**CÁC VẤN ĐỀ CHỌN LỌC TRONG THỊ GIÁC MÁY TÍNH**

**BÁO CÁO**

**ĐỀ TÀI: NHẬN DẠNG KÝ TỰ TỪ ẢNH SỐ**

**GVHD: TH.S Cáp Phạm Đình Thăng**

**Sinh viên thực hiện:**

**Từ Nguyên Gia Khánh – 12520200**

**Nguyễn Anh Quân – 12520337**

**Nguyễn Kim Hoàn - 12520149**

**GIỚI THIỆU CHUNG**

Hiện nay, nhu cầu về việc rút trích từ ngữ từ hình ảnh đang ngày càng phát triển, bên cạnh sự gia tăng về nhu cầu là sự phát triển của công nghệ nhận dạng ký tự quang học (Optical Character Recognition) hay còn được gọi tắt là OCR. Đây là một công nghệ giúp chuyển đổi hình ảnh của chữ viết tay hoặc đánh máy thành các ký tự đã được mã hóa trong máy tính.

Giả sử chúng ta cần chỉnh sửa một số tài liệu giấy như: Các bài viết trên tạp chí, tờ rơi, hoặc một tập tin PDF hình ảnh. Rõ ràng, chúng ta không thể sử dụng một máy quét để chuyển các tài liệu này thành tập tin văn bản để có thể chỉnh sửa (ví dụ như trình soạn thảo Microsoft Word). Tất cả những gì máy quét có thể làm là tạo ra một hình ảnh hoặc một bản chụp của các tài liệu. Để giải nén và sử dụng lại dữ liệu từ tài liệu được quét, hình ảnh máy ảnh hoặc hình ảnh của các tập tin PDF, chúng ta cần một phần mềm OCR. Nó sẽ xuất ra kí tự trên hình ảnh, ghép chúng thành từ và sau đó ghép các từ thành câu. Nhờ vậy, chúng ta có thể truy cập và chỉnh sửa nội dung của tài liệu gốc.

Tương tự, những tài liệu cổ đang bị hư hại theo thời gian và việc viết tay hay đánh máy lại những tài liệu này sẽ tốn rất nhiều chi phí, thời gian và không đảm bảo được độ chính xác cũng như là sự an toàn cho tài liệu nền. Việc này rất cần một công nghệ lấy từ ngữ từ hình ảnh chụp.

**Phạm vi ứng dụng:**

OCR thường được ứng dụng như một phần mềm cài đặt trên máy tính hoặc tích hợp đi kèm với phần cứng (cụ thể là máy Scanner) hoặc được thiết lập như một ứng dụng trực tuyến.

Ví dụ:

* **ABBYY FineReader** là một phần mềm ứng dụng công nghệ OCR nổi tiếng nhất hiện nay.
* **OmniPage** là một ứng dụng đi kèm với dòng máy Kodak ScanMate i1120 giúp nhận dạng văn bản sau khi quét.
* Website www.ocrnow.com là một trang web cung cấp giải pháp OCR trực tuyến giúp người dùng có thể upload trực tiếp file cần OCR lên server và nhận lại kết quả dưới dạng text chỉ sau vài phút.

**Khó khăn, hạn chế và một số công nghệ có liên quan:**

Có nhiều phương pháp để tạo ra một phần mềm dạng OCR, độ chính xác của các phương pháp này phụ thuộc vào công nghệ tạo nên phần mềm. Các phương pháp này đạt được độ tin cậy trong các hình ảnh có chất lượng tốt và vừa.

Các chương trình hỗ trợ OCR có thể nhận dạng ký tự với tỷ lệ trên 90% đối với chất lượng hình ảnh rõ nét và font chữ thông thường. Đối với hình chất lượng kém, font chữ đặc biệt hoặc chữ viết tay thì kết quả cho ra không mấy khả quan.

Độ chính xác của việc rút trích văn bản là điều quan trọng nhất. Nhóm tác giả Kirill Safronov [1] cho rằng một số sai sót trong quá trình chuyển đổi thường không quá quan trọng trừ các trường hợp như rút trích số serial từ ảnh chụp,...

Để khắc phục tình trạng kết quả xuất ra không chính xác của công nghệ OCR, nhiều công nghệ khác đã ra đời, tác giả A. Vinutha M H [2] đã ứng dụng định hướng robot (Optical Character Recognition Based Auto Navigation of Robot). Việc định hướng của robot dựa vào bảng tính hiệu như là một cột mốc đánh dấu đường đi tiếp theo của robot. Định hướng tự động của các robot trong một vùng lớn đòi hỏi nhiều bảng tín hiệu khác nhau với mô hình nhận dạng duy nhất. Ngoài ra, hệ thống này còn cho phép nhận diện vị trí tên riêng.

Bên cạnh việc cải thiện độ chính xác, cần có sự thay đổi kích thước của thiết bị nhận dạng, tác giả Ali Ahmadi [3] đã đề cập trong nghiên cứu của mình, tốc độ xử lý và độ chính xác cao là yêu cầu lớn hiện nay của các thiết bị nhận dạng ký tự dạng nhỏ, ví dụ như bút biết nhận dạng.

Ngoài sự đa dạng trong cách thức nhận dạng, OCR còn đa dạng về cách dùng, nó được chia thành hai cách, dùng online và dùng offline, tác giả Priya Sharma [4] có nhận xét về hai cách dùng này như sau: (1) Nhận dạng offline: nhận dạng các văn bản in ra giấy hoặc các bản viết tay và nó đòi hỏi quá trình scan trên mặt giấy hoặc mặt vật liệu có chữ. Cách này thường đòi hỏi con người phải thực hiện một số thao tác như phân loại, lưu trữ và chỉnh sửa văn bản trước khi scan. (2) Nhận dạng online: thường chỉ được dùng cho nhận dạng chữ viết tay được lưu trữ ở dạng kỹ thuật số, thông thường để scan dạng này chúng ta thường dùng một loại bút đặc biệt nhưng do sự thành công của các nghiên cứu gần đây mà giờ đã có các thiết bị khác thay thế. Việc nhận dạng online nhằm giúp con người giao tiếp với máy tính tốt hơn bằng cách viết tay thay vì gõ phím.

