

*A Synopsis***“Vehicle Number Plate Detection”**

Submitted in fulfillment of the requirement for the award of the Degree

of

**Bachelor of Technology**

in

**Electronics & Communication Engineering**

**Submitted by**

|                  |             |
|------------------|-------------|
| Prakriti Maiti   | 23600319021 |
| Sourabh Shao     | 23600319008 |
| Abishek Palit    | 23600319036 |
| Shibjyoti Singha | 23600319007 |
| Ramsundor Halder | 23600319034 |

**Under the Supervision of**

**“Prof. Gour Gopal Jana”**



**DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING**

**Greater Kolkata College of Engineering and Management**

**Baruipur, near Phultala, South 24 Parganas – 743387, West Bengal**

**November, 2021**

# Contents

| <b>Details</b>                       | <b>Page Number</b>  |
|--------------------------------------|---|
| • <b>Title</b>                       | <b>3</b>  |
| • <b>Introduction</b>                | <b>4</b>  |
| • <b>HARDWARE REQUIREMENTS</b>       | <b>5</b>  |
| • <b>SOFTWARE REQUIREMENTS</b>       | <b>6</b>  |
| • <b>Process Figure</b>              | <b>7</b>  |
| • <b>Process</b>                     | <b>8</b>  |
| • <b>Code</b>                        | <b>9</b>  |
| • <b>Applications of the project</b> | <b>12</b>   |
| • <b>References</b>                  | <b>13</b>   |
| • <b>For All Programming</b>         | <a href="https://github.com/PrakritiMaiti/Vehicle-Number-Plate-Detection-Using-Matlab.git">https://github.com/PrakritiMaiti/Vehicle-Number-Plate-Detection-Using-Matlab.git</a> |

## **Title**

# **Vehicle Number Plate Detection Using Matlab**

## **Introduction**

Our project involves the extraction of text from the number plate of a car with the help of MATLAB. We want to build an automated system that reads the number of a car from the image of the number plate. The objective is to design an efficient automatic authorized vehicle identification system by using the vehicle number plate. The system is implemented on the entrance for security control of a highly restricted area like military zones or areas around top government offices e.g. Parliament, Supreme Court etc. The developed system first detects the vehicle and then captures the vehicle image. Vehicle number plate region is extracted using the image segmentation in an image. Optical character recognition technique is used for the character recognition. The resulting data is then used to compare with the records on a database so as to come up with the specific information like the vehicle's owner, place of registration, address, etc. The system is implemented and simulated in MATLAB, and its performance is tested on real images. It is observed from the experiment that the developed system successfully detects and recognizes the vehicle number plate on real images.

## **HARDWARE REQUIREMENTS**

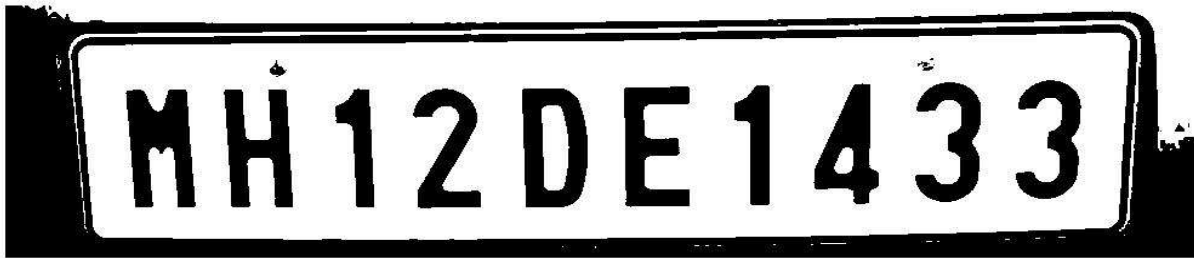
Here, the detection is completely based on MATLAB code; therefore no hardware implementation is used. The actual security systems which connect this project with the database use Hardware. We are directly providing images to the system, while the actual implementation requires certain cameras and servers.

