VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SANATHAN NARAYANAN

17BCE1094

Dr. Jagadeesh Kannan

B1+TB1

CSE4019 – Image Processing

Project Report

Security & Surveillance

Abstract – This project is a security and surveillance system for toll gates. Here there would be a camera which would capture photos of vehicles passing through the toll gate. Their licence plate nos. will be captures and be processed and stored in the database. This keeps the track of the vehicles passing through toll gates and the process of noting down the licence plate nos. and sharing of license plate nos. across different toll gates would be automated and made easy.

I. Introduction

In these days where the number of vehicles keep increasing, it becomes very important to automate the noting and sharing of license plate nos. of vehicles in toll gates. So, this project has been built keeping in mind the usefulness of automation in toll gates. This project was made successful in MATLAB. I have used operations like sobel, etc for edge detection. First the image is segmented or cropped to the required region. Then sobel operation is done to detect the edges. Before all this the image is first converted to grayscale and then made to a 2-tone image being black and white.

Motivation:

As the world's population is increasing rapidly and many toll gate workers are trying to get better jobs, we see that workers there are decreasing. So it is important to automate this.

Objective:

The goal of this project is to automate the toll gate process of noting down license plate nos.

Significance:

This project would have great effect in improving the toll gate security. Now as the major part is automated, toll gate workers can concentrate on other work.

II. Methodology

To go in detail, we can see that the image is read first by using the imread command in MATLAB. So, the image is read from its position.

Now the image is resized so that we don't waste processing power of the cpu. This also increases the efficiency of the project.

Now we convert the RGB image to grayscale so that the detection of characters would be closer to accurate.

But grayscale image is not enough. So, we convert to a 2-tone image which is nothing but binary image with only black and white pixels.

Now, we start detecting the edges of the image using the sobel operator. We use image dilation operation after this to get better results.

We are going to fill holes in the image to increase the accuracy of the detection.

Now, we erode the unwanted noises and details off the image.

We regionprop it and then we resize the number plate to 240 NaN. We will clear the dust which may come into picture while doing all these processes by using the opening operation in MATLAB.

We would also remove some object if it's width is too long or too small (than 500).

III. Related Works

AUTOMATIC NUMBER PLATE RECOGNITION SYSTEM FOR VEHICLE

IDENTIFICATION USING OPTICAL CHARACTER RECOGNITION - Muhammad Asif

In this paper we see that the main and most important part of the system in the software model would be to use series of the techniques which are used in image processing wherein they are implemented in MATLAB. [4] This will be done using ANPR algorithm which is broadly divided into so called 3 parts:

- Capture the Image
- Extract plate from image
- Recognize nos. from extracted plate

So, the first step would be do capture the image due to obvious reasons by an USB camera which is connected to PC. [3]

The images would be captured in the RGB format so that it can be processed further for the plate extraction where the plate is no. plate or nothing but the license plate of the vehicles.

We will now use an yellow algorithm wherein it is used for the extraction [1] of the ROI in an image in likelihood.

As we can see that the number plate officially of Sindh would have an yellow background wherein the alphanumeric [2] characters would be written in black only, so it would be easy to search the plate area by just scanning it for yellow colour.

The second step would be to extract the numbers in the number plate image. Here we do the yellow searching algorithm wherein we would extract the likelihood [5] ROI in the image which would be easy to detect as the plate area would be only consisting of searching the yellow pixel values.

So, the USB port would capture the image of the vehicle [5] wherein the ANPR algorithm which would be installed in the PC as a software package, would be used for the processing of the image.

The number is then compared with the authorized [3] number wherein we confirm its validity and also would finally provide the signal to the microcontroller which can be installed in PC, which would control the system's hardware.

If the plate which is inputted [2] would contain the authorized number then we would see the barrier on the entrance of the toll gate light to turn on and the 'Access Granted' will appear in the display and a green light will glow.

Otherwise there can be an 'Access Denied' or a 'Warning' [1] with a Red or Green light to be made to glow respectively.

Once the lines are successfully extracted, we can have the line separation process to be done.

Now this would be done as wherein the individual character can be separated. [4] The separated individual characters can be then stored in separate variables which reside in a database.

