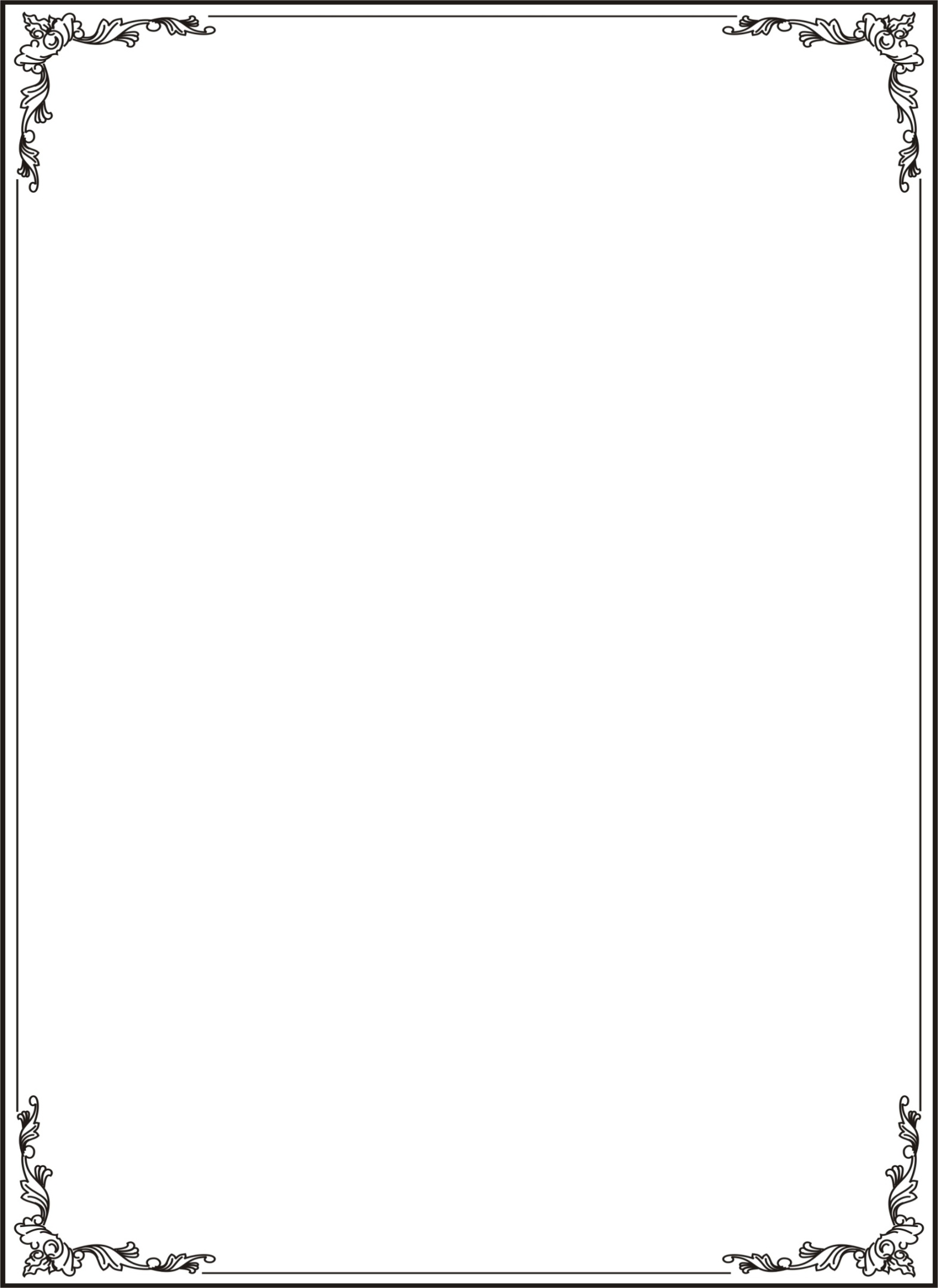
**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA TOÁN – CƠ – TIN HỌC**



**BÁO CÁO XỬ LÝ ẢNH**

***Môn: Xử Lý Ảnh***

**ĐỀ TÀI:** Khoanh vùng và nhận dạng chữ xuất hiện trên ảnh

Thành viên:

Nguyễn Quốc Cường

Trần Đức Dũng

Nguyễn Văn Hinh

Trần Thanh Hóa

Lê Văn Lượng

Nguyễn Thị Oanh

Phạm Thị Hồng Thoa

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Mục lục:**   1. **Phân công công việc** 2. **Giới thiệu chung** 3. **Các bước khoanh vùng và nhận dạng chữ trong ảnh**   3.1) Sử dụng kỹ thuật MSER (Maximally Stable Extremal Regions) và OCR (Optical Character Recognition )  3.2) Nhận dạng ký tự ( character recognition)  3.3) Hậu xử lý (Post-Processing)   1. **Ứng dụng nhận dạng văn bản trên ảnh**   4.1) Sử dụng OpenCV, MSER, pytesseract (OCR)  4.2) Sử dụng OpenCV, pytesseract (OCR)  **5, Tài liệu tham khảo**     1. **Phân công công việc:**  |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Tên thành viên** | **Tìm hiểu  thuật toán** | **Code  Matlab** | **Code  Python** | **Làm  báo cáo** | | Nguyễn Quốc Cường | **x** | **x** |  | **x** | | Trần Đức Dũng | **x** | **x** |  |  | | Nguyễn Văn Hinh | **x** | **x** | **x** |  | | Trần Thanh Hóa | **x** | **x** |  | **x** | | Lê Văn Lượng | **x** | **x** | **x** | **x** | | Nguyễn Thị Oanh | **x** | **x** |  |  | | Phạm Thị Hồng Thoa | **x** | **x** |  | **x** | | 5 |
|  |  |

