xây dựng đồ án :

- ✓ Matlab : Sử dụng phiên bản Matlab 2017a
- ✓ Github : Lưu sourcecode, các bài thực hành trong quá trình học và các báo cáo
- ✓ Code tham khảo: Do giảng viên hướng dẫn và cung cấp, ngoài ra có 1 số chức năng tham khảo thêm.
- ✓ Bộ dữ liệu nhận dạng chữ viết tay gồm tập huấn luyện và tập kiểm tra:

<http://yann.lecun.com/exdb/mnist/index.html>

- ✓ Tham khảo MATLAB GUI :

<https://bitbucket.org/intelligenceagent/cudacnn-public/wiki/Home>

2. Kỹ thuật thực hiện :

- ✓ Em đã sử dụng chức năng **Matlab GUI** để xây dựng chương trình.
- ✓ Áp dụng các kiến thức đã học liên quan tới rút trích đặc trưng ảnh (HOG, LBP, Bag Of Word, Deep Learning), các thuật toán máy học(KNN,SVM) vào đồ án.

3. Giới thiệu đồ án :

➤ Thông tin đồ án :

- ✓ Đồ án làm theo **dạng 1** : Nhận dạng chữ số viết tay với tập dữ liệu MNIST
- ✓ ***Địa chỉ github lưu sourcecode và các báo cáo :***

<https://github.com/lamkbd/VRA>

Thư mục đồ án

✓ ***Địa chỉ YouTube link đến video minh họa cho việc cài đặt, chạy chương trình, và hiển thị kết quả :***

➤ Các chức năng chính của chương trình :

✓ Cho phép người dùng vẽ hình ảnh của 1 con số ở khung bên trái, sau đó chọn các kiểu rút trích đặc trưng và thuật toán học máy. Chương trình sẽ chạy và cho ra kết quả dự đoán con số.

✓ Đánh giá kết quả chính xác của phương pháp rút trích đặc trưng và thuật toán máy học, dựa trên tập dữ liệu huấn luyện và tập dữ liệu kiểm tra đã download về . Kết quả sẽ là số ảnh mà chương trình dự đoán đúng, tỉ lệ đúng trên tập kiểm tra.

➤ Cấu trúc thư mục, sourcecode và chức năng từng hàm :

✓ Thư mục DataTest : Chứa thư mục từ 0-9, với mỗi thư mục chứa ảnh mang nhãn của thư mục đó, được định dạng lại từ tập dữ liệu kiểm tra đã download về

✓ Thư mục DataTrain : Chứa thư mục từ 0-9, với mỗi thư mục chứa ảnh mang nhãn của thư mục đó, được định dạng lại từ tập dữ liệu huấn luyện đã download về

✓ Thư mục DataInput : Chứa file ảnh cho người dùng vẽ ra được hệ thống lưu lại.

➤ Các files và chức năng trong chương trình:

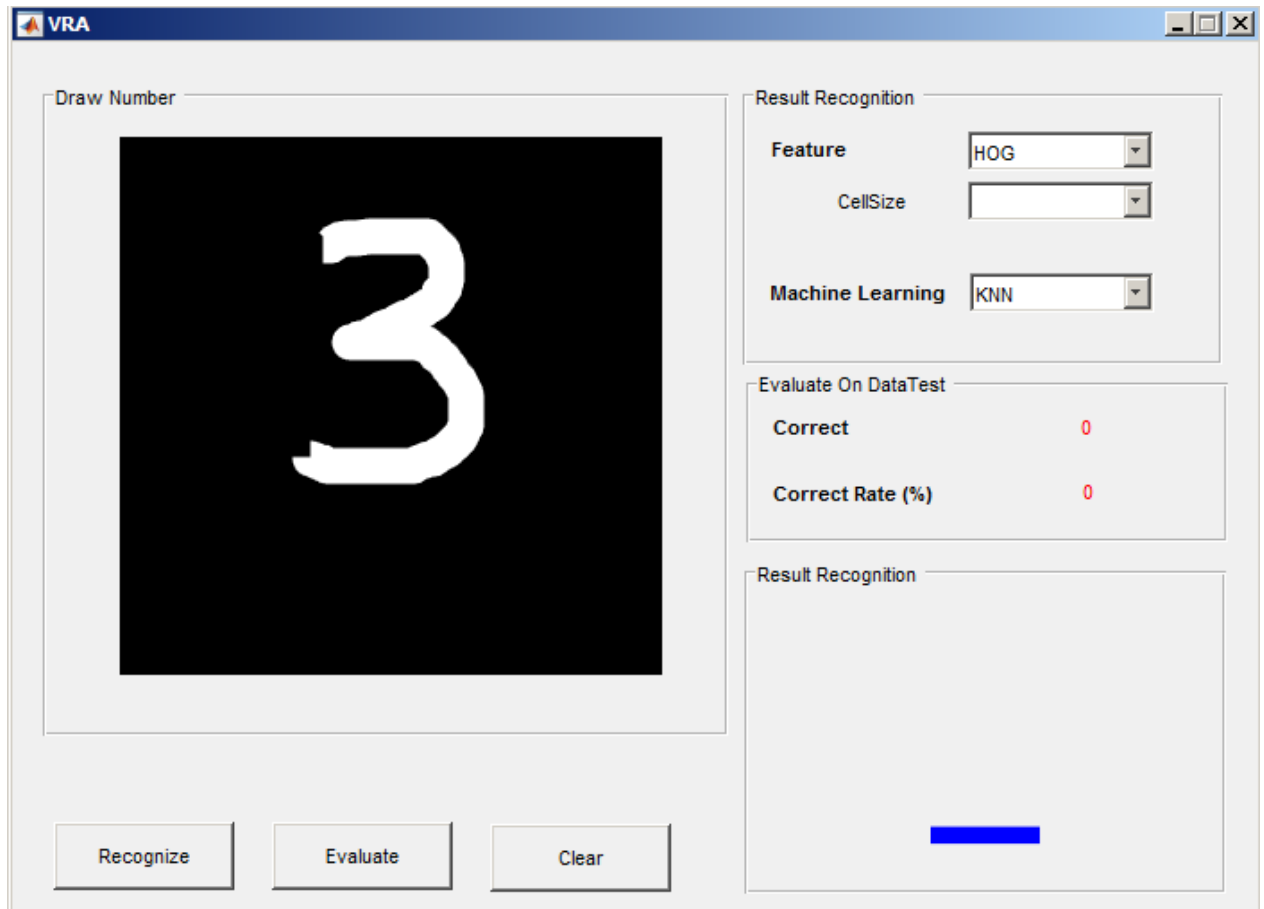
✓ VAR.fig : File giao diện chính của chương trình.

✓ VAR.m : File chứa sourcecode thực thi từ các chức năng trên giao diện người dùng (VAR.fig).