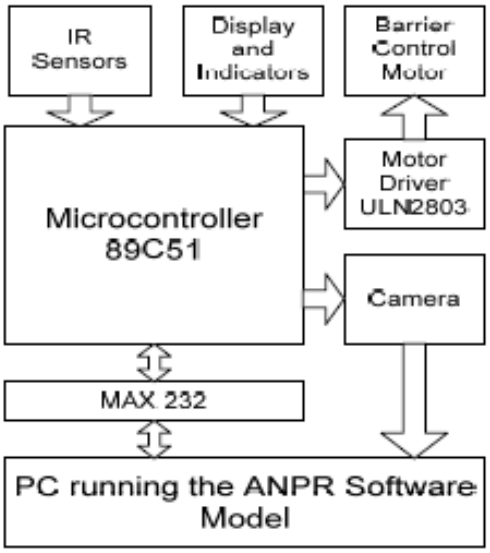


Figure 2 Hardware setup of ANPR system

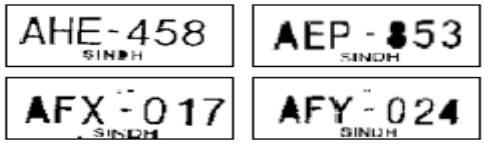


Figure 6 Binary image

Figure 7 Inverted binary image

Figure 8 Line separation using row segmentation

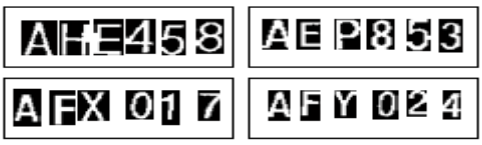


Figure 9 Character separation using column segmentation

Figure 10 Recognize character using OCR

IV. Existing Methods

Here we can see the existing methods wherein the APNR algorithm is used for recognizing the characters in the license plate in vehicles in the tolls gate.

The number is then compared with the authorized number wherein we confirm its validity and also would finally provide the signal to the

microcontroller which can be installed in PC, which would control the system's hardware.

If the plate which is inputted would contain the authorized number then we would see the barrier on the entrance of the toll gate light to turn on and the 'Access Granted' will appear in the display and a green light will glow.

Otherwise there can be an 'Access Denied' or a 'Warning' with a Red or Green light to be made to glow respectively.

Once the lines are successfully extracted, we can have the line separation process to be done.

Now this would be done as wherein the individual character can be separated. The separated individual characters can be then stored in separate variables which reside in a database.

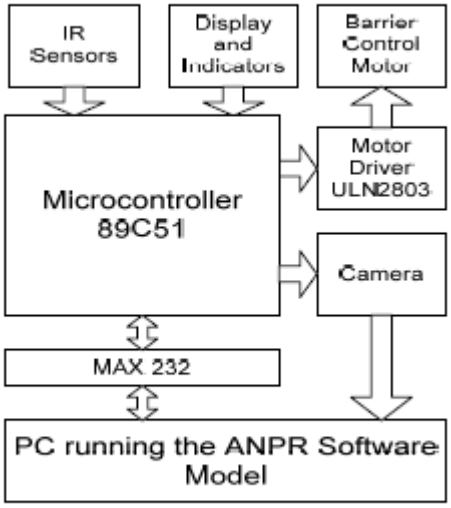


Figure 2 Hardware setup of ANPR system

V. Existing Demerits of Proposed Problem

Here we can see that the existing methods have demerits of slow response and bad accuracy.

VI. Proposed Method

Here the method which I proposed would include steps like Capturing image, and then to search the background which would contain yellow pixels. Now we would proceed to image filtering. After this we would use the smearing algorithm to extract the plate region. After all this, we would use the line separation method to segment the row. Now comes to the final steps for character recognition using column segmentation and then comes to the OCR for the character recognition.

