INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

NEW APPROACH TO RECOGNIZE NUMBER PLATES FOR INDIAN CONDITIONS

P.B.N.Chakravarthy[1], E. Jagadeeswararao[2]

[1]*Computer Networks Information Security & JNTU-SIT, India, chakripaluri@gmail.com*

[2]*Software Engineering &JNTU-SIT, India, jagadish513@gmail.com*

Abstract

OCR based number plate recognition system automatically recognizes the license number of vehicles. This system has algorithms like: 'Feature based number plate Localization' for locating the number plate, 'horizontal and vertical projection' for character segmentation Template method to normalize the characters and OCR for character recognition. The other main feature of this system is connecting to the database. This survey presents different types of techniques to recognize characters and different techniques to improve accuracy.

Keywords: *OCR, Image Processing, Feature Extraction, Number recognition.*

2. Introduction:

OCR based number plate recognition system for Indian Conditions is used to recognize the numbers on number plate along with characters like state and district identification characters. In this system we have to upload an image and pre-process it using algorithms which will be specified in this report.

The result is dependent on the uploaded image, if image consists of higher noise levels then it results to null output or "Not A valid Number Plate" output. So the uploaded image must has no noise. For Indian conditions the License plate does not have specific attributes or rules so that we have to use modified algorithms to detect a character on image along with OCR.

2. Process in proposed system:

2.1 Take a Picture:

As shown above the first step is to upload an image from memory. The uploaded image must be either noise free or having less noise. The images are taken at outside conditions which include variations in light, wind speed, pollution and motion of the object results to noise image. To reduce noise specific ISO 400 range is used, which is suitable for all conditions. Here in our situation we have to capture motion vehicle image without blur so shutter speed should be high so you should increase the ISO when there is not enough light for the camera to be able to

quickly capture an image i.e. Motion object.

Flow chart of overall project:

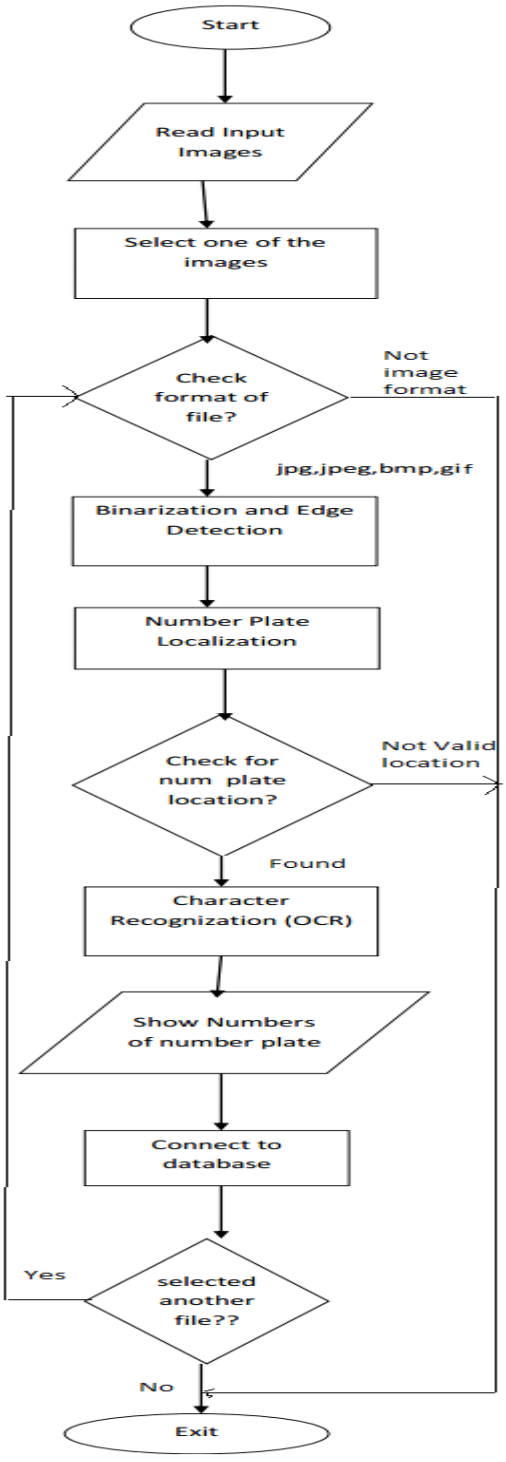

figure 1: flowchart of OCR based number plate recognition system

2.2. Read input Image:

We can read input image only after uploading that image. If you have uploaded so many image files then select one of them to recognize the license plate on the image.