**Tóm tắt đề tài:**

Tên đề tài: Nhận dạng ký tự từ ảnh số

Công nghệ sử dụng:

* Phương pháp và thuật toán nhận dạng ký tự quang học OCR (**Optical Character Recognition**)
* Phần mềm Matlab và công nghệ xử lý ảnh trong Matlab

1. **Giới thiệu chung về phần mềm Matlab**

**1.1 Khái niệm về Matlab**

Matlab là một ngôn ngữ lập trình thực hành bậc cao được sử dụng để giải các bài toán về kỹ thuật. Matlab tích hợp được việc tính toán, thể hiện kết quả, cho phép lập trình, giao diện làm việc rất dễ dàng cho người sử dụng. Dữ liệu cùng với thư viện được lập trình sẵn cho phép người sử dụng có thể có được những ứng dụng sau đây.

* Sử dụng các hàm có sẵn trong thư viện, các phép tính toán học thông thường.
* Cho phép lập trình tạo ra những ứng dụng mới.
* Cho phép mô phỏng các mô hình thực tế.
* Phân tích, khảo sát và hiển thị dữ liệu.
* Với phần mềm đồ hoạ cực mạnh.
* Cho phép phát triển, giao tiếp với một số phần mềm khác như C++, Fortran

**1.2 Tổng quan về cấu trúc dữ liệu của Matlab, các ứng dụng**

Matlab là một hệ thống tương giao, các phần tử dữ liệu là một mảng (mảng này không đòi hỏi về kích thước). Chúng cho phép giải quyết các vấn đề liên quan

đến lập trình bằng máy tính, đặc biệt sử dụng các phép tính về ma trận hay vectơ và có thể sử dụng ngôn ngữ C học Fortran lập trình rồi thực hiện ứng dụng lập trình đó bằng các câu lệnh gọi từ Matlab. Matlab  được viết tắt từ chữ “**MAT**rix **LAB**oratory” tức là thư viện về ma trận, từ đó phần mềm Matlab được viết nhằm

cung cấp cho việc truy cập vào phần mềm ma trận một cách dễ dàng, phần mềm ma trận này được phát triển bởi các công trình Linpack và Eispack. Ngày nay Matlab được phát triển bởi Lapack và Artpack tạo nên một nghệ thuật phần mềm cho ma trận.

**1.2.1 Dữ liệu**

Dữ liệu của Matlab thể hiện dưới dạng ma trận (hoặc mảng - tổng quát), và có các kiểu dữ liệu được liệt kê sau đây:

* Kiểu đơn single, kiểu này có lợi về bộ nhớ dữ liệu vì nó đòi hỏi ít byte nhớ hơn, kiểu dữ liệu này không được sử dụng trong các phép tính toán học, độ chính xác kém hơn.
* Kiểu double kiểu này là kiểu thông dụng nhất của các biến trong Matlab.
* Kiểu Sparse.
* Kiểu uint8, uint8, uint16, uint64...
* Kiểu char ví dụ “Hello”.
* Kiểu cell.
* Kiểu Structure.

Trong Matlab kiểu dữ liệu double là kiểu mặc định sử dụng trong các phép tính số học.

**1.2.2 Ứng dụng**

Matlab tạo điều kiện thuận lợi cho:

* Các khoá học về toán học.
* Các kỹ sư, các nhà nghiên cứu khoa học.
* Dùng Matlab để tính toán, nghiên cứu tạo ra các sản phẩm tốt nhất trong sản xuất.

**1.2.3  Toolbox là một công cụ quan trọng trong Matlab**

Công cụ này được Matlab cung cấp cho phép bạn ứng dụng các kỹ thuật để phân tích, thiết kế, mô phỏng các mô hình.

Ta có thể tìm thấy toolbox ở trong môi trường làm việc của.

* Mạng nơron.
* Logic mờ.
* Simulink.

**Giới thiệu về ảnh số và xử lý ảnh số trong matlab:**

Xử lý ảnh là một lĩnh vực mang tính khoa học và công nghệ. Nó là một ngành khoa học mới mẻ so với nhiều ngành khoa học khác nhưng tốc độ phát triển của nó rất nhanh, kích thích các trung tâm nghiên cứu, ứng dụng, đặc biệt là máy tính chuyên dụng riêng cho nó.

Xử lý ảnh được đưa vào giảng dạy ở bậc đại học ở nước ta khoảng chục năm nay. Nó là môn học liên quan đến nhiều lĩnh vực và cần nhiều kiến thức cơ sở khác. Đầu tiên phải kể đến Xử lý tín hiệu số là một môn học hết sức cơ bản cho xử lý tín hiệu chung, các khái niệm về tích chập, các biến đổi Fourier, biến đổi Laplace, các bộ lọc hữu hạn… Thứ hai, các công cụ toán như Đại số tuyến tính, Sác xuất, thống kê. Một số kiến thứ cần thiết như Trí tuệ nhân tao, Mạng nơ ron nhân tạo cũng được đề cập trong quá trình phân tích và nhận dạng ảnh.

Các phương pháp xử lý ảnh bắt đầu từ các ứng dụng chính: nâng cao chất lượng ảnh và phân tích ảnh. Ứng dụng đầu tiên được biết đến là nâng cao chất lượng ảnh báo được truyền qua cáp từ Luân đôn đến New York từ những năm 1920. Vấn đề nâng cao chất lượng ảnh có liên quan tới phân bố mức sáng và độ phân giải của ảnh. Việc nâng cao chất lượng ảnh được phát triển vào khoảng những năm 1955. Điều này có thể giải thích được vì sau thế chiến thứ hai, máy tính phát triển nhanh tạo điều kiện cho quá trình xử lý ảnh sô thuận lợi. Năm 1964, máy tính đã có khả năng xử lý và nâng cao chất lượng ảnh từ mặt trăng và vệ tinh Ranger 7 của Mỹ bao gồm: làm nổi đường biên, lưu ảnh. Từ năm 1964 đến nay, các phương tiện xử lý, nâng cao chất lượng, nhận dạng ảnh phát triển không ngừng. Các phương pháp tri thức nhân tạo như mạng nơ ron nhân tạo, các thuật toán xử lý hiện đại và cải tiến, các công cụ nén ảnh ngày càng được áp dụng rộng rãi và thu nhiều kết quả khả quan.