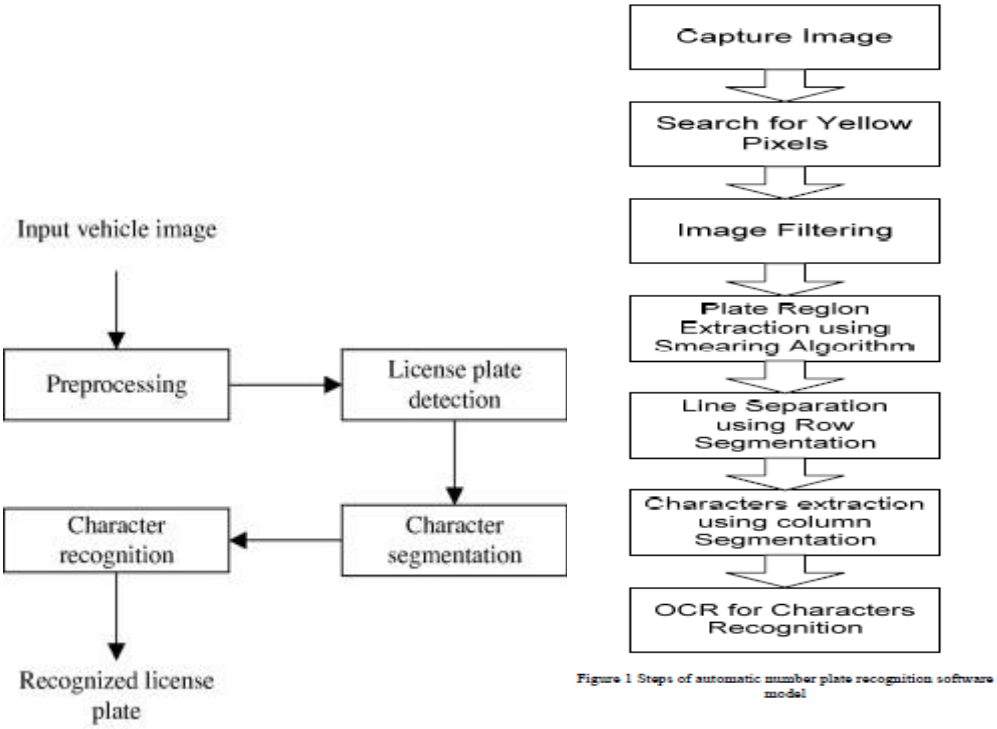


Figure 1 Steps of automatic number plate recognition software model

VII. Algorithms

1) Start

2) Capture the image using USB Camera and then send to the PC

3) Search for the yellow pixels in the background

4) Use image filtering techniques like Arithmetic mean filters,

Geometric, etc

5) Use smearing algorithm to segment or crop to the license plate part

6) Separate characters and lines using segmentation of the image

7) Extract characters and store it in the database for future requirements

8) Stop

VII.A. Code:

main.m:

```
close all;
clear all;

im = imread('img/car.jpg');
im = imresize(im, [480 NaN]);
imgray = rgb2gray(im);
imbin = imbinarize(imgray);
im = edge(imgray, 'sobel');


im = imdilate(im, strel('diamond', 2));
im = imfill(im, 'holes');
im = imerode(im, strel('diamond', 10));


Iprops=regionprops(im,'BoundingBox','Area',
'Image');
area = Iprops.Area;
count = numel(Iprops);
maxa= area;
boundingBox = Iprops.BoundingBox;
for i=1:count
   if maxa<Iprops(i).Area
```

```
        maxa=Iprops(i).Area;
        boundingBox=Iprops(i).BoundingBox;
    end
end

%all above step are to find location of number
plate

im = imcrop(imbin, boundingBox);

%resize number plate to 240 NaN
im = imresize(im, [240 NaN]);

%clear dust
im = imopen(im, strel('rectangle', [4 4]));

%remove some object if it width is too long or too
small than 500
im = bwareaopen(~im, 500);
%%%get width
 [h, w] = size(im);
% Iprops=regionprops(im,'BoundingBox','Area',
'Image');
% image = Iprops.Image;
% count = numel(Iprops);
% for i=1:count
%    ow = length(Iprops(i).Image(1,:));
%    if ow<(h/2)
%         im = im .* ~Iprops(i).Image;
%    end
% end


imshow(im);


%read letter
Iprops=regionprops(im,'BoundingBox','Area',
'Image');
count = numel(Iprops);
```

```
noPlate=[]; % Initializing the variable of number
plate string.

for i=1:count
   ow = length(Iprops(i).Image(1,:));
   oh = length(Iprops(i).Image(:,1));
   if ow<(h/2) & oh>(h/3)
       letter=readLetter(Iprops(i).Image); %
Reading the letter corresponding the binary image
'N'.
       figure; imshow(Iprops(i).Image);
       noPlate=[noPlate letter]; % Appending every
subsequent character in noPlate variable.
   end
end
```