1. **Giới thiệu chung :**

Nhận dạng chữ xuất hiện trong ảnh là chuyển đổi các [hình ảnh](https://en.wikipedia.org/wiki/Image)văn bản đã được đánh máy, viết tay hay in thành văn bản mã hoá bằng máy, dù là từ tài liệu được quét, ảnh tài liệu, ảnh chụp cảnh ví dụ như văn bản trên các bảng hiệu và biển quảng cáo trong một bức ảnh phong cảnh) hoặc từ văn bản phụ đề chồng lên một hình ảnh (ví dụ như từ một chương trình truyền hình). Nó được sử dụng rộng rãi như là một hình thức nhập thông tin từ các bản ghi dữ liệu giấy in, tài liệu hộ chiếu, hóa đơn, bảng sao kê ngân hàng, biên lai trên máy vi tính, danh thiếp, thư, bản in của dữ liệu tĩnh hoặc bất kỳ tài liệu phù hợp nào. Đây là một phương pháp phổ biến để số hóa văn bản in để chúng có thể được chỉnh sửa, tìm kiếm, lưu trữ gọn hơn, hiển thị trực tuyến, và được sử dụng trong các quy trình máy như [máy tính nhận thức](https://en.wikipedia.org/wiki/Cognitive_computing) , [dịch máy](https://en.wikipedia.org/wiki/Machine_translation) , trích xuất [văn bản thành giọng nói](https://en.wikipedia.org/wiki/Text-to-speech) , dữ liệu chính và [khai thác văn bản](https://en.wikipedia.org/wiki/Text_mining)

1. **Các bước khoanh vùng và nhận dạng chữ trong ảnh**

**3.1 Sử dụng kỹ thuật MSER (Maximally Stable Extremal Regions) và OCR (Optical Character Recognition )**

**Bước 1: Phát hiện vùng văn bản sử dụng MSER**

Thuật toán MSER hoạt động tốt trong việc phát hiện vùng văn bản bởi màu sắc và độ tương phản cao văn bản.

Sử dụng hàm dectectMSERFeatures để tìm tất cả các vùng trong hình ảnh và tô màu những kết quả này. Chú ý rằng có rất nhiều vùng không phải văn bản cũng tô màu.

colorImage = imread('handicapSign.jpg');

I = rgb2gray(colorImage);

% Detect MSER regions.

[mserRegions, mserConnComp] = detectMSERFeatures(I, ...

'RegionAreaRange',[200 8000],'ThresholdDelta',4);

figure

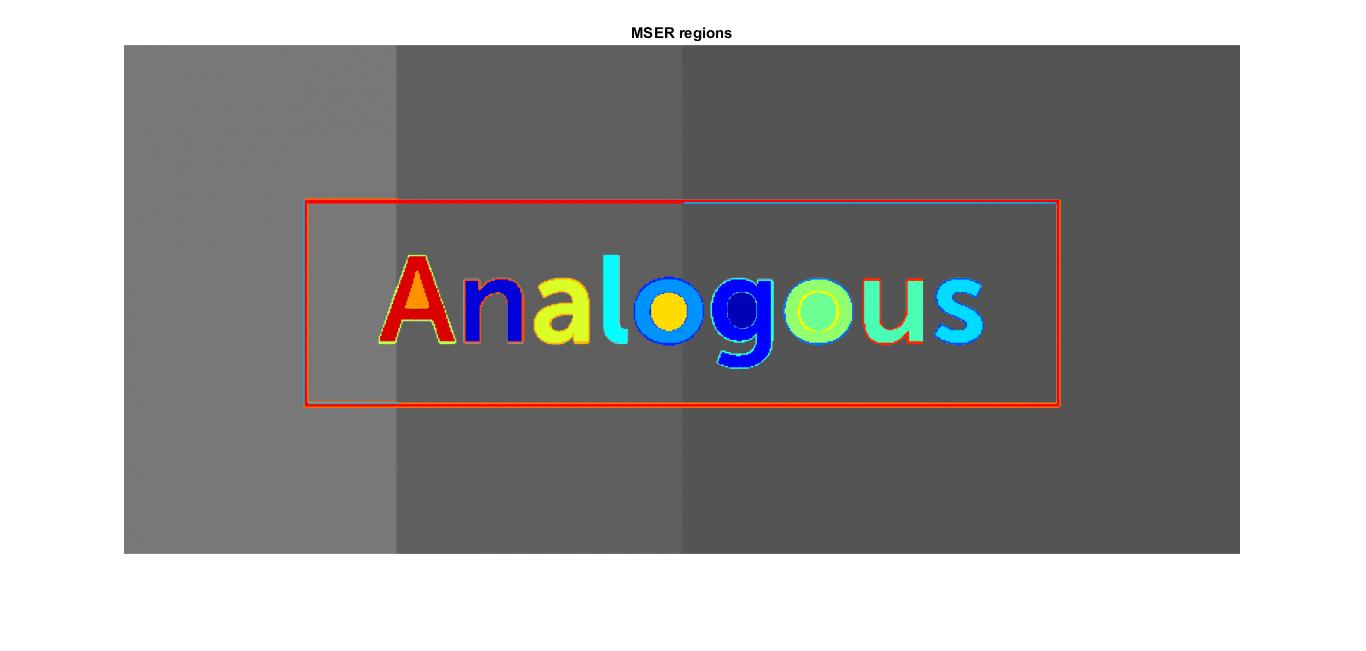
imshow(I)

hold on

plot(mserRegions, 'showPixelList', true,'showEllipses',false)

title('MSER regions')

hold off

****

**Bước 2: Loại bỏ các vùng không phải văn bản dựa trên tính chất hình học cơ bản**

Mặc dù thuật toán MSER chọn ra hầu hết các vùng văn bản nhưng nó cũng phát hiện ra nhiều khu vực khác mà không phải văn bản.