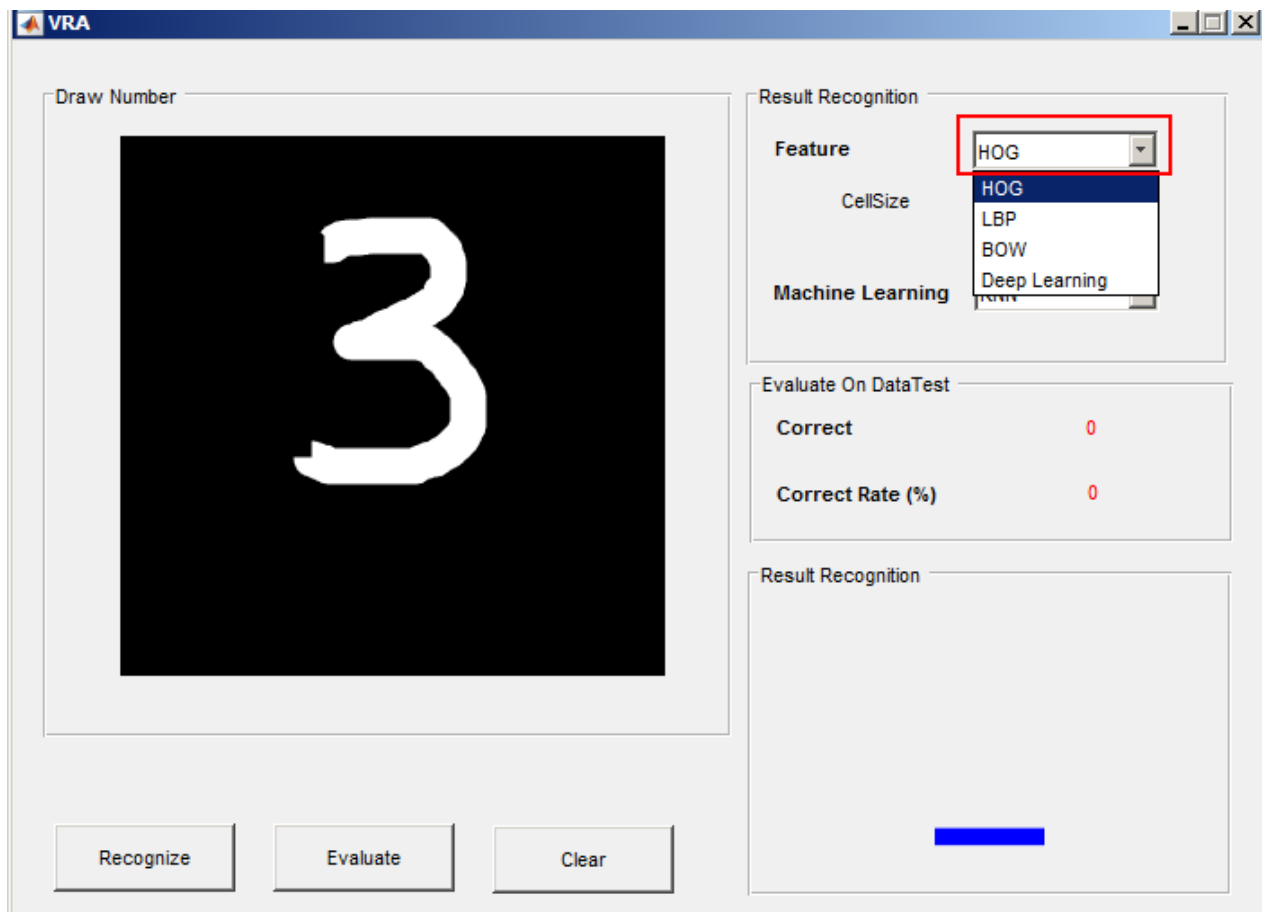
- ✓ createCategories.m : File định nghĩa hàm kiểm tra xem thư mục DataTrain, DataTest đã được tạo chưa, nếu chưa thì thực hiện tạo và phân chia Image vào các thư mục với nhãn tương ứng.
 - ✓ CreateDataTrain.m, CreateDataTest.m : File chứa hàm tạo thư mục và chia ảnh vào thư mục được gọi từ file createCategories.m
 - ✓ getBag.m : File chứa hàm rút trích đặc trưng theo phương pháp Bag Of Words, hàm trả về 1 bag (túi từ)
 - ✓ getFeatures_DL.m : File chứa hàm rút trích đặc trưng theo phương pháp Deep Learning
 - ✓ getFeatures_HOG_LBP.m : File chứa hàm rút trích đặc trưng theo phương pháp HOG hoặc LBP
 - ✓ getIMDS.m : File chứa hàm biến đổi thư mục ảnh thành imageDatastore (imds)
 - ✓ getMachineLearning_BOW.m : File chứa hàm học máy phân lớp ảnh khi người dùng chọn phương pháp rút trích đặc trưng là BOW
 - ✓ getMachineLearning_DL.m : File chứa hàm học máy phân lớp ảnh (SVM có tham số) khi người dùng chọn phương pháp rút trích đặc trưng là Deep Learning
 - ✓ getMachineLearning_KNN_SVM.m : File chứa hàm học máy phân lớp ảnh (SVM, KNN) khi người dùng chọn phương pháp rút trích đặc trưng là HOG hoặc LBP.
 - ✓ getCellSize.m : Trả về kích thước ô theo sự lựa chọn.
 - ✓ loadData.m, loadMNISTImages.m, loadMNISTLabels.m, readAndPreprocessImage.m :
- Các file phụ chứa hàm nạp ảnh vào chương trình.
- ✓ *.idx3-ubyte, *.idx1-ubyte : File dữ liệu gốc đã download