create templates.m:

```
%CREATE TEMPLATES
%Letter
A=imread('char/A.bmp');B=imread('char/B.bmp');
C=imread('char/C.bmp');D=imread('char/D.bmp');
E=imread('char/E.bmp');F=imread('char/F.bmp');
G=imread('char/G.bmp');H=imread('char/H.bmp');
I=imread('char/I.bmp');J=imread('char/J.bmp');
K=imread('char/K.bmp');L=imread('char/L.bmp');
M=imread('char/M.bmp');N=imread('char/N.bmp');
O=imread('char/O.bmp');P=imread('char/P.bmp');
Q=imread('char/Q.bmp');R=imread('char/R.bmp');
S=imread('char/S.bmp');T=imread('char/T.bmp');
U=imread('char/U.bmp');V=imread('char/V.bmp');
W=imread('char/W.bmp');X=imread('char/X.bmp');
Y=imread('char/Y.bmp');Z=imread('char/Z.bmp');
Afill=imread('char/fillA.bmp');
Bfill=imread('char/fillB.bmp');
Dfill=imread('char/fillD.bmp');
Ofill=imread('char/fillO.bmp');
Pfill=imread('char/fillP.bmp');
Qfill=imread('char/fillQ.bmp');
```

```
Rfill=imread('char/fillR.bmp');



%Number
one=imread('char/1.bmp');
two=imread('char/2.bmp');
three=imread('char/3.bmp');four=imread('char/4.bmp
');
five=imread('char/5.bmp');
six=imread('char/6.bmp');
seven=imread('char/7.bmp');eight=imread('char/8.bm
p');
nine=imread('char/9.bmp');
zero=imread('char/0.bmp');
zerofill=imread('char/fill0.bmp');
fourfill=imread('char/fill4.bmp');
sixfill=imread('char/fill6.bmp');
sixfill2=imread('char/fill6_2.bmp');
eightfill=imread('char/fill8.bmp');
ninefill=imread('char/fill9.bmp');
ninefill2=imread('char/fill9_2.bmp');



%*-*-*-*-*-*-*-*-*-*-
letter=[A Afill B Bfill C D Dfill E F G H I J K L
M N O Ofill P Pfill Q Qfill R Rfill S T U V W X Y
Z];

number=[one two three four fourfill five six
sixfill sixfill2 seven eight eightfill nine
ninefill ninefill2 zero zerofill];

character=[letter number];

NewTemplates=mat2cell(character,42,[24 24 24 24 24
24 24 24 ...
    24 24 24 24 24 24 24 ...
```

```
    24 24 24 24 24 24 24 ...
    24 24 24 24 24 24 24 ...
    24 24 24 24 24 24 24 ...
    24 24 24 24 24 24 24 ...
    24 24 24 24 24 24 24]);




save ('NewTemplates','NewTemplates')
clear all
```

readLetter.m

```
function letter=readLetter(snap)
%READLETTER reads the character fromthe
character's binary image.
%   LETTER=READLETTER(SNAP) outputs the character
in class 'char' from the
%   input binary image SNAP.

load NewTemplates % Loads the templates of
characters in the memory.
snap=imresize(snap,[42 24]); % Resize the input
image so it can be compared with the template's
images.
comp=[ ];
for n=1:length(NewTemplates)
    sem=corr2(NewTemplates{1,n},snap); %
Correlation the input image with every image in
the template for best matching.
    comp=[comp sem]; % Record the value of
correlation for each template's character.
    %display(sem);

end
vd=find(comp==max(comp)); % Find the index which
correspond to the highest matched character.
%display(max(comp));
```

```
%*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
% Accodrding to the index assign to 'letter'.
% Alphabets listings.
if vd==1 || vd==2
    letter='A';
elseif vd==3 || vd==4
    letter='B';
elseif vd==5
    letter='C';
elseif vd==6 || vd==7
    letter='D';
elseif vd==8
    letter='E';
elseif vd==9
    letter='F';
elseif vd==10
    letter='G';
elseif vd==11
    letter='H';
elseif vd==12
    letter='I';
elseif vd==13
    letter='J';
elseif vd==14
    letter='K';
elseif vd==15
    letter='L';
elseif vd==16
    letter='M';
elseif vd==17
    letter='N';
elseif vd==18 || vd==19
    letter='O';
elseif vd==20 || vd==21
    letter='P';
elseif vd==22 || vd==23
    letter='Q';
elseif vd==24 || vd==25
    letter='R';
elseif vd==26
    letter='S';
```

```
elseif vd==27
    letter='T';
elseif vd==28
    letter='U';
elseif vd==29
    letter='V';
elseif vd==30
    letter='W';
elseif vd==31
    letter='X';
elseif vd==32
    letter='Y';
elseif vd==33
    letter='Z';
    %*-*-*-*-*
% Numerals listings.
elseif vd==34
    letter='1';
elseif vd==35
    letter='2';
elseif vd==36
    letter='3';
elseif vd==37 || vd==38
    letter='4';
elseif vd==39
    letter='5';
elseif vd==40 || vd==41 || vd==42
    letter='6';
elseif vd==43
    letter='7';
elseif vd==44 || vd==45
    letter='8';
elseif vd==46 || vd==47 || vd==48
    letter='9';
else
    letter='0';
end
end
```