Steps:

i. Click on “Image” button from menu option.

ii. Select “Load snapshots from directory”

iii. Select any folder which contain images (format: *.bmp;*.jpg; *.jpeg; *.png).

iv. Select files from the folder using FileListModel class.

2.3. Checking format of the file:

After uploading and selecting an image file pre-processing should be started over that image. Pre-processing include checking the format of the file, if the image file has JPG, JPEG, TIF, PNG, GIF, WebP, BMP as its file extension then it is a image file and further process is done. Otherwise it displays not an image file exception.

Steps:

i. Select one of the image shown in FileList panel.

ii .After selection the event fileListValueChanged() is triggered

iii. In that method ImageFileFilter class is used to check whether to accept that file or not using following code:

```
String type = new                String(name.substring(name.lastIndexOf('.')+1,name.length()).toLowerCase());

        if (!type.equals("bmp") &&

            !type.equals("jpg") &&

            !type.equals("jpeg") &&

            !type.equals("png") &&

            !type.equals("gif") &&

            !f.isDirectory()) return false;
```

iv. If the return value is false then the file is not an image file, otherwise it is an image file.

v. Show selected image in the panel panelCar.

2.4. Image Binarization:

If the file is an image file then that image is converted into gray scale image. To convert normal image to gray scale

image we use following steps:

Algorithm:

i. After clicking on "Recognize plate" button, it first invokes recognize() method of Intelligence class.

Ii. In the Intelligence class constructor, Brightness, saturation and Hue of pixel (x, y) in the image are calculated using following code:

```
int r = image.getRaster().getSample(x,y,0);

int g = image.getRaster().getSample(x,y,1);

int b = image.getRaster().getSample(x,y,2);

float[] hsb = Color.RGBtoHSB(r,g,b,null);
```

Iii. In the hsb[] array, hsb[0] is for Hue, hsb[1] is for saturation, hsb[2] is for brightness of that pixel.

Iv. If brightness is less than 0.5 then it is marked as zero (false), otherwise it is one (true).

V. False or zero indicates white color or brighter and true or one value indicates dark or black.

Vi. Hence the image with less brightened pixel is set to black and brightened pixel is set to white thus image is converted to black and white image.

2.5. Edge Detection:

Edge Detection technique is used to identify points where brightness changes sharply and continuously in the image. Edges are boundaries between two different regions.

In this project, to detect edges of an image,first we need to find out brightness of each pixel (x,y) using getBrightness() method of Photo.java class. This method converts given pixel RGB value to its HSB value and then stores each pixel Brightness to an array. If the brightness is less than or equal to the brightness of previous pixels then mark that array value to same as that of previous value otherwise mark it to 2. The pixels which are having array value one are edges, color that edges with white / black and remaining with black / white.

Algorithm to detect edges:

1. Select binary array of the image.

2. Select one point in the binary array.

3. For each point in binary array draw another 2D graphic image using the same color values by using gray color background,white black as normal colors.

4. The pixels with white color (bright color) with minimum value of (x,y) [(minX,minY)] is starting and maximum of(x,y) [(maxX,maxY)] is ending point.

5. Color the remaining part with black/gray color.

6. Normalize the image.

2.6. Number Plate Localization:

The very next step after edge detection is number plate localization. Localization is difficult if the images are of different colors, size. Various techniques were introduced recently for efficient detection of number plate regions from images. The technique we are going to use is selecting Best piece from the edge detection image i.e., cutting down remaining part from the image.