➤ Demo chương trình :

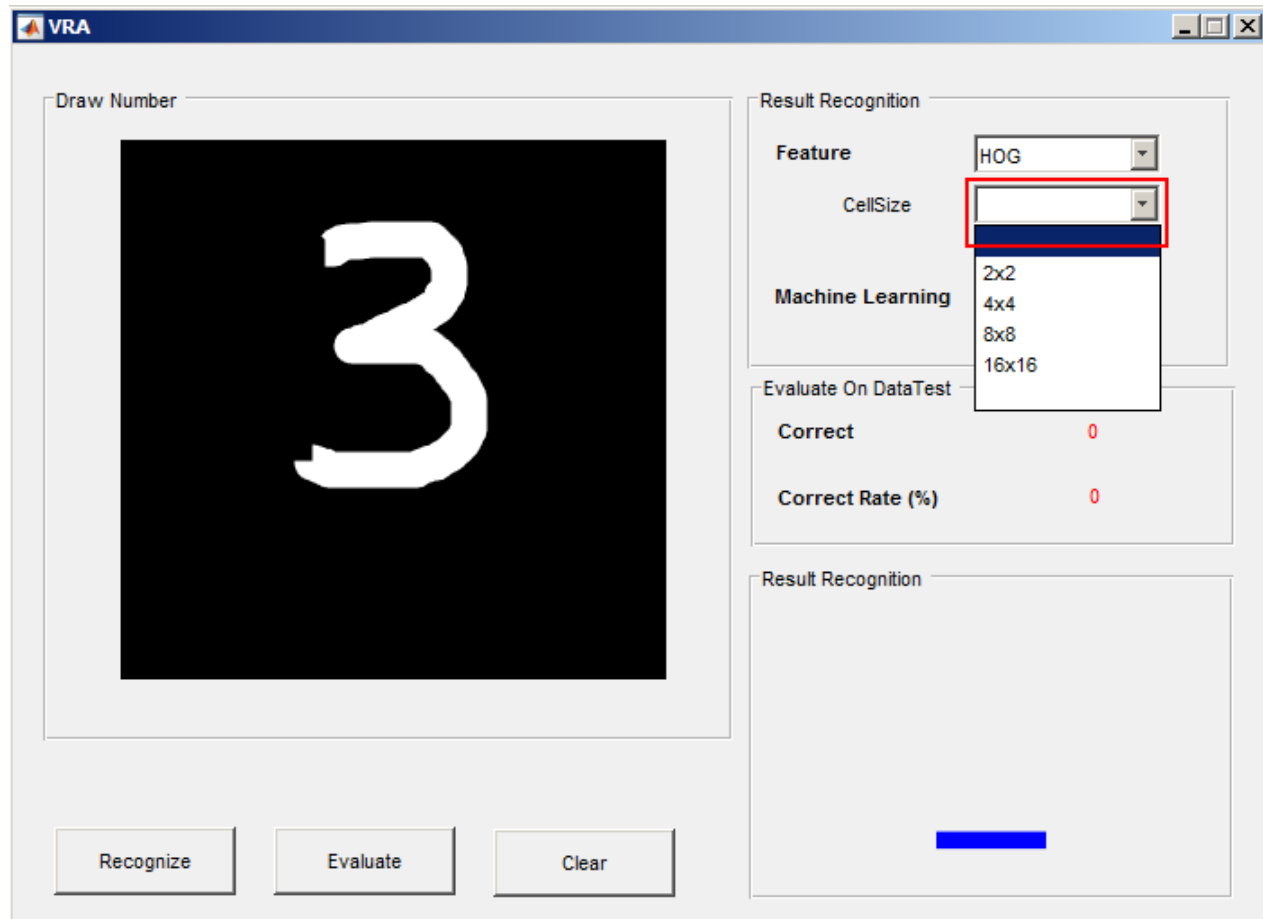
✓ Bước 1 : vẽ 1 chữ số trên màn hình :