**B.Phương Pháp và Thuật Toán Nhận Dạng Kí Tự**

1**.Xử lý dữ liệu (Phân tích ảnh)**

-Quá trình phân tích ảnh thực chất là phân tích chuỗi văn bản ảnh thành từng ký tự một. Việc phân tích này dựa trên việc nhận dạng các pixel màu của ký tự ảnh với dữ liệu đầu vào đã được xử lý và chuyển thành định dạng \*.bmp chỉ có 2 pixel điểm đen RGB(0,0,0) và điểm trắng RGB(255, 255, 255).

Quá trình phân tích ảnh được thực hiện tuần tự như sau :

-Phân tích văn bản thành từng dòng văn bản .

-Phân tích từng dòng văn bản thành từng ký tự ảnh một.

**1.1.Tách dòng kí tự**

- Việc tách dòng kí tự chính là cơ sở để tiến hành tách từng kí tự một cách dễ dàng và nhanh chóng.

* Thuật toán :

-Bắt đầu từ điểm ảnh (x, y) đầu tiên của file ảnh (0,0) , gán số dòng đang xét lines =0;

a.-Xác định giới hạn trên của dòng : Quét hết chiều rộng của ảnh, trên cùng một giá trị y ( chạy x<=chiều rộng ảnh, giữ nguyên y )

+ Nếu gặp một pixel đen thì ghi nhận y chính là tọa độ giới hạn trên của dòng(top\_line).Dừng duyệt x.

+Nếu không , tiếp tục duyệt

+Nếu khi quét hết chiều rộng của ảnh, vẫn không tìm thấy giá trị điểm màu đen thì reset lại x=0, và tiếp tục duyệt đến dòng tiếp theo (tăng y=y+1)

b.-Xác định giới hạn dưới của dòng :

+ Bắt đầu duyệt từ giới hạn trên (đỉnh ) vừa tìm thấy của dòng (0,top\_line)

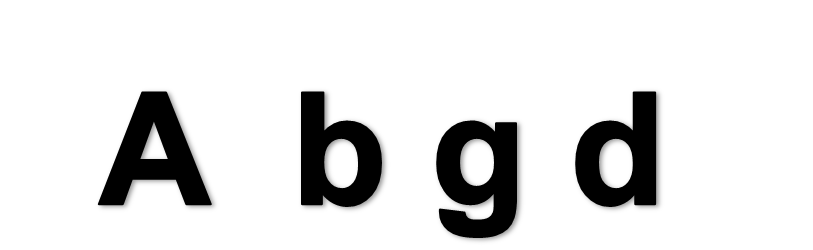
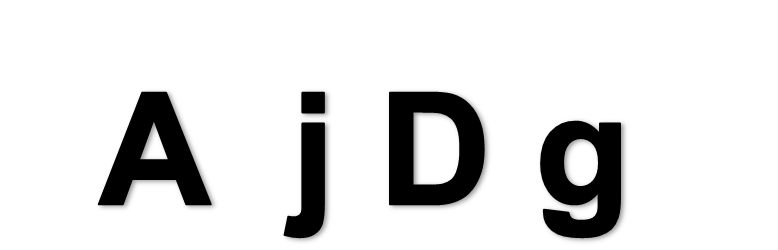
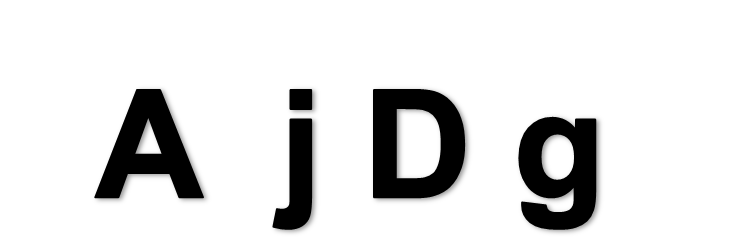
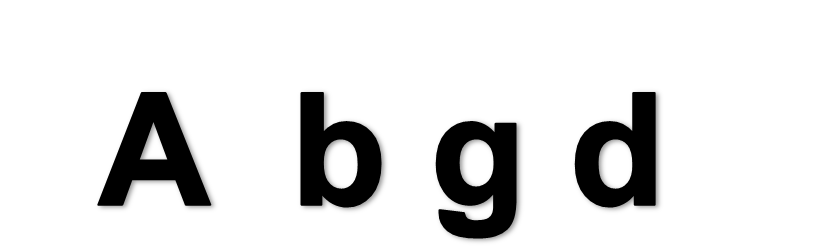
+ Tương tự như xác định giới hạn trên , ta duyệt hết chiều rộng của ảnh trên cùng một giá trị y.

+Nếu duyệt hết dòng mà không tìm thấy ký tự pixel đen nào thì ghi nhận y-1 là giới hạn dưới của dòng (bottom\_line).Dừng duyệt.Tăng số dòng lên (lines++).

+Nếu chưa tìm thấy bottom\_line, tiếp tục duyệt đến dòng tiếp theo (tăng y, reset x=0).

c.Bắt đầu từ giới hạn dưới y (bottom\_line) vừa tìm thấy sau cùng,lặp lại các bước a,b để xác định các giới hạn của các dòng tiếp theo , cho đến khi duyệt hết chiều cao của ảnh thì dừng, quá trình xác định dòng ký tự hoàn tất.

*Ảnh đầu vào*



Quá trình tách dòng

**Giới hạn trên**

**Giới hạn dưới**

**1.2.Tách từng kí tự**

* Thuật toán :

Xét lần lượt từng dòng

1.-Bắt đầu xét từ giá trị giới hạn trên y của dòng (top\_line) và giá trị x đầu tiên (x=0)

2.Xác định giới hạn trên của ký tự : Quét hết chiều rộng của ảnh, trên cùng giá trị y.

+ Nếu phát hiện pixel đen thì đánh dấu y là giá trị đỉnh của ký tự (top\_character).Dừng quét.

+Nếu quét hết chiều rộng, mà vẫn không tìm thấy pixel đen nào thì tăng y và reset lại x, tiếp tục thực hiện lại bước 2.

