

Image processing

Vehicle number plate recognition
Using matlab

Submitted By:

HARISH AGGARWAL(16BCE0123)

ASHISH DESWAL(16BCE0907)

Under the guidance of:

Prof. AKILA VICTOR



VIT[®]
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)

VELLORE ■ CHENNAI

www.vit.ac.in

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
OCTOBER 2017

1. **Abstract**

The proposed method uses scan line evaluation and averaging method to localize the number plate followed by a border removal mechanism combined with character mending and approximation of character height to extract the number plate characters. Finally, a template matching approach is used to recognize the characters. A Graphical User Interface has been created and the algorithm is experimented successfully on a variety of real images, both single as well as double line plates. The sample results obtained on testing with various images are also detailed

2. Literature Review

a. Vehicle Plate Localization

- i. It is done to remove the unwanted background details, and thereby focusing on to the essential details in the image. Methods->
 - ii. Applying a top-hat filter to the whole image followed by a multiscale region search has been described .
 - iii. To detect the vertical edges, to extract the license plate using Sobel operators.
 - iv. A technique using edge detection and Hough transforms, to detect the vertical and horizontal edges, by making use of the rectangular shape of the license plate has been presented.
- Sorin developed an approach to analyse the input image, looking for areas with high contrast gradients at the given scale of about 15 pixels followed by histogram stretching

b. Character extraction

- i. It is done by segmenting the character portions from the localized number plate. Noise contents that may also be present along

with these number plate characters, makes the job very difficult. Methods->

- ii. To remove the frame lines in the number plate followed by digit filtering .
- iii. Color image segmentation, region growing and clustering.
- iv. A simple algorithm for labeling connected components on binary threshold image is detailed, using the elongation and direction to discard the components that do not correspond to digits .
- v. A system to solve the character extraction problem on a higher level, by combining the results of various thresholding techniques, also incorporating the feedback from character segmentation and character recognition .

c. **Character recognition**

- i. Character recognition is an essential and inevitable phase in such an application, where the system is put up to deal with nebulous or distorted characters in the license plate due to environmental hazards.

3. FUNCTIONS & IMPLEMENTATION

a. License plate localization

- i. As the image may have complex background details, number plate localization is the central issue that demands great attention. Scan line evaluation and averaging method is used here to accomplish this task. The image obtained from the sensor is filtered and binarized and the set of connected components are segmented.

b. License plate character extraction

- i. To extract the characters from the localized number plate, the image obtained from the previous step is complemented. The contents in this image will either be trivial noise components or characters to be identified. A border removal mechanism followed by the approximation of character height is performed to extract the characters

c. License plate character recognition

- i. Recognition is done through a template matching approach. The set of templates previously stored in the database are compared with this extracted character set. Correlation coefficient is used to define the relationship between the templates and the character in hand.

4. Code

```
1.   clc
2.   close all;
3.   clear;
4.   load imgfildata;
5.
6.   [file,path]=uigetfile({'*.jpg;*.bmp;*.png
   ;*.tif'}, 'Choose an image');
7.   s=[path,file];
8.   picture=imread(s);
9.   [~,cc]=size(picture);
10.  picture=imresize(picture,[300 500]);
11.
12.  if size(picture,3)==3
13.      picture=rgb2gray(picture);
14.  end
15.  % se=strel('rectangle',[5,5]);
16.  % a=imerode(picture,se);
17.  % figure,imshow(a);
18.  % b=imdilate(a,se);
19.  threshold = graythresh(picture);
20.  picture = ~im2bw(picture,threshold);
21.  picture = bwareaopen(picture,30);
22.  imshow(picture)
23.  if cc>2000
24.      picture1=bwareaopen(picture,3500);
25.  else
26.      picture1=bwareaopen(picture,3000);
27.  end
28.  figure,imshow(picture1)
29.  picture2=picture-picture1;
30.  figure,imshow(picture2)
31.  picture2=bwareaopen(picture2,200);
32.  figure,imshow(picture2)
33.
```

```

34. [L,Ne]=bwlabel (picture2);
35. propied=regionprops (L, 'BoundingBox');
36. hold on
37. pause(1)
38. for n=1:size(propied,1)
39.
    rectangle('Position',propied(n).BoundingBox,
    'EdgeColor','g','LineWidth',2)
40. end
41. hold off
42.
43. figure
44. final_output=[];
45. t=[];
46. for n=1:Ne
47.     [r,c] = find(L==n);
48.
    n1=picture(min(r):max(r),min(c):max(c));
49.     n1=imresize(n1,[42,24]);
50.     imshow(n1)
51.     pause(0.2)
52.     x=[ ];
53.
54.     totalLetters=size(imgfile,2);
55.
56.     for k=1:totalLetters
57.
58.         y=corr2(imgfile{1,k},n1);
59.         x=[x y];
60.
61.     end
62.     t=[t max(x)];
63.     if max(x)>.45
64.         z=find(x==max(x));
65.         out=cell2mat(imgfile(2,z));

```

```
66.  
67.     final_output=[final_output out];  
68.     end  
69.     end  
70.  
71.     file =  
        fopen('C:\Users\Dell\Desktop\Vehicle number  
        plate recognition\number_Plate.txt', 'wt');  
72.         fprintf(file, '%s\n', final_output);  
73.         fclose(file);  
74.  
        winopen('C:\Users\Dell\Desktop\Vehicle  
        number plate recognition\number_Plate.txt')
```

5. Screenshots

Original image



Detecting the number plate



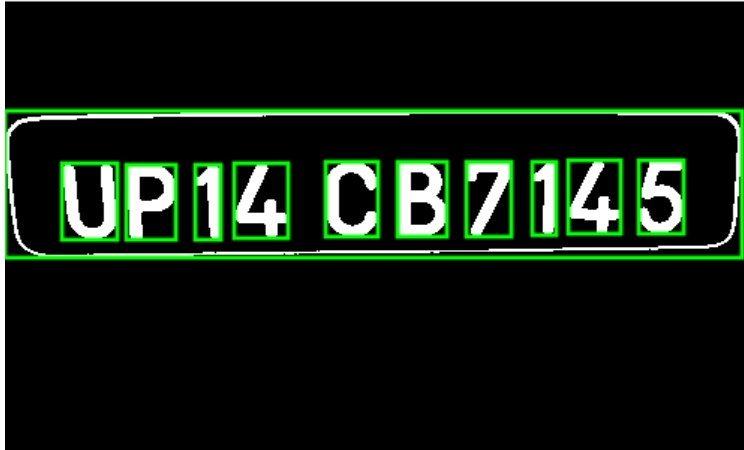
Removing unneeded parts



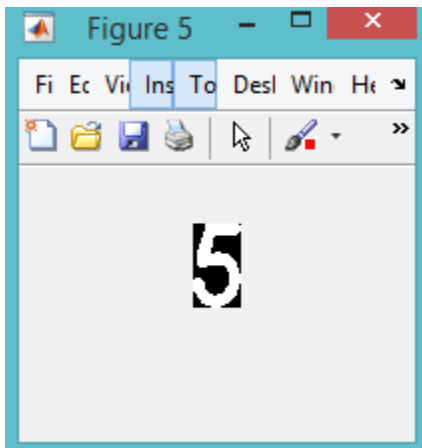
Result after removal



Detecting words



Displaying characters in order of detection



Number gets saved in a text document