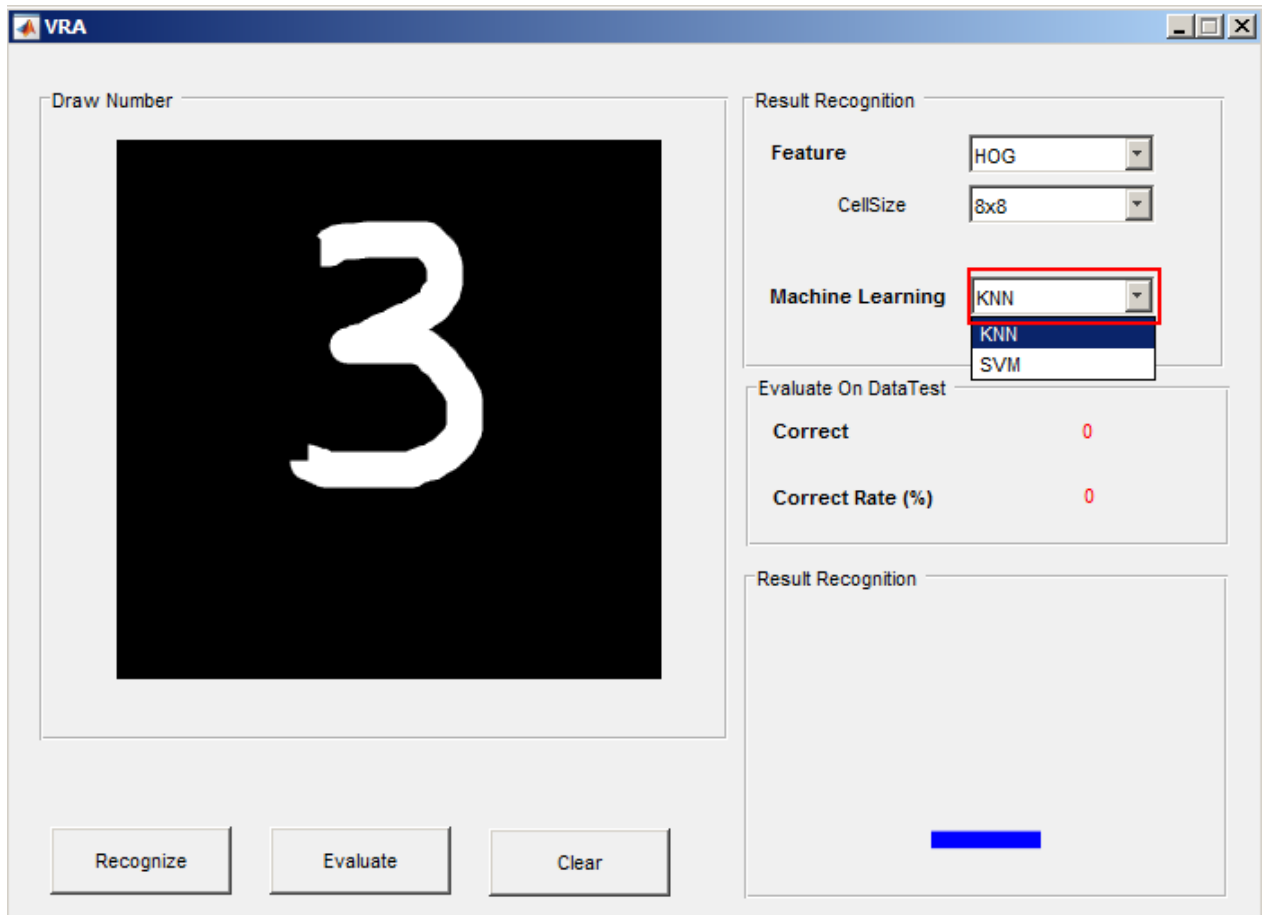
✓ Bước 2 : Chọn phương pháp rút trích đặc trưng