A camera that takes the images of the car (front or rear side). A controlled light that can bright up the plate, and allow day and night operation. In most cases the illumination is Infra-Red (IR) which is invisible to the driver. An interface board between the camera and the PC, allows the software to read the image information. A PC running Windows or Linux. It runs the LPR application which controls the system, reads the images, analyses and identifies the plate, and interfaces with other applications and systems.

## **SOFTWARE REQUIREMENTS**

The MATLAB application and the recognition package. Usually the recognition package is supplied as a DLL (Dynamic Link Library). A database software in which the events are recorded on a local database or transmitted over the network. The data includes the recognition results and (optionally) the vehicle.

## Process Figure



## Process

License Plate Recognition is an image processing technology which uses number (license) plate to identify the vehicle. The objective is to design an efficient automatic authorized vehicle identification system by using the vehicle number plate. The developed system first detects the vehicle and then captures the vehicle image. Vehicle number plate region is extracted using the image segmentation in an image. Optical character recognition technique is used for the character recognition. The resulting data is then used to compare with the records on a database so as to come up with the specific information like the vehicle owner, place of registration, and address.

The software aspect of the system runs on standard home computer hardware and can be linked to other applications or databases. It first uses a series of image manipulation techniques to detect, normalize and recognise the image of the number plate, and then optical character recognition (OCR) to extract the alpha-numeric of the license plate. This Automatic Number Plate Recognition system can be generally deployed in one of two basic approaches: one allows for the entire process to be performed at the lane location in real-time, and the other transmits all the images from many lanes to a remote computer location and performs the OCR process there at some later point in time. Here we have designed the system such that it gives the output in textual form after segmentation and recognition.

Since the vehicle plates are based on different country standards, they are usually different in form, shape and material. Therefore, the LPR systems are country specific and are adapted to the country where they are installed and used. Our system will recognize the image taken by the IR cameras and just give the output in the textual form regardless of the country specifics.

Some plates have additional information written or attached to the plate, such as state identification, province initials, plate issue date label, vehicle type, country flag and more. For most applications this additional information is not required (and is usually not identified due to the complexity, variation, quality and resolution). Thus, we only intend to give the registered number as the output.



## Code

### 1. Main Code:

```

clc;
clear;
[file,path]=uigetfile({'*.jpg;*.png'},'Choose an image');
s=[path,file];
im1 = imread(s);
imgray = rgb2gray(im1);
imbin = imbinarize(imgray);
im = edge(imgray, 'prewitt');

%Below steps are to find location of number plate
Iprops1=regionprops(im,'BoundingBox','Area', 'Image');
area1 = Iprops1.Area;
count1 = numel(Iprops1);
maxa= area1;
boundingBox1 = Iprops1.BoundingBox;
for i=1:count1
    if maxa<Iprops1(i).Area
        maxa=Iprops1(i).Area;
        boundingBox1=Iprops1(i).BoundingBox;
    end
end

im = imcrop(imbin, boundingBox1);%crop the number plate area
im = bwareaopen(~im, 500); %remove some object if it width is too long or too small than 500

[h, w] = size(im);%get width

imshow(im);

Iprops=regionprops(im,'BoundingBox','Area', 'Image'); %read letter
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=Letter_detection(Iprops(i).Image); % Reading the letter corresponding the binary image 'N'.
        noPlate=[noPlate letter]; % Appending every subsequent character in noPlate variable.
    end
end
%output
disp(noPlate)
file=fopen('number_plate.txt','wt');
fprintf(file,'%s\n',noPlate);
fclose(file);
winopen('number_plate.txt')