3. Xác định giới hạn dưới của ký tự :Bắt đầu duyệt từ giới hạn trên (đỉnh ) vừa tìm thấy của ký tự (0,top\_character)

+ Tương tự như xác định giới hạn trên , ta duyệt hết chiều rộng của ảnh trên cùng một giá trị y.

+Nếu duyệt hết dòng mà không tìm thấy ký tự pixel đen nào thì ghi nhận y-1 là giới hạn dưới của ký tự (bottom\_character).Dừng duyệt.

+Nếu chưa tìm thấy bottom\_character, tiếp tục duyệt đến dòng tiếp theo (tăng y, reset x=0).

4.Xác định giới hạn trái của ký tự (xác định giá trị của x )

-Bắt đầu từ giới hạn trên (đỉnh của ký tự - top\_character), giá trị x đầu tiên (x=0) .

-Quét đến giới hạn dưới của dòng (bottom\_character), giữ nguyên x (quét theo chiều thẳng đứng )

+Nếu gặp pixel đen đầu tiên, ghi nhận x là giới hạn trái của kí tự (left-character).Dừng quét.

+Nếu quét đến cuối giới hạn dưới, vẫn không tìm thấy pixel đen nào, thì reset lại y = giới hạn trên vừa tìm thấy, tăng x lên.(x++),và tiếp tục thực hiện lại bước 4.

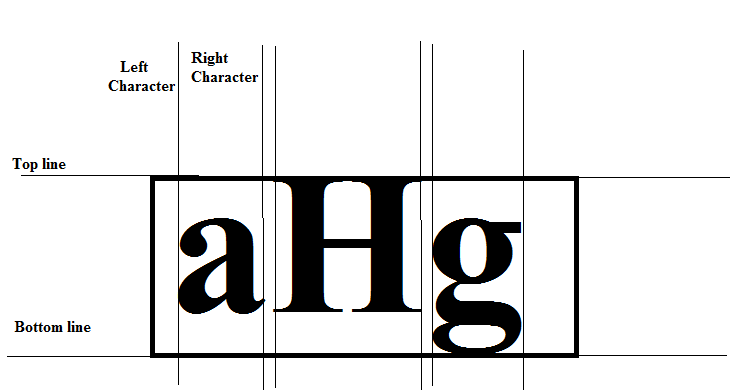
5.Xác định giới hạn phải của ký tự (xác định giá trị x )

-Bắt đầu từ giới hạn trên -đỉnh của ký tự (top\_character), giới hạn trái của ký tự (left\_character,top\_character) .Quét theo chiều thẳng đứng đến giới hạn dưới của dòng.

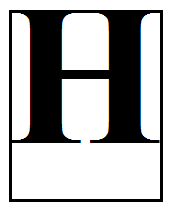
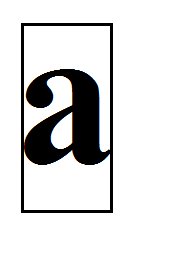
+Nếu khi quét hết chiều đứng ,mà vẫn không gặp được pixel đen nào thì ghi nhận x-1 là giới hạn phải của ký tự (right\_character).Dừng quét.

+Nếu gặp pixel đen thì tăng x (x++) và reset lại y =top\_character của ký tự đang xét ,để xét đường thẳng đứng tiếp theo.

6. Lặp lại bước 1 đến 5 để xác định giới hạn của ký tự tiếp theo trên cùng dòng.Với y=top\_line của dòng đang xét và giá trị x đầu tiên = right\_character của ký tự vừa tìm thấy.



Sau khi tách ký tự

**1.3.Xác Định Giới Hạn Chính Xác Cho Từng Ký Tự**

Trong một số trường hợp, việc tách ký tự chưa thực sự chính xác (giống như ví dụ trên ).Ta cần xác định lại một lần nữa giới hạn chính xác cho từng ký tự.

* **Thuật toán :**

Xét từng ký tự :

***1.Xác định đỉnh ký tự***

-Bắt đầu từ đỉnh của ký tự tạm thời,và giới hạn trái của ký tự vừa xác định ở trên( left\_character,top\_character).Quét đến giới hạn phải của ký tự, giữ nguyên y.

+Nếu tìm thấy pixel đen thì chọn lại y là giới hạn đỉnh của ký tự (top\_character=y).Dừng quét.

+Nếu không , xét điểm tiếp theo.

+Nếu không tìm thấy pixel đen nào thì tăng y (y++) , reset lại x .(x= left\_character).Thực hiện lại bước 1.

***2.Xác định giới hạn dưới của ký tự.***

- Bắt đầu từ giới hạn dưới hiện thời của ký tự (được xác định ở trên –phần 1.2 ), và giới hạn trái của ký tự (left\_character, bottom\_character).Quét x đến giới hạn phải của ký tự, trên cùng giá trị y .

+ Nếu tìm thấy pixel đen ,đánh dấu y là giới hạn dưới của ký tự ( bottom\_character= y).Ngừng quét.

+Nếu không tìm thấy pixel đen nào, thì reset lại x (= left character ), giảm y (y--).Thực hiện lại bước 2.

Xác định giới hạn

Giới hạn trên

Giới hạn dưới

**1.4.Ánh Xạ Ký Tự Ảnh Vào Ma Trận Giá Trị**

Sau khi tách, ký tự ảnh sẽ được ánh xạ vào ma trận 2 chiều, có kích thước 15 x 10. Vì các ký tự ảnh dù đã chuẩn hóa, chọn lọc nhưng sau khi tách đều có kích thước khá lớn so với ma trận ( khoảng 58x 74), nên các ký tự ảnh này đều phải qua 1 bước chuyển đổi tỉ lệ trước khi chuyển giá trị vào ma trận nhị phân 15 x10.

* Lưu ý : Việc chuyển đổi tỉ lệ này là một trong những nguyên nhân gây ra sai số dẫn đến nhận dạng ký tự bị sai.

**Thuật toán** : ánh xạ ký tự ảnh vào ma trận pixel ảnh.

***a.Xét chiều rộng***

-Ánh xạ điểm đầu và điểm cuối của ký tự ảnh tương ứng với giá trị đầu và cuối của ma trận .

-Khởi tạo với 10 phần tử tương ứng.

