**Digital Image Processing (UCS-615) Project Report**

**(B.E 3rd Year May 2018)**


# Optical Character Recognition

**Submitted by:**

Abhi Mahajan - 101683033

Group: COE-1

**Submitted to:**

Dr. Jhilik Bhattacharya

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

# Extract Text from Images Using MATLAB

This MATLAB program explains you to extract text from images. This code snippet could be used for applications like license plate recognition, OCR, Text to speech convertor and other applications.

**Step 1**: The first step is to read the input image and display the input image, you will get the result as below

```
%% Read Image

Inputimage=imread('mam.jpg');

%% Show image

figure(1)

imshow(Inputimage);

title('INPUT IMAGE WITH NOISE')
```


**Sample Image**

**Step 2**: The second step is to convert the colour(RGB) image to a Gray scale

```
pkg load image

%% Convert to gray scale

if size(Inputimage,3)==3 % RGB image

 Inputimage=rgb2gray(Inputimage);

end
```

**Step 3**: The Third step is to Convert the Gray scale image to binary image

```
%Given an image img finds the optimal threshold value level for
%conversion to a binary image with im2bw. Color images are converted
%to grayscale before level is computed
%Default OTSU
%Convert to binary image
threshold = graythresh(Inputimage);
%Convert image to binary, black and white, by threshold
Inputimage =~im2bw(Inputimage,threshold);
```

**Step 4**: To remove all Boundary connected Objects which are less than 30 pixels (Area opening)

```
%Perform area opening.
%bwareaopen (bw, lim)
%Remove objects with less than lim elements from a binary image bw.
%% Remove all object containing fewer than 30 pixels
Inputimage = bwareaopen(Inputimage,30);
pause(1);
```

**Step 5**: Label all the connected components

```
% Label connected components
%bwlabeln(BW) returns a label matrix, L, containing labels for the connected
%components in BW. bwlabeln uses a default connectivity of 8 for two dimensions,
%26 for three dimensions
[L Ne]=bwlabel(Inputimage);
%L is a matrix containing the pixel value as label_number
%Ne is number of labels
```

**Step 6**: Measure the properties of the Image regions and Plot the bounding Box

```
%regionprops(BW,properties) returns measurements for the set of
%properties specified by properties for each 8-connected component
%(object) in the binary image
propied=regionprops(L,'BoundingBox');
imshow(~Inputimage);
hold on
%propied(1) gives 171.500    93.500    90.000   115.000
%x y width height
for n=1:size(propied,1)
 rectangle('Position',propied(n).BoundingBox,'EdgeColor','g','LineWidth',2)
end
hold off
pause (1);
```


## **Step 7**: Letter segmentation

```
Resultados='C:\Users\Abhi Mahajan\Desktop\ocr\images';
%% Objects extraction
figure
for n=1:Ne
 [r,c] = find(L==n);
 %returns the row_number and column_number of all the cells in label n
 n1=Inputimage(min(r):max(r),min(c):max(c));
%The Activity given at evaluation to resize the alternate 3-letters in the original image
%and then apply OCR on that image
 if mod(n,3)==0
 [rows columns]=size(n1);
 binaryImage = padarray(n1, 100);
 binaryImage = imresize(binaryImage, [rows, columns]);
 n1=binaryImage;
 Inputimage(min(r):max(r),min(c):max(c))=n1;
  end
 imshow(~n1);
 %Write the image scanned to a folder
 baseFileName = sprintf('%d.png',n);
 fullFileName = fullfile(Resultados, baseFileName); % No need to worry about slashes
now!
 imwrite(~n1, fullFileName);
 %Write the image in text file
 basefilename = sprintf('%d.txt',n);
 fullfilename = fullfile(Resultados, basefilename);
 imwrite(~n1,fullfilename);
 pause(0.5)
end
%Image with alternate 3-letters resized
imshow(~Inputimage);
```


**Every 3rd alternate letter resized**

**(Activity given in evaluation)**

## Result:

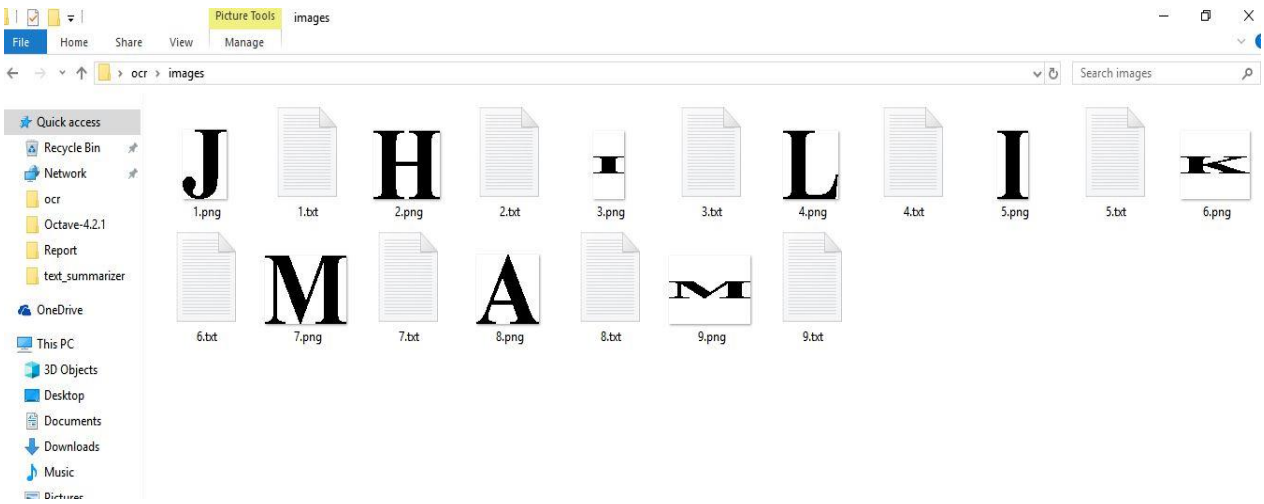


**Image of each letter(.png file) and its corresponding pixel values(as text file) saved in folder**