Chúng ta sẽ sử dụng một số quy tắc để loại bỏ vùng phi văn bản. Ví dụ, ta có thể dùng các tính chất hình học của văn bản để loại bỏ các vùng không phải văn bản. Ngoài ra còn có thể sử dụng học máy để huấn luyện từ điển. Thông thường ta sẽ kết hợp 2 cách để cho kết quả tốt nhất.

Có rất nhiều tính chất hình học tốt cho việc phân biệt giữa các vùng văn bản và phi văn bản bao gồm: Tỉ lệ, Độ lệch tâm, Số Euler,...

% Use regionprops to measure MSER properties

mserStats = regionprops(mserConnComp, 'BoundingBox', 'Eccentricity', ...

'Solidity', 'Extent', 'Euler', 'Image');

% Compute the aspect ratio using bounding box data.

bbox = vertcat(mserStats.BoundingBox);

w = bbox(:,3);

h = bbox(:,4);

aspectRatio = w./h;

% Threshold the data to determine which regions to remove. These thresholds

% may need to be tuned for other images.

filterIdx = aspectRatio' > 3;

filterIdx = filterIdx | [mserStats.Eccentricity] > .995 ;

filterIdx = filterIdx | [mserStats.Solidity] < .3;

filterIdx = filterIdx | [mserStats.Extent] < 0.2 | [mserStats.Extent] > 0.9;

filterIdx = filterIdx | [mserStats.EulerNumber] < -4;

% Remove regions

mserStats(filterIdx) = [];

mserRegions(filterIdx) = [];

% Show remaining regions

figure

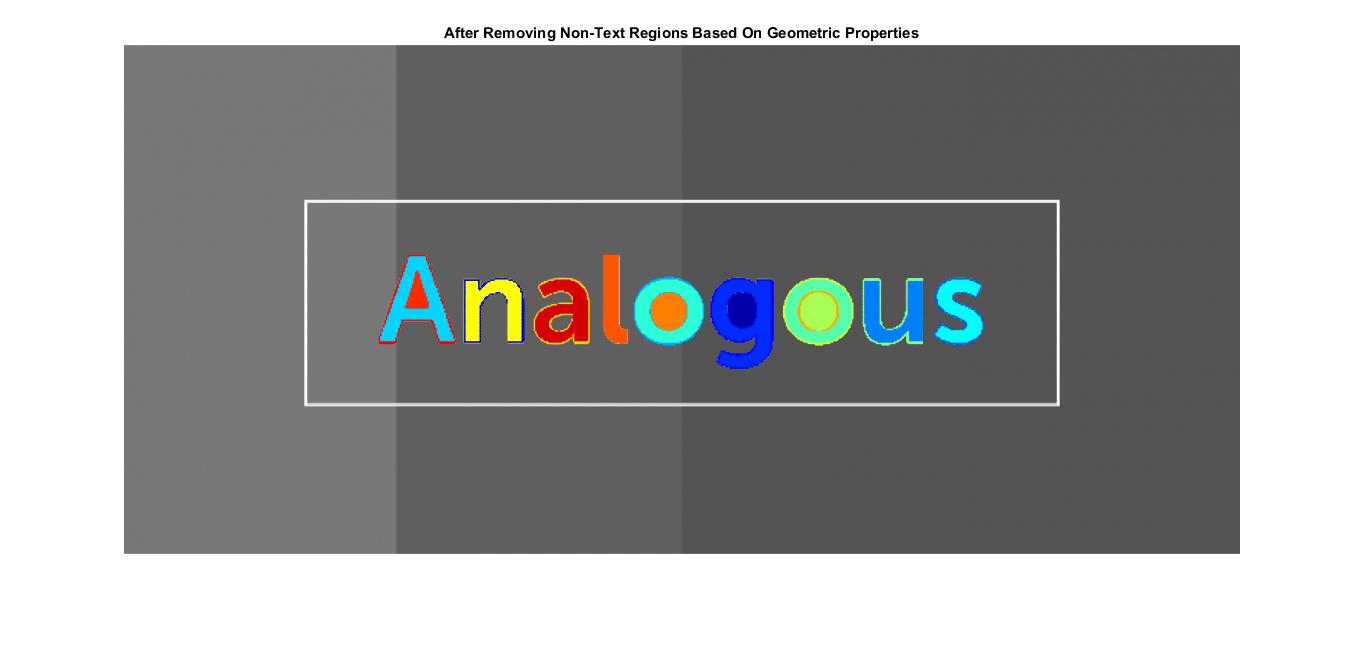
imshow(I)

hold on

plot(mserRegions, 'showPixelList', true,'showEllipses',false)

title('After Removing Non-Text Regions Based On Geometric Properties')

hold off



**Bước 3: Xóa bỏ các vùng không phải văn bản dựa trên sự biến đổi độ rộng của đường kẻ**

Một thông số thường được dùng để phân biệt giữa các văn bản và phi văn bản là độ rộng đường kẻ. Độ rộng đường kẻ là một thước đo độ rộng của đường cong và đường thẳng tạo nên đối tượng. Các vùng kí tự ít có biết đổi về độ rộng đường kẻ hơn là vùng không phải kí tự.