VII.b. Datasets: (Got from Kaggle)

Standard Benchmark Training Dataset:

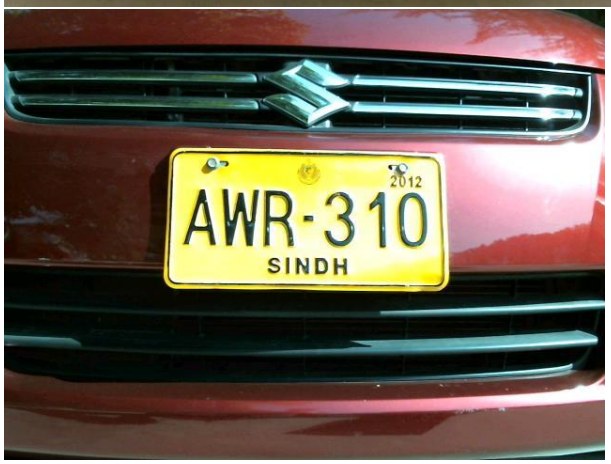
Testing Dataset:


IND KA 19 P 8488


KA 03 NA 303


GT
MH 14 DT 8831

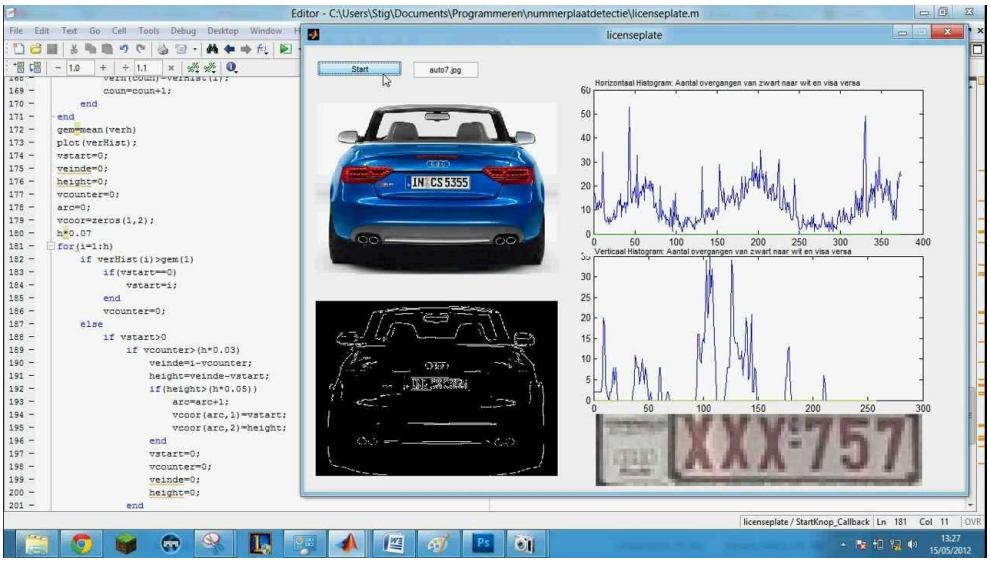

aspire
2012
AXZ-016
SINDH

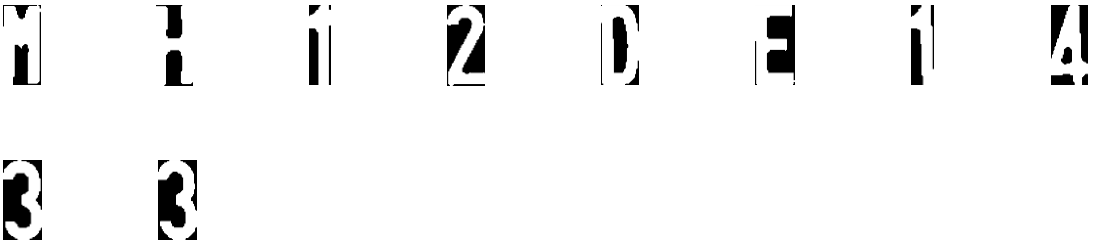


VIII. Results

Hence, we can see that this project can recognize the characters in an efficient and a quicker way than few of the existing methods.

This project being done in MATLAB using many operations like Sobel, etc which promote faster processing.





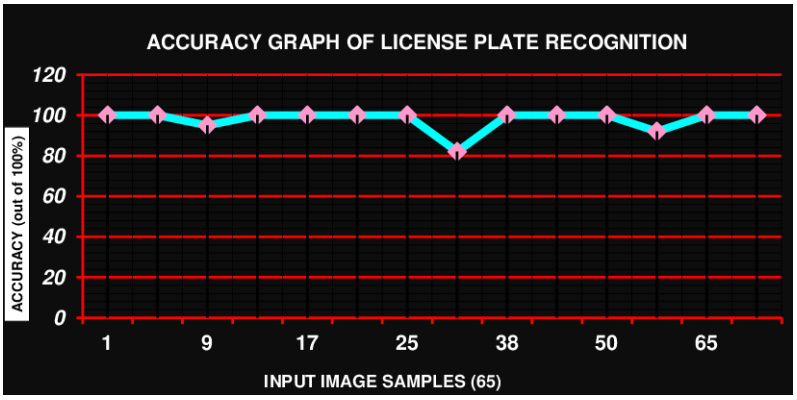


Table:

Methods	Properties
1st best: APNR	Uses high end APNR techniques
2nd best: My Method	Uses high end filtering, segmentation, noise removal, edge detection & OCR techniques with antiblur features and recognize cursive writing
3rd best: Basic OpenCV	Uses basic recognition techniques with lesser speed and accuracy

IX. Conclusion

Thus, we are able to see that this project has the capability to increase GDP of India to a good extent when this project is implemented in a mass scale.

This is a powerful project implemented with a powerful and professional tool MATLAB where it is easy to add features and we can see the performance of this project to be very good and is at par with the current techniques.