-Ánh xạ tọa độ điểm đầu (0,y) và điểm cuối (width,y) của ảnh kí tự tương ứng với giá trị đầu (0,y) và giá trị cuối (10,y) của ma trận.

-Chia nhỏ chiều rộng thành 10 mốc giá trị, ánh xạ vào chiều rộng của ma trận pixel.

Ví dụ :

xMT[ 0]=0;

xMT[5]=width/2.

xMT[10]=width.

xMT[2]=xMT[5]/2.

.....

***b.Đối với chiều cao:***

Khởi tạo với 15 phần tử tương ứng.

Ánh xạ điểm đầu (x,0) và điểm cuối (x,height) của ảnh kí tự tương ứng với giá trị đầu (x,0) và giá trị cuối (x,15) của ma trận.

Chia nhỏ chiều cao thành 15 mốc giá trị, ánh xạ tương ứng vào chiều cao của ma trận pixel.

Vd: yMT[ 0]=0;

yMT[5]=height /3.

xMT[10]=height x 2/3.

xMT[15]=height.

...

-Lấy các giá trị pixel trong file ảnh tuyến tính hóa vào trong ma trận pixel bằng cách kết hợp tọa độ chiều cao và chiều rộng.

Vd:MT[i,j]=img[xMT[i],yMT[j] ]



Hình : Ánh xạ lưới ký tự.

***c.Chuyển ma trận lưới pixel thành ma trận giá trị***

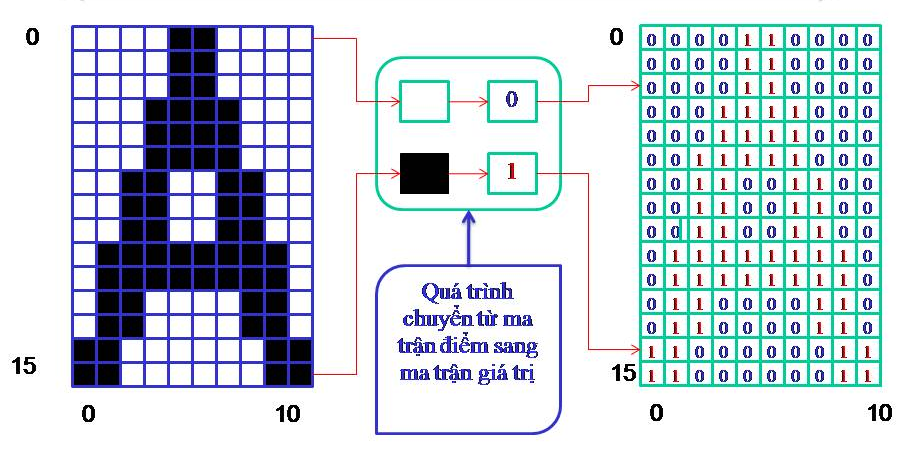
-Bắt đầu từ tọa độ (0,0) của ma trận pixel.Giữ y , duyệt qua x:

Nếu pixel đen thì ghi nhận giá trị 1 vào ma trận giá trị có tọa độ tương ứng.

Nếu pixel trắng thì ghi nhận giá trị 0 vào ma trận giá trị có tọa độ tương ứng.

Nếu x= chiều rộng thì reset lại x=0 , tăng y (y++).

***Từ ma trận ảnh này ta ánh xạ thành mảng 1 chiều 150 cột 1 dòng, và chuyển thành 150 noron đầu vào.***



**2.Huấn Luyện Mạng**

* **Thuật toán :**

1.Xây dựng cấu trúc mạng với mô hình 3 lớp .

2.Khởi tạo các giá trị ngẫu nhiên cho trọng số trong khoảng [-độ lệch,độ lệch ]

3.Nạp file ảnh đầu vào và file huấn luyện (file text đầu ra mong muốn ).

4.Phân tích ảnh và ánh xạ thành 150 noron đầu vào.

5.Đọc các kí tự đầu ra mong muốn, chuyển thành giá trị nhị phân Unicode và lưu trữ lại.

6.Với mỗi kí tự ảnh :

+Tính giá trị đầu ra trong mạng FeedForward.

+So sánh với ký tự tương ứng ở đầu ra mong muốn và tính toán lỗi.

+Lan truyền ngược lỗi và điều chỉnh lại giá trị của trọng số .

7.Chuyển đến xét ký tự ảnh tiếp theo thực hiện lại bước 6 , cho đến khi duyệt hết các ký tự ảnh.

8.Tính giá trị lỗi trung bình cho tất cả các ký tự ảnh.

9. Lặp lại bước 6 đến bước 8 cho đến khi chạy đủ số vòng dạy

+Nếu lỗi trung bình đạt thấp hơn giá trị ngưỡng của lỗi thì dừng .

+Nếu không thì tiếp tục.

Xây dựng mạng

Khởi tạo trọng số

Nạp file huấn luyện

Phân tích ảnh

Phát hiện kí tự tiếp theo

Các kí tự sẵn sàng ?

Các kí tự sẵn sàng ?

Đọc đầu ra mong muốn

Tính đầu ra của mạng

Tính lỗi

Các kí tự sẵn sàng ?

Lỗi <Ngưỡng ?

Số lần lặp =max epochs ?

Tính trung bình lỗi

Vector đầu vào tiếp theo

Đ

S

**3.Nhận Dạng Ký Tự Quang Học**

* Thuật toán :

-Nạp file ảnh đầu vào

-Phân tích ảnh thành từng ký tự ảnh.

-Đưa ma trận giá trị cho đầu vào mạng feed-forward , tính toán giá trị đầu ra dựa trên các trọng số đã được lưu sẵn khi tiến hành huấn luyện mạng.

-Chuyển mã nhị phân Unicode thành ký tự tương ứng.

Đ

S

Đ

S

Tính toán đầu ra

Dòng đầu tiên

Ánh xạ kí tự tới vector

Phân tích dòng ảnh

Chuyển từ Unicode sang kí tự

Hết kí tự trên dòng ?

Hết Các dòng ?