% Get a binary image of the a region, and pad it to avoid boundary effects

% during the stroke width computation.

regionImage = mserStats(6).Image;

regionImage = padarray(regionImage, [1 1]);

% Compute the stroke width image.

distanceImage = bwdist(~regionImage);

skeletonImage = bwmorph(regionImage, 'thin', inf);

strokeWidthImage = distanceImage;

strokeWidthImage(~skeletonImage) = 0;

% Show the region image alongside the stroke width image.

figure

subplot(1,2,1)

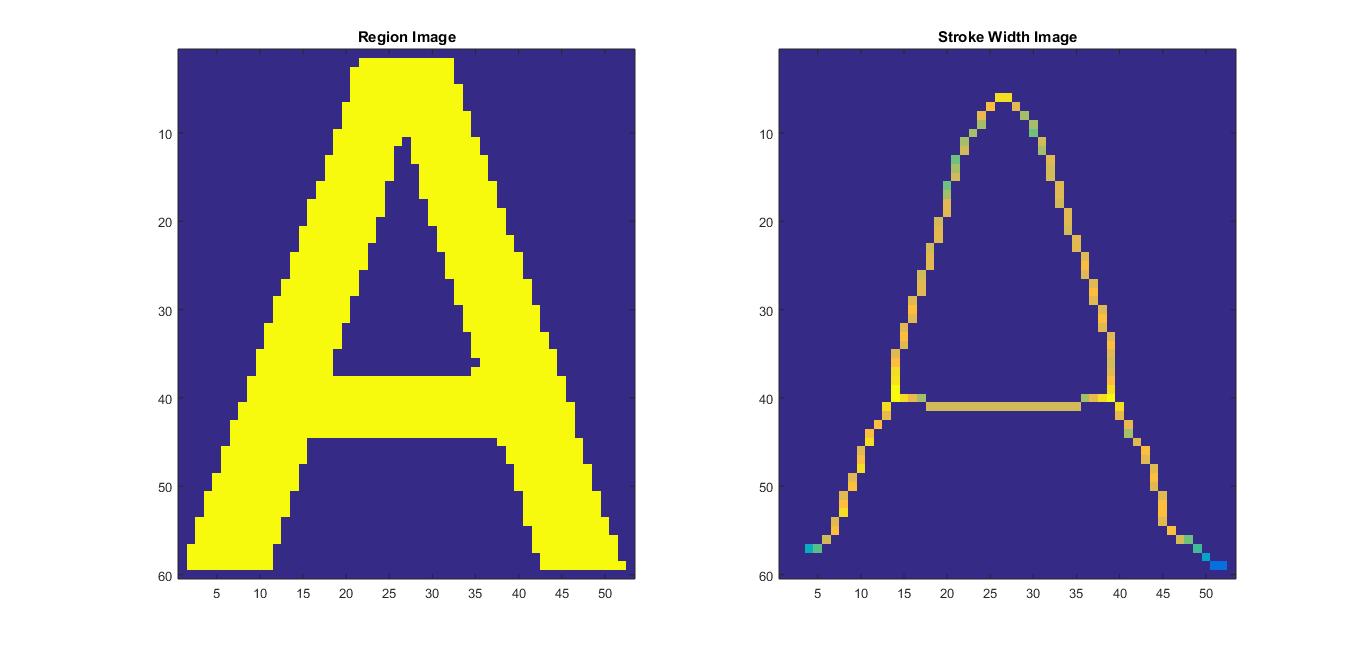
imagesc(regionImage)

title('Region Image')

subplot(1,2,2)

imagesc(strokeWidthImage)

title('Stroke Width Image')



Để hiểu cách thức sử dụng độ rộng đường kẻ để xóa bỏ vùng phi văn bản cần ước tính độ rộng đường kẻ của một trong các vùng MSER đã được phát hiện.Ta có thể làm điều này bằng các sử dụng phương pháp biến đổi khoảng cách và nhị phân mỏng.

Để sử dụng biến đổi độ rộng đường kẻ để loại bỏ vùng phi văn bản cần sử dụng một giá trị ngưỡng, sự thay đổi trên toàn bộ khu vực phải được định lượng vào một thước đo duy nhất như sau:

% Compute the stroke width variation metric

strokeWidthValues = distanceImage(skeletonImage);

strokeWidthMetric = std(strokeWidthValues)/mean(strokeWidthValues);

Sau đó một ngưỡng của độ rộng có thể được thêm vào để xóa vùng không phải văn bản. Chú ý rằng giá trị ngưỡng có thể thay đổi cho các ảnh khác nhau với các kiểu chữ khác nhau

% Threshold the stroke width variation metric

strokeWidthThreshold = 0.4;

strokeWidthFilterIdx = strokeWidthMetric > strokeWidthThreshold;

Đọan code trên phải được áp dụng riêng cho mỗi khu vực MSER được phát hiện. Cho vòng lặp để xử lý tất cả các vùng, sau đó ta sẽ thấy được kết quả của việc loại bỏ các vùng phi văn bản sử dụng biến đổi độ rộng đường kẻ

% Process the remaining regions

for j = 1:numel(mserStats)

regionImage = mserStats(j).Image;

regionImage = padarray(regionImage, [1 1], 0);

distanceImage = bwdist(~regionImage);

skeletonImage = bwmorph(regionImage, 'thin', inf);

strokeWidthValues = distanceImage(skeletonImage);

strokeWidthMetric = std(strokeWidthValues)/mean(strokeWidthValues);

strokeWidthFilterIdx(j) = strokeWidthMetric > strokeWidthThreshold;

end

% Remove regions based on the stroke width variation

mserRegions(strokeWidthFilterIdx) = [];

mserStats(strokeWidthFilterIdx) = [];