```

### 2. Letter Detection:

```
function letter=Letter_detection(snap)
```

```
load NewTemplates.mat NewTemplates
```

```
snap=imresize(snap,[42 24]);
rec=[ ];

for n=1:length(NewTemplates)
    cor=corr2(NewTemplates{1,n},snap);
    rec=[rec cor];
end

ind=find(rec==max(rec));
display(ind);

% Alphabets listings.
if ind==1 || ind==2
    letter='A';
elseif ind==3 || ind==4
    letter='B';
elseif ind==5
    letter='C';
elseif ind==6 || ind==7
    letter='D';
elseif ind==8
    letter='E';
elseif ind==9
    letter='F';
elseif ind==10
    letter='G';
elseif ind==11
    letter='H';
elseif ind==12
    letter='I';
elseif ind==13
    letter='J';
elseif ind==14
    letter='K';
elseif ind==15
    letter='L';
elseif ind==16
    letter='M';
elseif ind==17
    letter='N';
elseif ind==18 || ind==19
    letter='O';
elseif ind==20 || ind==21
    letter='P';
elseif ind==22 || ind==23
    letter='Q';
elseif ind==24 || ind==25
    letter='R';
elseif ind==26
    letter='S';
elseif ind==27
    letter='T';
elseif ind==28
    letter='U';
elseif ind==29
    letter='V';
```

```

elseif ind==30
    letter='W';
elseif ind==31
    letter='X';
elseif ind==32
    letter='Y';
elseif ind==33
    letter='Z';
    % Numerals listings.
elseif ind==34
    letter='1';
elseif ind==35
    letter='2';
elseif ind==36
    letter='3';
elseif ind==37 || ind==38
    letter='4';
elseif ind==39
    letter='5';
elseif ind==40 || ind==41 || ind==42
    letter='6';
elseif ind==43
    letter='7';
elseif ind==44 || ind==45
    letter='8';
elseif ind==46 || ind==47 || ind==48
    letter='9';
else
    letter='0';
end
end

```

### 3. Template Creation

```

%Alphabets
A=imread('Alpha/A.bmp');B=imread('Alpha/B.bmp');C=imread('Alpha/C.bmp');D=imread('Alpha/D.bmp');
E=imread('Alpha/E.bmp');F=imread('Alpha/F.bmp');G=imread('Alpha/G.bmp');H=imread('Alpha/H.bmp');
I=imread('Alpha/I.bmp');J=imread('Alpha/J.bmp');K=imread('Alpha/K.bmp');L=imread('Alpha/L.bmp');
M=imread('Alpha/M.bmp');N=imread('Alpha/N.bmp');O=imread('Alpha/O.bmp');P=imread('Alpha/P.bmp');
Q=imread('Alpha/Q.bmp');R=imread('Alpha/R.bmp');S=imread('Alpha/S.bmp');T=imread('Alpha/T.bmp');
U=imread('Alpha/U.bmp');V=imread('Alpha/V.bmp');W=imread('Alpha/W.bmp');X=imread('Alpha/X.bmp');
Y=imread('Alpha/Y.bmp');Z=imread('Alpha/Z.bmp');

%Natural Numbers
one=imread('Alpha/1.bmp');two=imread('Alpha/2.bmp');three=imread('Alpha/3.bmp');
four=imread('Alpha/4.bmp');five=imread('Alpha/5.bmp');six=imread('Alpha/6.bmp');
seven=imread('Alpha/7.bmp');eight=imread('Alpha/8.bmp');nine=imread('Alpha/9.bmp');
zero=imread('Alpha/0.bmp');

%Creating Array for Alphabets
letter=[A B C D E F G H I J K L M N O P Q R S T U V W X Y Z];
%Creating Array for Numbers
number=[one two three four five six seven eight nine zero];

NewTemplates=[letter number];
save ('NewTemplates','NewTemplates')

```

## **Applications of the project**

Parking

Tolling

Marketing Tool

Border Control

Identifying Stolen Cars Easily

Help In Identification of Cars With Unpaid Tickets

## References

1. <https://github.com/PrakritiMaiti/Vehicle-Number-Plate-Detection-Using-Matlab.git>
2. <https://circuitdigest.com/tutorial/vehicle-number-plate-detection-using-matlab-and-image-processing>
3. <https://www.electronicsforu.com/electronics-projects/hardware-diy/vehicle-number-plate-detection-using-matlab>
4. “**Matlab**”, [Online] Available: “[Matlab Help](#)”
5. “Google”, [Online] Available: “[Google](#)”