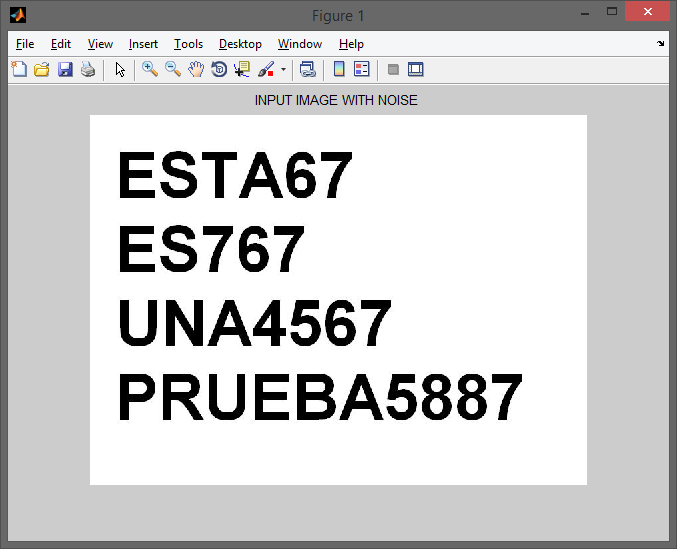
Dòng tiếp theo

Kí tự tiếp theo

**4. Các bước nhận dạng kí tự trong matlab**

1. **Nhập vào hình ảnh chứa các kí tự cần nhận dạng:**

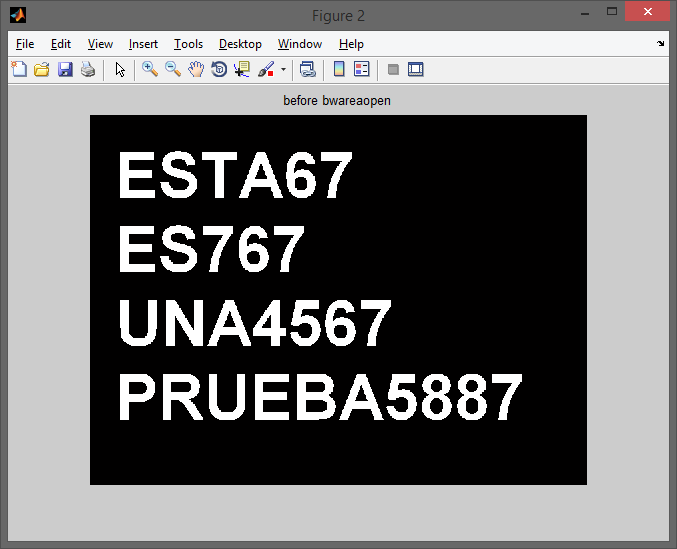
[filename, pathname] = uigetfile('\*','LOAD AN IMAGE');

imagen=imread(fullfile(pathname, filename));

1. **Chuyển hình ảnh sang dạng ảnh nhị phân:**

threshold = graythresh(imagen);

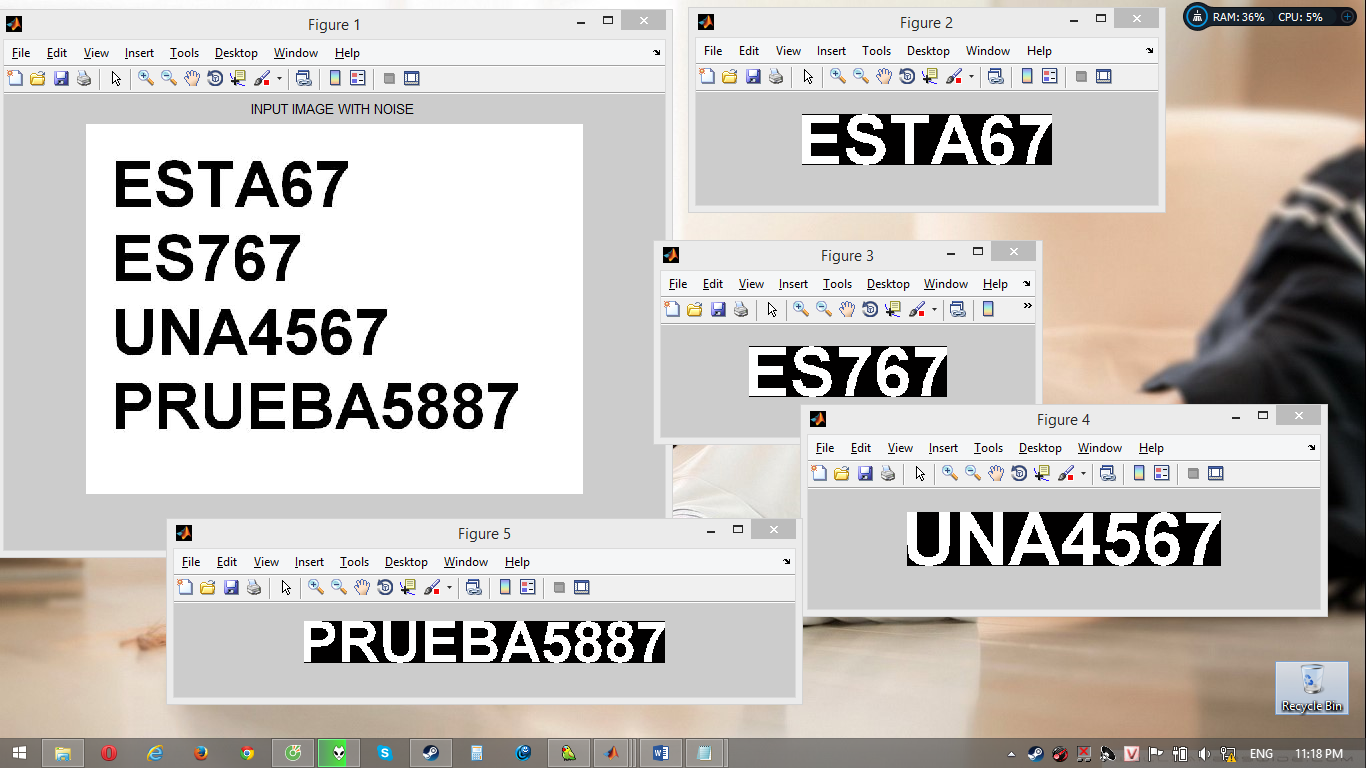
imagen =~im2bw(imagen,threshold);



