**DIP PROJECT**

**GROUP NO -14**

# TITLE OF MINI PROJECT: [Number Plate Detection using](https://circuitdigest.com/tutorial/vehicle-number-plate-detection-using-matlab-and-image-processing) image processing

|  |  |
| --- | --- |
| **NAME** | **SRN** |
| EESHA SOUGANDHEEKA KOTHURU | PES1UG19EC141 |
| B. PRAGNANA REDDY | PES1UG19EC077 |
| NAVYA SREE PALADI | PES1UG19EC192 |

**Introduction to topic:**

ANPR (Automatic Number Plate Recognition) is a method to detect the number plate of a vehicle, the camera of the ANPR system captures image of vehicle license plate and then the image is processed through multiple number of algorithms to provide an alpha numeric conversion of the image into a text format. ANPR system is used at many places like Petrol Pumps, shopping malls, Airports, highways, toll booths, Hotels, Hospitals, Parking lots, Defense & Military check points etc. There are many image processing tools available for this Number plate detection.

**Theory and code:**

* We used 3 .m files for the project given below is the corresponding explanation of all 3 files
* **Template Creation (template\_creation.m)–** This is used to call the saved images of alphanumerics and then save them as a new template in MATLAB memory.
* **Letter Detection (Letter\_detection.m)** – Reads the characters from the input image and find the highest matched corresponding alphanumeric.
* **Plate Detection (Plate\_detection.m)** – Process the image and then call the above two m-files to detect the number.
* We also created folder called alpha which have images of alphabets , so we assigned the corresponding alphabet with the particular alphabet image and similarly numbers.

# Code of template\_creation.m

%CREATE TEMPLATES

%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')

clear all

* We created a matrix of ‘letter’ and ‘number’ and saved it in NewTemplates.

# Code of Letter\_Detection.m

function letter=readLetter(snap)

load 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

# In the above code we have created a function named letter which gives us the alphanumeric output of the input image from class *‘*alpha*’* by using command ‘readLetter()’. And then load the saved templates by using command load ‘NewTemplates*.*

# After that, we have resized the input image so it can be compared with the template’s images by using the command ‘imresize(filename,size)’. Then for loop is used to correlates the input image with every image in the template to get the best match.

# A matrix ‘rec’ is created to record the value of correlation for each alphanumeric template with the characters template from the input image, as shown in the below code,

# cor=corr2(NewTemplates{1,n},snap);

# Then ‘find()’ command is used to find the index which corresponds to the highest matched character. Then according to that index, corresponding character is printed using‘if-else’ statement

# Code of Plate\_detection.m

close all;

clear all;

im = imread('C:\Users\Admin\Desktop\sem5\cs\Number Plate Detection\Number Plate Images\image1.png');

imgray = rgb2gray(im);

imbin = imbinarize(imgray);

im = edge(imgray, 'prewitt');

%Below steps are to find location of number plate

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

im = imcrop(imbin, boundingBox);%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

# imread() – This command is used to open the image into the MATLAB from the target folder.

# rgb2gray() –This command is used to convert the RGB image into grayscale format.

# imbinarize() – This command is used to Binarize 2-D grayscale image or simply we can say it converts the image into black and white format.

# edge() – This command is used to detect the edges in the image, by using various methods like Roberts, Sobel, Prewitt and many others.

# regionprops() – This command is used to measure properties of image region.

# numel() – This command is used to calculate the number of array elements.

# imcrop() – This command is used to crop the image in the entered size.

# bwareaopen() – This command is used to remove small objects from binary image.

# By using the above commands in the code, we are calling the input image and converting it into the grayscale. Then the grayscale is converted into the binary image, and the edge of the binary images is detected by the Prewitt method.

# Detecting the location of the number plate in the entire input image after that

# crop the number plate and remove the small objects from the binary image by using command ‘imcrop()’ and ‘bwareaopen()’ respectively.

# Process that cropped license plate image and display the detected number in the image and text format.

# Output:

# 

# 

# 

# 