% Show remaining regions

figure

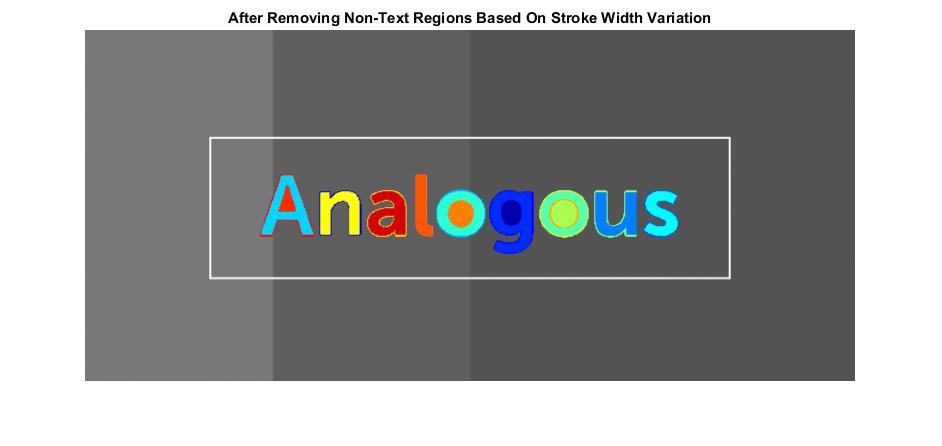
imshow(I)

hold on

plot(mserRegions, 'showPixelList', true,'showEllipses',false)

title('After Removing Non-Text Regions Based On Stroke Width Variation')

hold off



**Bước 4: Gộp các vùng văn bản để trả về kết quả nhận dạng**

Tại thời điểm này, kết quả trả về sẽ bao gồm tất cả các kí tự của văn bản. Để sử dụng kết quả cho việc nhận dạng (dùng OCR) các kí tự cần được gộp vào các từ hoặc dòng văn bản. Điều này cho phép nhận biết các từ trong một hình ảnh thay vì từng kí tự. VD: Nhận dạng chuỗi 'EXIT' thay vì các kí tự 'E','X','T','I' không theo thứ tự và không có nghĩa.

Một cách tiếp cận cho việc gộp các kí tự vào từ là đầu tiên tìm các vùng biên của kí tự sau đó tạo ra các khung bao xung quanh các từ ấy. Để tìm các vùng lân cận, mở rộng khung bao được tính toán trước đó khi dùng hàm 'regionprops'. Điều này sẽ làm các khung bao quanh các kí tự liền kề nhau sẽ chồng chéo lên nhau, từ đó các kí tự là một phần của 1 từ hoặc một dòng sẽ tạo thành một chuỗi các khung chồng chéo.

% Get bounding boxes for all the regions

bboxes = vertcat(mserStats.BoundingBox);

% Convert from the [x y width height] bounding box format to the [xmin ymin

% xmax ymax] format for convenience.

xmin = bboxes(:,1);

ymin = bboxes(:,2);

xmax = xmin + bboxes(:,3) - 1;

ymax = ymin + bboxes(:,4) - 1;

% Expand the bounding boxes by a small amount.

expansionAmount = 0.02;

xmin = (1-expansionAmount) \* xmin;

ymin = (1-expansionAmount) \* ymin;

xmax = (1+expansionAmount) \* xmax;

ymax = (1+expansionAmount) \* ymax;

% Clip the bounding boxes to be within the image bounds

xmin = max(xmin, 1);

ymin = max(ymin, 1);

xmax = min(xmax, size(I,2));

ymax = min(ymax, size(I,1));

% Show the expanded bounding boxes

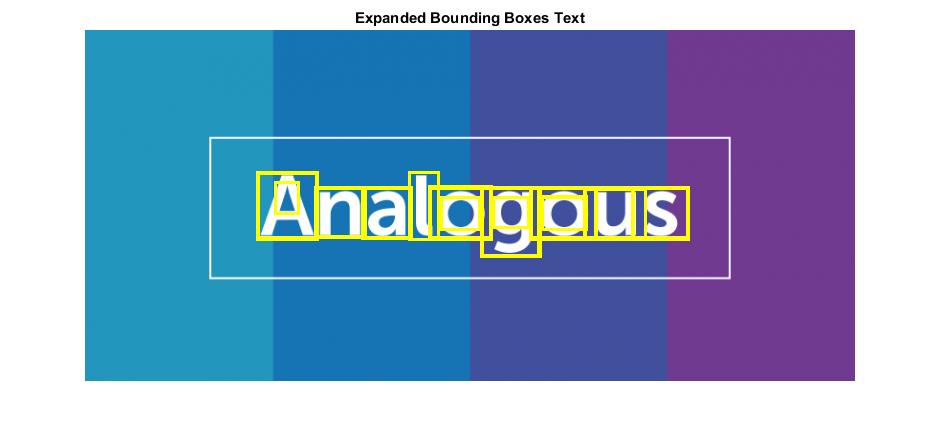
expandedBBoxes = [xmin ymin xmax-xmin+1 ymax-ymin+1];

IExpandedBBoxes = insertShape(colorImage,'Rectangle',expandedBBoxes,'LineWidth',3);

figure

imshow(IExpandedBBoxes)

title('Expanded Bounding Boxes Text')



Bây giờ, các khung đã chồng chéo lên nhau có thể gộp lại để tạo thành một khung bao quanh từng từ hoặc từng dòng văn bản. Để làm được điều này, cần tính toán tỉ lệ giao nhau giữa tất cả các khung của kí tự. Định lượng này là khoảng cách giữa tất cả các cặp của vùng kí tự do đó có thể tìm các nhóm của các vùng kí tự liền kề bằng cách tìm kiếm tỉ lệ giao nhau khác 0.

Sử dụng hàm 'bboxOverlapRatio' để tính toán tỉ lệ chồng chéo của các cặp cho tất cả các khung bao, sau đó sử dụng đồ thị để tìm tất cả các vùng đã được kết nối.