1. **Cắt hình ảnh thành từng dòng nếu ảnh chứa nhiều dòng có kí tự:**

Sử dụng hàm lines\_crop

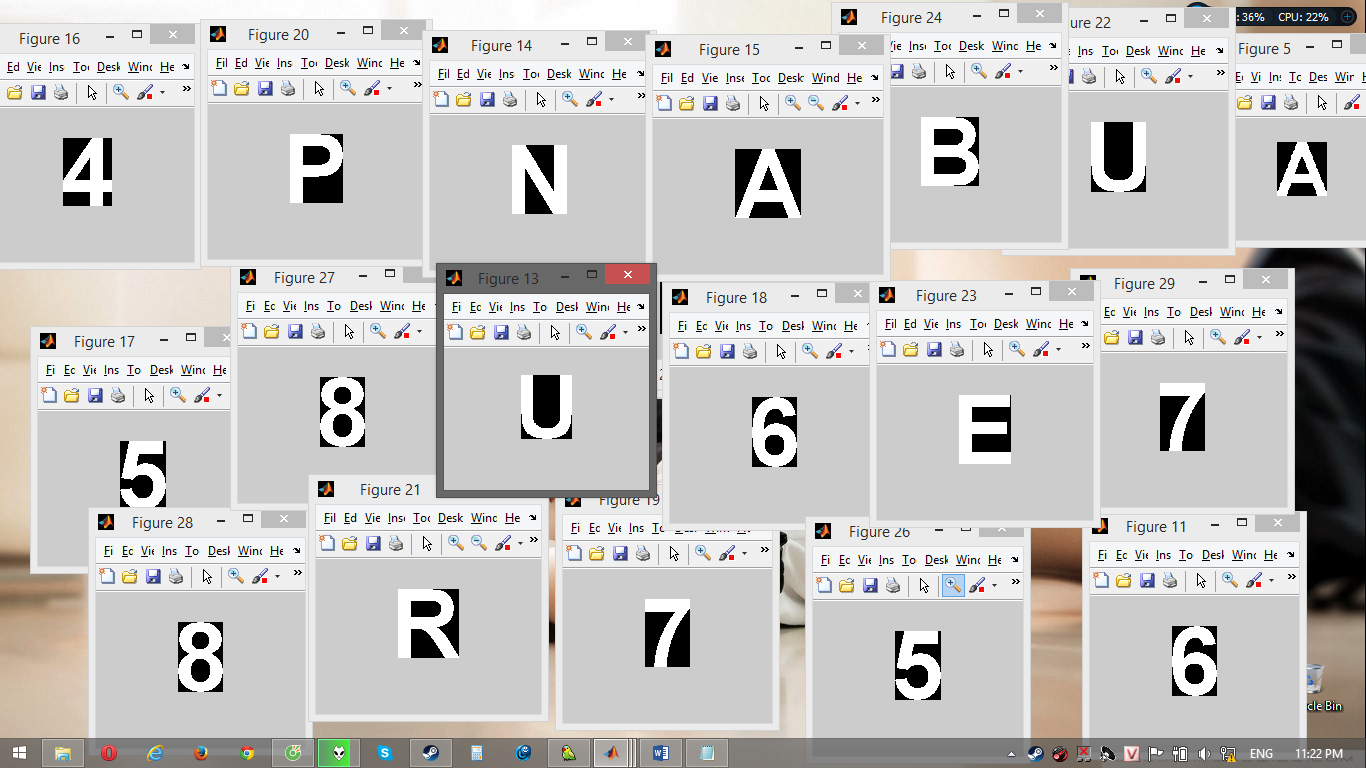
Kết quả sau khi cắt:



1. **Cắt các dòng kí tự thành các kí tự riêng biệt:**

Sử dụng hàm letter\_crop

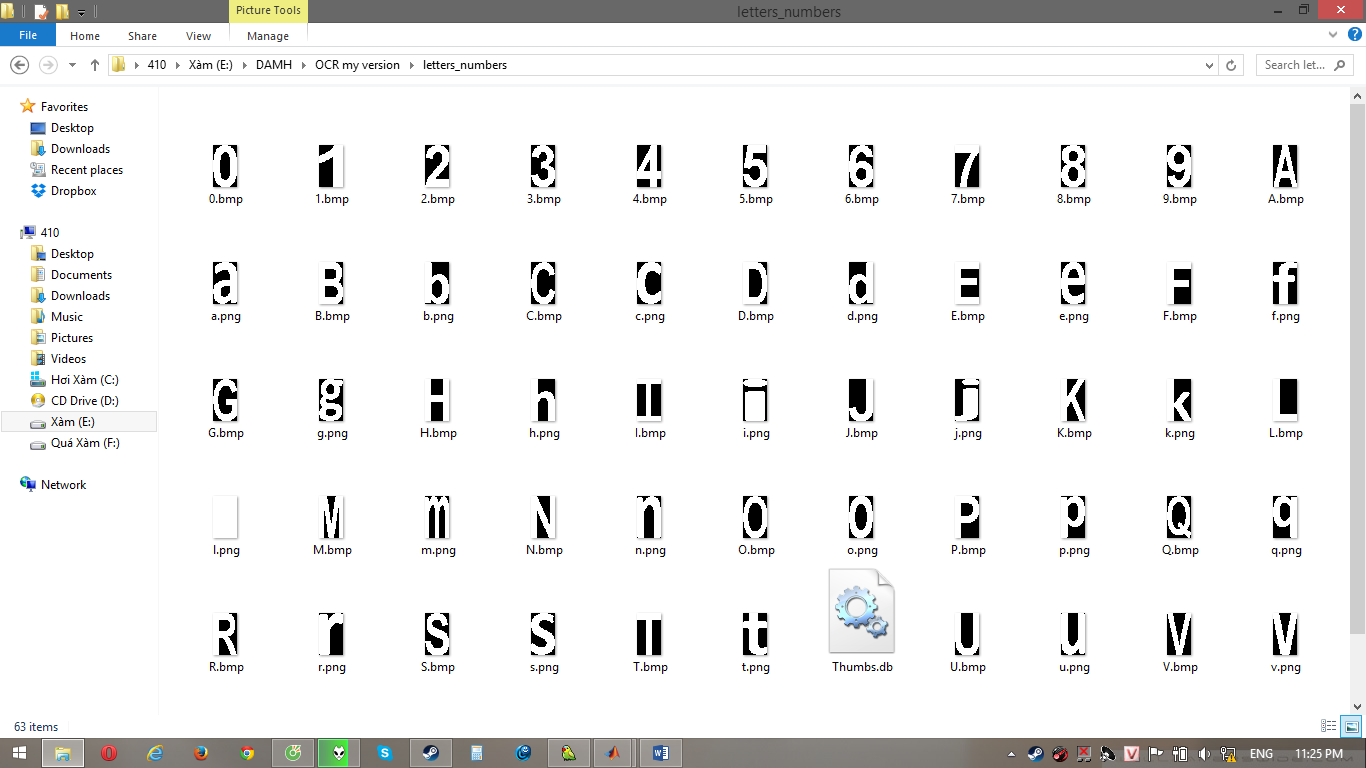
Kết quả sau khi cắt:

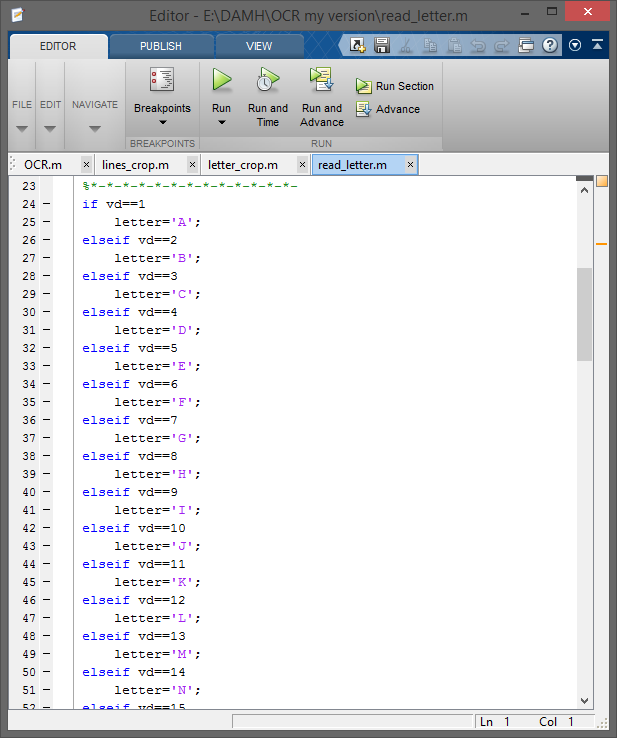


1. **Phần mềm sẽ so sánh ma trận của từng ảnh nhị phân với mẫu đã được học trước nhằm chọn ra kí tự tương đồng nhất với từng kí tự trong ảnh:**

Sử dụng hàm read\_letter

Mẫu các chữ cái và số mà phần mềm đã được học:

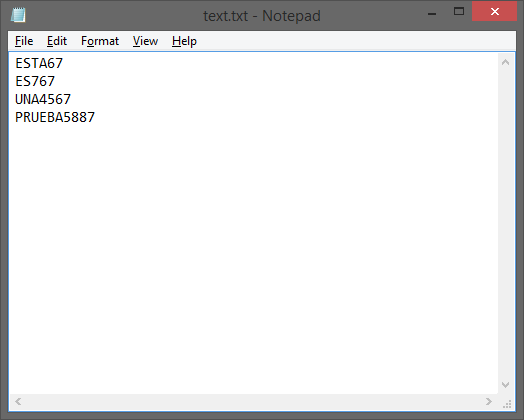




1. **Mở một file text để đưa nội dung đã được nhận dạng vào:**

Sử dụng hàm fopen('text.txt', 'wt');

Xuất file text này sau khi xác nhận đã hết kí tự nhận dạng



**Trong matlab sẽ hỗ trợ các công cụ Image Processing Toolbox giúp xử lý và biển đổi hình ảnh và dạng ma trận số để có thể so sánh với mẫu huấn luyện ban đầu**

***Tài liệu tham khảo:***

 Schantz, Herbert F. (1982). *The history of OCR, optical character recognition*. [Manchester Center, Vt.]: Recognition Technologies Users Association.[ISBN](http://en.wikipedia.org/wiki/International_Standard_Book_Number) [9780943072012](http://en.wikipedia.org/wiki/Special:BookSources/9780943072012).

'Albe, E. E. F. (1 July 1914). "On a Type-Reading Optophone". *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **90** (619): 373–375.[doi](http://en.wikipedia.org/wiki/Digital_object_identifier):[10.1098/rspa.1914.0061](http://dx.doi.org/10.1098%2Frspa.1914.0061).

 PrintToBraille Tool. ["ocr-gui-frontend"](https://code.google.com/p/ocr-gui-frontend/). MILE Lab, Dept of EE, IISc.

Tappert, C. C.; Suen, C. Y.; Wakahara, T. (1990). "The state of the art in online handwriting recognition". *IEEE Transactions on Pattern Analysis and Machine Intelligence***12** (8): 787. [doi](http://en.wikipedia.org/wiki/Digital_object_identifier):[10.1109/34.57669](http://dx.doi.org/10.1109%2F34.57669). [edit](http://en.wikipedia.org/w/index.php?title=Template:Cite_doi/10.1109.2F34.57669&action=edit&editintro=Template:Cite_doi/editintro2)

["Optical Character Recognition (OCR) – How it works"](http://www.nicomsoft.com/optical-character-recognition-ocr-how-it-works/). Nicomsoft.com.

["How OCR Software Works"](http://ocrwizard.com/ocr-software/how-ocr-software-works.html). OCRWizard. Retrieved 2013-06-16.

["How does OCR document scanning work?"](http://www.explainthatstuff.com/how-ocr-works.html). Explain that Stuff. 2012-01-30.

["OCR Introduction"](http://www.dataid.com/aboutocr.htm). Dataid.com. Retrieved 2013-06-16.

["Future Challenges in Handwriting and Computer Applications"](http://users.erols.com/rwservices/pens/biblio88.html#Suen88). 3rd International Symposium on Handwriting and Computer Applications, Montreal, May 29, 1987.

**Cùng một số tài liệu, tiểu luận tiếng việt.**