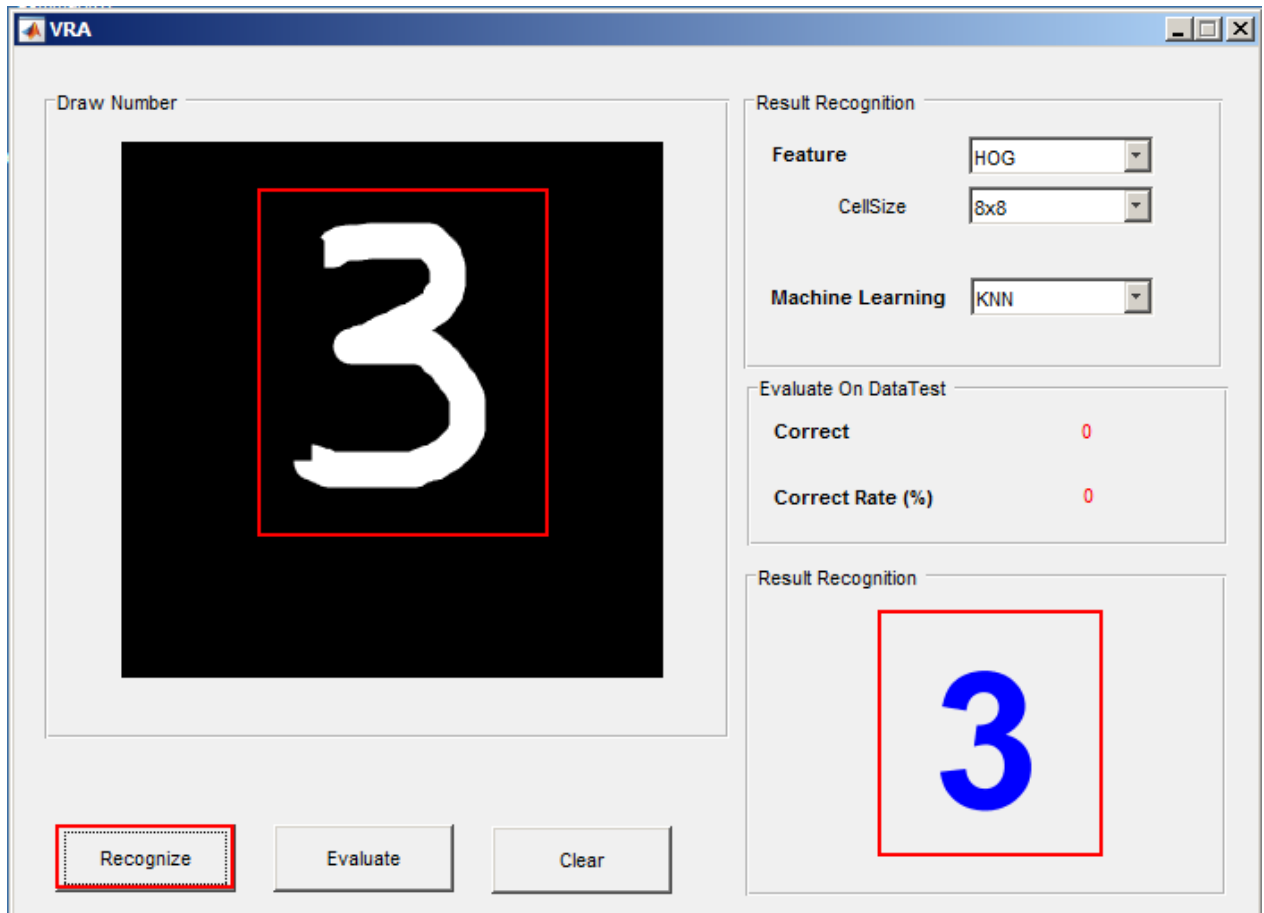
✓ Bước 3 : Chọn kích thước ô



✓ Bước 4 : Chọn phương pháp máy học



- ✓ Bước 5 : Bấm Recognize để chạy chương trình, cho ra kết quả dự đoán

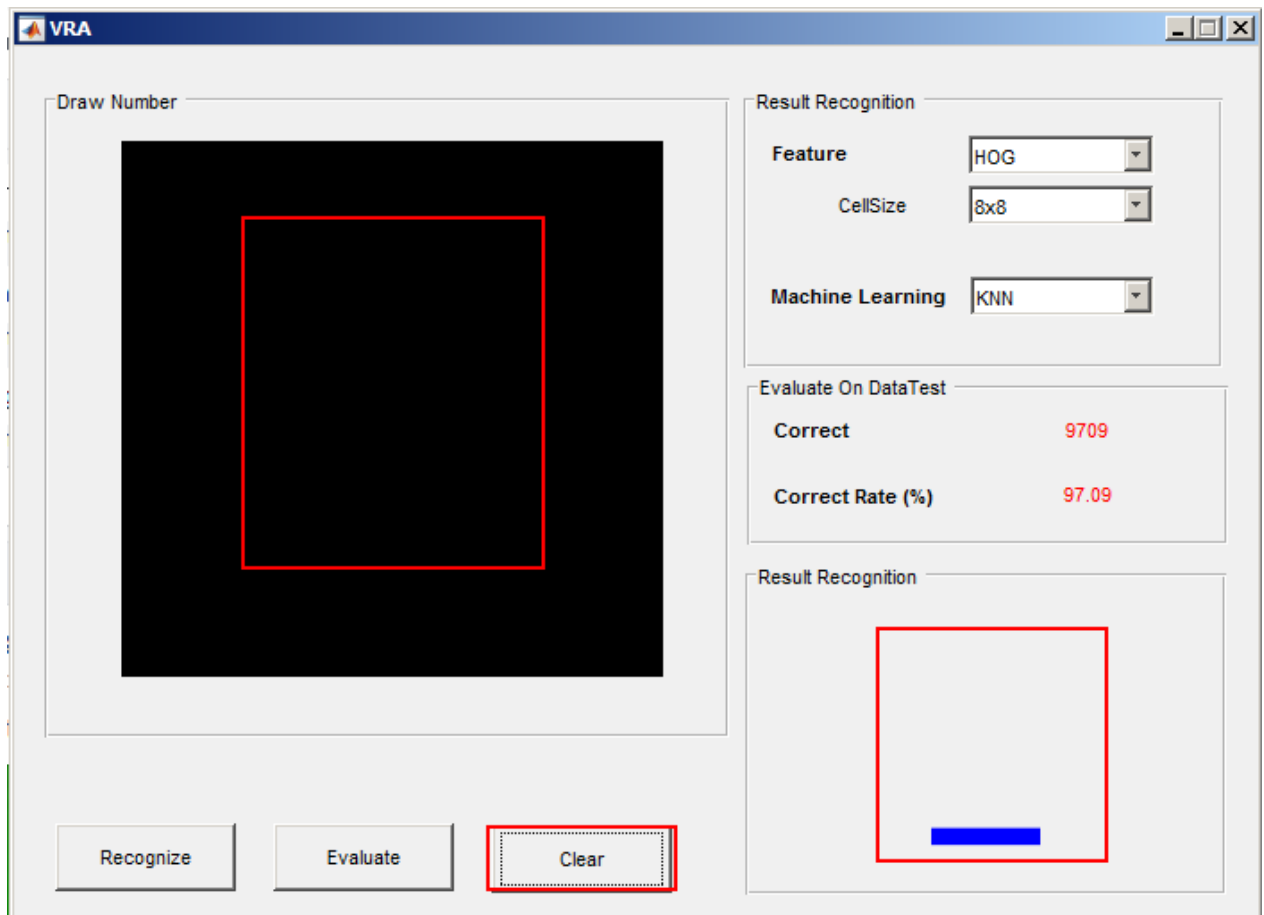


- ✓ Bấm Evaluate : Xem kết quả đánh giá nhận đúng của phương pháp đã chọn trên tập kiểm tra

The screenshot shows the VRA (Visual Recognition Assistant) software interface. On the left, a large black square contains a white digit '3' under the label 'Draw Number'. On the right, the 'Result Recognition' section shows settings: 'Feature' is set to 'HOG', 'CellSize' is '8x8', and 'Machine Learning' is 'KNN'. Below this, the 'Evaluate On DataTest' section displays a table with two rows: 'Correct' with a value of 9709, and 'Correct Rate (%)' with a value of 97.09. This table is highlighted with a red border. At the bottom of the interface are three buttons: 'Recognize', 'Evaluate' (which is highlighted with a red border), and 'Clear'. On the far right, a large blue digit '3' is displayed under another 'Result Recognition' label.