Future Scope:

We can also add features like Neural Networks, Machine Learning, AI, Big Data wherein the machine should learn about the different colours, styles, formatting, and also noise in the captured images. It should be able to deblur blurred images in case of fog, moving vehicles, etc.

X. References

[1] Optasia Systems Pte Ltd, "The World Leader in License Plate Recognition Technology" Sourced from: www.singaporegateway.com/optasia, Accessed 22 November 2008.

[2] V. Kasmat, and S. Ganesan, "An efficient implementation of the Hough transform for detecting vehicle license plates using DSP's,"

IEEE International Conference on Real-Time Technology and Application Symposium, Chicago, USA, pp. 58-59, 2005.

[3] J. W. Hsieh, S. H. Yu, and Y. S. Chen. Morphology based license plate detection from complex scenes. 16th International Conference on Pattern Recognition (ICPR'02), pp. 79–179, 2002.

[4]
K.K. KIM, K.I., KIM, J.B. KIM, and H.J. KIM, "Learning-Based Apporach for License Plate Recognition" Proceeding of IEEE Signal Processing Society Workshop, Vol. 2, pp.614-623, 2000.

[5] S.H. Park, K.I. kim, K. Jung and H.J. Kim, "Locating car license plate using Neural Network," Electronic Letters, Vol. 35, No. 17, pp. 1474 – 1477, 1999.