% Compute the overlap ratio

overlapRatio = bboxOverlapRatio(expandedBBoxes, expandedBBoxes);

% Set the overlap ratio between a bounding box and itself to zero to

% simplify the graph representation.

n = size(overlapRatio,1);

overlapRatio(1:n+1:n^2) = 0;

% Create the graph

g = graph(overlapRatio);

% Find the connected text regions within the graph

componentIndices = conncomp(g);

Đầu ra của hàm 'conncomp' là các chỉ số kết nối của các vùng kí tự. Sử dụng các cỉ số này để gộp các khung hàng xóm trở thành một khung bằng cách tính toán min và max...

% Merge the boxes based on the minimum and maximum dimensions.

xmin = accumarray(componentIndices', xmin, [], @min);

ymin = accumarray(componentIndices', ymin, [], @min);

xmax = accumarray(componentIndices', xmax, [], @max);

ymax = accumarray(componentIndices', ymax, [], @max);

% Compose the merged bounding boxes using the [x y width height] format.

textBBoxes = [xmin ymin xmax-xmin+1 ymax-ymin+1];

Cuối cùng trước khi đưa ra kết quả nhận dạng, ngăn việc nhận dạng văn bản sai bằng cách xóa bỏ khung được tạo thành bở một vùng kí tự. Điều này loại bỏ vùng ít có khả năng là văn bản trong thực tế (1 kí tự riêng lẻ)

% Remove bounding boxes that only contain one text region

numRegionsInGroup = histcounts(componentIndices);

textBBoxes(numRegionsInGroup == 1, :) = [];

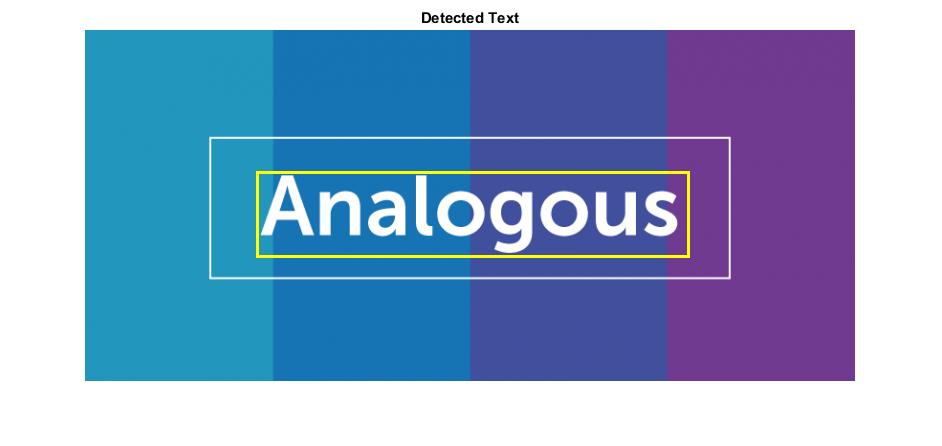
% Show the final text detection result.

ITextRegion = insertShape(colorImage, 'Rectangle', textBBoxes,'LineWidth',3);

figure

imshow(ITextRegion)

title('Detected Text')