Evaluate On DataTest	
Correct	9709
Correct Rate (%)	97.09

- ✓ Bấm Clear : Để xóa ảnh đã vẽ và có thể vẽ lại ảnh khác



4. Các phương pháp rút trích đặc trưng và kỹ thuật máy học đã áp dụng trong chương trình

- Các phương pháp rút trích đặc trưng ảnh sử dụng :

- ✓ HOG :

Ý tưởng của phương pháp là tính gradient gồm 2 giá trị cường độ và hướng tại mỗi pixel, sau đó chia ảnh thành nhiều cell mỗi cell gồm mảng pixel kích thước $[A \times A]$, và tính histogram trên mỗi cell đó dựa trên gradient đã tính, một khối gồm mảng cell kích thước $[B \times B]$, được tính bằng cách ghép các histogram của các cell con. Thông thường mỗi histogram được chia thành 9 bin, 1 số các tham số chính đi kèm với phương pháp này trong matlab :

- Cellsize : Kích thước của 1 ô , mặc định [8x8] pixel
- Blocksize : Kích thước của 1 khối , mặc định [2x2] ô
- numBins : Kích thước của bin trong mỗi ô , mặc định 9 bins

✓ LBP :

Thông tin LBP của pixel tại trung tâm của mỗi khối ảnh sẽ được tính dựa trên thông tin của các pixel lân cận. Thuật toán định nghĩa 1 ma trận pixel có kích thước [A x A] lần lượt quét hết ảnh so sánh pixel lân cận với vị trí trung tâm để chuyển về dạng nhị phân (local binary pattern), sau đó tính giá trị LBP cho từng ma trận quét bằng cách cộng tổng trọng số (1-128) tại vị trí 1.

- NumNeighbors : Qui định số phần tử lân cận dùng để tính local binary pattern cho mỗi pixel, mặc định là 8
- Upright : Mặc định là true, nếu không cần rút trích khi xoay hình ảnh. Ngược lại là false để hỗ trợ tính năng này.
- CellSize : Qui định kích thước pixel trong mỗi ô

✓ BOW :

Các đặc trưng được rút trích theo 1 thuật toán nào đó (mặc định trong MATLAB khi không cấu hình là SURF), sau đó các đặc trưng được phân lớp từ điển như 1 bộ từ (túi từ) dựa vào thuật toán K-Means. Dựa vào túi từ đó định nhãn cho hình ảnh cần dự đoán. 1 số tham số chính của phương pháp :

- VocabularySize : Qui định kích thước túi từ, chính là k phần tử trong thuật toán K-Mean
- BlockWidth : Qui định kích thước ma trận quét theo phương pháp rút trích đặc trưng SURF
- Upright : Mặc định là false, nếu cần rút trích khi xoay hình ảnh. Ngược lại true sẽ tắt tính năng này.

✓ Deep Learning (Alexnet):

Mô hình được học từ dữ liệu của Imagnet với kiến trúc gồm 8 tầng (layer) trong đó có 5 layer đầu là convolutional layer và 3 layer còn lại là fully connected layer. Đầu ra lớp cuối cùng là 1000 chiều tương ứng với số lớp cần phân lớp.

➤ Các phương pháp học máy :

✓ SVM :

Về ý tưởng thì SVM sử dụng thủ thuật để ánh xạ tập dữ liệu ban đầu vào không gian nhiều chiều hơn. Sau khi ánh xạ vào không gian nhiều chiều SVM tìm ra 1 mặt cắt phù hợp để phân lớp tập dữ liệu đó.

Mặt cắt tương tự như phương trình đường thẳng có dạng : $y = ax + b$

✓ KNN :

Ý tưởng của thuật toán khá đơn giản là với 1 nhãn cần dự đoán, thuật toán sẽ tìm ra những nhãn gần nhất (k) với nhãn cần dự đoán, sau đó tính khoảng cách từ nhãn cần dự đoán đến từng phần tử thuộc k, khoảng cách tới phần tử k nào gần nhất thì nhãn sẽ được gán vào lớp k đó.