Algorithm:

1. Copy the image into another Buffered Image.

2. Draw a 2D graphic image with same pixel color values as of (minX, minY) and (maxX,maxY).

3. Exclude remaining part (ignore the remainig image).

4. The drawn image is our Number Plate image numImg.

2.7.Check for Number Plate Localization:

After locating the number plate detect whether that image really has numbers/ characters. We have to check it before going to the character recognition so that we can reduce the detection time if it is not a number plate. Generally this step is not necessary but to reduce processing time for invalid number plate i.e. Image with brighter edges but not having any characters at that edges for example the image may be as follows.

The above shown image is in correct format (image format) and it also has brightness varying points hence edge detection detects, number plate localization results to a part of the image. The part may be as follows:

Vol.2 Issue.9, Pg.: 69-75
September 2014

Here the image doesn't have any characters in it but it has brighter edges as that of car image with a number plate. So detecting that it is not a number plate before detecting the characters even it doesn't have any character in it leads to time saving.

The checking involves further detecting edges in the numImg. If there are characters present then the image surely has colored lines i.e., letters printed on the image must be of any colored straight or curved lines. So confirm that whether it has any lines on it or not.

2.8. Character Recognition:

It is the main part of the project, here to recognize character we are using OCR technique. In OCR numbers, characters on image are converted into editable text. Before going to use OCR method we have to do segmentation. Segmentation can be done using horizontal and vertical projection.

We can detect a character after segmentation only, horizontal and vertical projection on a image results to many rectangular parts with each part having one or more colored lines. For segmentation steps are as follows:

1. Use numImg as input to process segmentation.

2. Call renderVertically() method of Graph.java class for vertical projection.

3. In this method draw straight lines on the image and any special character to represent that it is end of that part of the image.

4. Copy that part of numImg to another rectangular graph.

5. Do the same for entire image.

6. In horizontal projection also we draw vertical lines using drawLine() method.

7. Use a special character to be placed at that projected area.

8. After horizontal and vertical projection, segment each part to BufferedImage array as a band of size 250X90.

9. The band image is compared with template which has different character images with the specified size and the name of the character is the character itself only. Hence we can find out character value easily.

2.9. Display Characters:

The characters that are recognized using OCR technique are displayed in the recognitionLabel.

2.10. Display Details:

After number plate detection, numbers displayed in the recognition Label are compared with the license plate in the database after database connection. If license number is matched with the database data, then details of the owner are displayed in new window.

3. Conclusion:

This system is useful in reducing crime rate if it is introduced at traffic and at high crime rate areas. Though plenty of algorithms were introduced, still user is facing problems with the recognition of characters. Finally I conclude that there is a necessity to improve OCR algorithm and this system should be introduced into the government of transportation or to RTO so that we can get correct details of the vehicle owner.

4. Acknowledgment:

This work is supported and made under the esteemed guidance of Mr. E Jagadeeshwar Rao, who is a lecturer and head of the department of Software Engineering at School of IT, Jawaharlal Nehru Technological University, and Hyderabad.

REFERENCES

[1] Automatic Number Plate Recognition (ANPR) System for Indian conditions,Prathamesh Kulkarni (Student Member, IEEE), Ashish Khatri, Prateek Banga, Kushal Shah,University of Pune, Dept. of Electronics and Telecommunication, India.2009
[2] Srivastava R: Transformations and distortions tolerant recognition of numerals using neural networks, ACM Annual Computer Science Conference, San Antonio, Texas, USA, 1991.
[3] Zhang Y., Zhang C.: New Algorithm for Character Segmentation of License Plate, Intelligent Vehicles Symposium, IEEE, 2003.
[4] Yo-Ping Huang, Shi-Yong Lai,Wei-Po Chuang, "A Template-Based Model for License Plate Recognition", IEEE International Conference on Networking, Sensing & Control, March 21-23, 2004.
[5] Indian Central Motor Vehicle Act, Bear Act, Rule no.49, 50.

BIOGRAPHY

P.B.N.Chakravarthy pursuing M.Tech in the stream Computer Networks and Information Security in JNTUH-School of Information Technology. Interested in the field of Image Processing, Security and Networking.

E.Jagadeeswararao, Assistant professor in Software Engineering JNTUH-SIT, has 5 years of teaching experience and interested in the field of Security, Software Engineering and Cloud Computing.