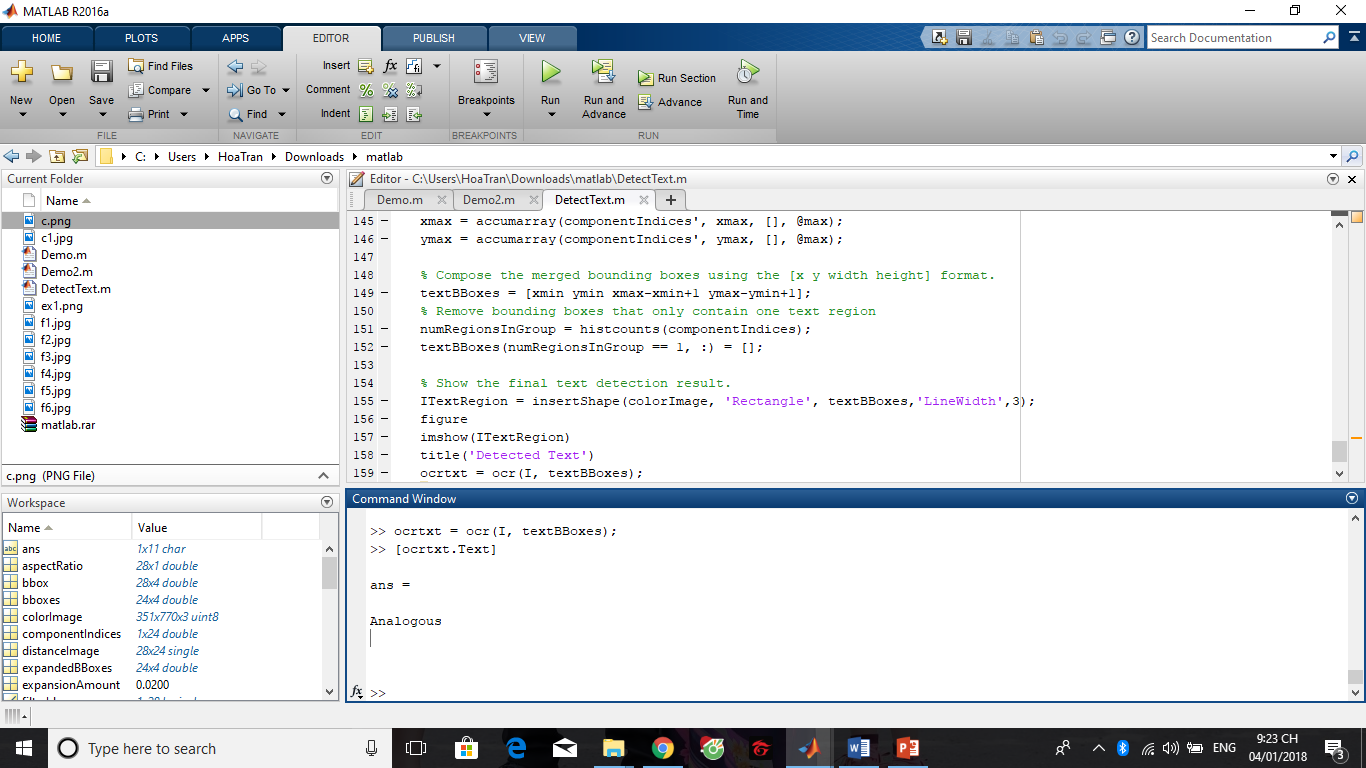


**Bước 5: Trả về văn bản nhận dạng được sử dụng OCR**

Sau khi phát hiện các vùng văn bản, sử dụng hàm OCR để trả về văn bản trong mỗi khung. Chú ý rằng nếu không tìm thấy vùng văn bản, đầu ra của OCR sẽ có nhiều nhiễu.

ocrtxt = ocr(I, textBBoxes);

[ocrtxt.Text]



 OCR là một lĩnh vực nghiên cứu về [nhận dạng khuôn mẫu](https://en.wikipedia.org/wiki/Pattern_recognition) , [trí thông minh nhân tạo](https://en.wikipedia.org/wiki/Artificial_intelligence) và [tầm nhìn máy tính](https://en.wikipedia.org/wiki/Computer_vision). Các kỹ thuật trong OCR:

Các phần mềm OCR thường thực hiện các kỹ thuật tiền xử lý ảnh để nâng cao cơ hội thành công trong nhận diện văn bản. Một số kỹ thuật phổ biến :

- De-Skew : Nếu văn bản bị nghiêng nó cần được căn chỉnh lại cho đúng

- Despeckle : Loại bỏ các điểm tiêu cực hoặc tích cực

- Binarisation: Chuyển đổi một hình ảnh từ màu hoặc [màu xám](https://en.wikipedia.org/wiki/Greyscale) sang đen trắng (gọi là " [hình nhị phân](https://en.wikipedia.org/wiki/Binary_image) " vì có hai màu). Binarisation được thực hiện như một cách đơn giản để tách văn bản (hoặc bất kỳ thành phần hình ảnh mong muốn nào khác) khỏi nền, nhiệm vụ của binarisation là cần thiết vì hầu hết các thuật toán nhận dạng thương mại chỉ hoạt động trên các hình ảnh nhị phân. Ngoài ra, hiệu quả của bước binarisation ảnh hưởng đến một mức độ đáng kể chất lượng của giai đoạn nhận dạng nhân vật và các quyết định cẩn thận được thực hiện trong việc lựa chọn các binarisation sử dụng cho một loại hình ảnh đầu vào nhất định; vì chất lượng của phương pháp binarisation được sử dụng để có được kết quả nhị phân tùy thuộc vào loại hình ảnh đầu vào (tài liệu được quét, hình ảnh văn bản cảnh, tài liệu bị suy biến trong lịch sử,... ).

- Line removal : xóa các đường và hộp không phải chữ viết hoa

- Layout analysis : phát hiện cột, đoạn, chú thích, ..

- Line and word detection : phát hiện dòng và từ

- Segmentation (Phân đoạn) : Đối với mỗi ký tự OCR, nhiều ký tự được kết nối do các hiện vật hình ảnh phải được tách; các ký tự đơn lẻ được chia thành nhiều phần do các hiện vật phải được kết nối . Việc phân đoạn các [phông chữ cố định](https://en.wikipedia.org/wiki/Fixed-pitch_font) được thực hiện tương đối đơn giản bằng cách căn chỉnh hình ảnh vào một lưới đồng nhất dựa trên nơi các đường kẻ thẳng đứng sẽ ít khi giao nhau với các vùng màu đen.

**3.2. Nhận dạng ký tự ( character recognition)**

Có hai loại cơ bản của thuật toán OCR lõi, có thể tạo ra một danh sách xếp hạng các ký tự.

Kết hợp ma trận liên quan đến việc so sánh một hình ảnh với một glyph lưu trữ trên cơ sở pixel-by-pixel; nó còn được gọi là "kết hợp mẫu", " [nhận dạng mẫu](https://en.wikipedia.org/wiki/Pattern_recognition) ", hoặc " [tương quan hình ảnh](https://en.wikipedia.org/wiki/Digital_image_correlation) ". Điều này dựa vào hình tượng đầu vào được cách ly chính xác từ phần còn lại của hình ảnh, và trên hình tượng lưu trữ được trong một phông chữ tương tự và ở cùng một quy mô. Kỹ thuật này hoạt động tốt nhất với văn bản đánh máy và không hoạt động tốt khi gặp phông chữ mới. Đây là kỹ thuật mà OCR quang học vật lý sớm được thực hiện, khá trực tiếp.

Sự khai thác tính năng phân hủy hình tượng thành "các tính năng" giống như các đường thẳng, các vòng khép kín, đường và các nút giao nhau. Các tính năng khai thác làm giảm chiều kích của biểu diễn và làm cho quá trình nhận dạng hiệu quả tính toán. Các tính năng này được so sánh với một biểu tượng đại diện vector trừu tượng của một nhân vật, có thể làm giảm đến một hoặc nhiều nguyên mẫu glyph. Các kỹ thuật chung về [phát hiện tính năng trong tầm nhìn máy tính](https://en.wikipedia.org/wiki/Feature_detection_(computer_vision)) được áp dụng cho loại OCR này, thường thấy trong [nhận dạng chữ viết](https://en.wikipedia.org/wiki/Handwriting_recognition) "thông minh" và phần mềm OCR hiện đại nhất. [phân loại hàng xóm gần nhất](https://en.wikipedia.org/wiki/Nearest_neighbour_classifiers) như [thuật toán láng giềng gần k nhất](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm)được sử dụng để so sánh các tính năng hình ảnh với các tính năng hình tượng được lưu trữ và chọn kết hợp gần nhất.

**3.3. Hậu xử lý (Post-Processing)**

Độ chính xác OCR có thể được tăng lên nếu đầu ra bị hạn chế bởi một [lexicon](https://en.wikipedia.org/wiki/Lexicon)  - một danh sách các từ được cho phép xảy ra trong một tài liệu ví dụ như khoanh vùng khu vực có văn bản, tăng độ tin cậy cho kết quả bằng cách chỉ định trước tập ký tự, ....

1. **Ứng dụng nhận dạng văn bản trên ảnh**

* Thư viện OpenCV phiên bản 3.0.4
* Thuật toán MSER
* Pytesseract (OCR)
* Python 2.7.2

**4.1 Sử dụng OpenCV, MSER, pytesseract (OCR)**

Thuật toán:

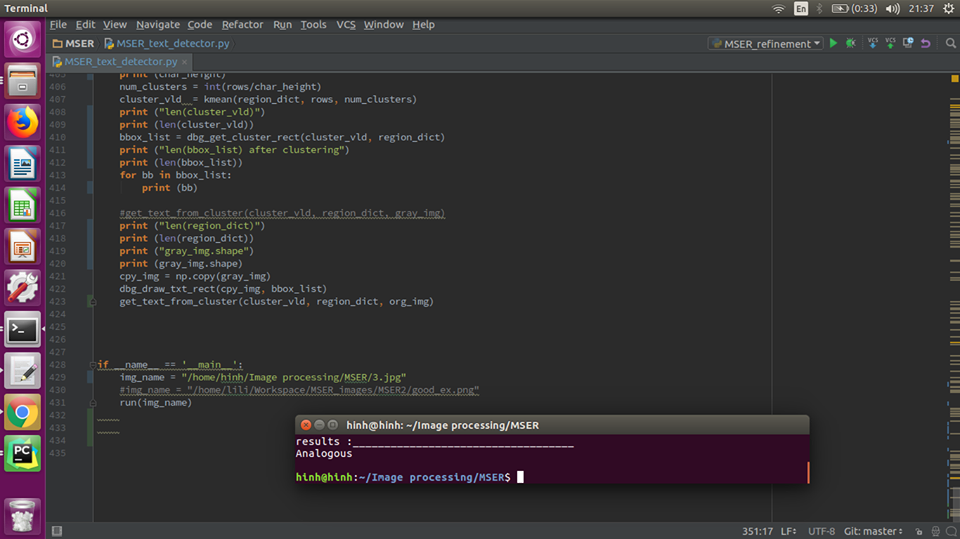
**Ảnh đầu vào**

**Dùng OCR nhận dạng văn bản trên vùng chứa**

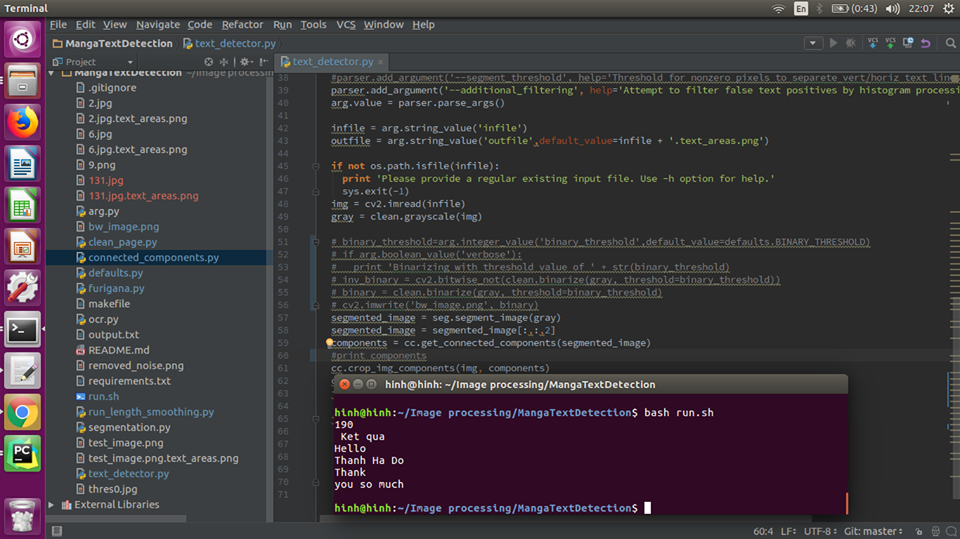
**Dùng MSER nhận dạng vùng chứa văn bản**

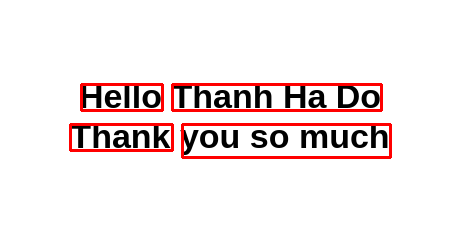
Kết quả:





**4.2 Sử dụng OpenCV, pytesseract (OCR)**





**5, Tài liệu tham khảo**

<https://www.mathworks.com/help/vision/examples/automatically-detect-and-recognize-text-in-natural-images.html>

https://www.mathworks.com/help/images/ref/bwdist.html?searchHighlight=bwdist&s\_tid=doc\_srchtitle