MATLAB (HTTPS://CIRCUITDIGEST.COM/MATLAB-PROJECTS)

Car Number Plate Detection Using MATLAB and Image Processing (/tutorial/vehicle-number-plate-detection-using-matlab-and-image-processing)

By   (page_author.html)Pankaj Khatri (/users/pankaj-khatri)   ⏱ Nov 26, 2018
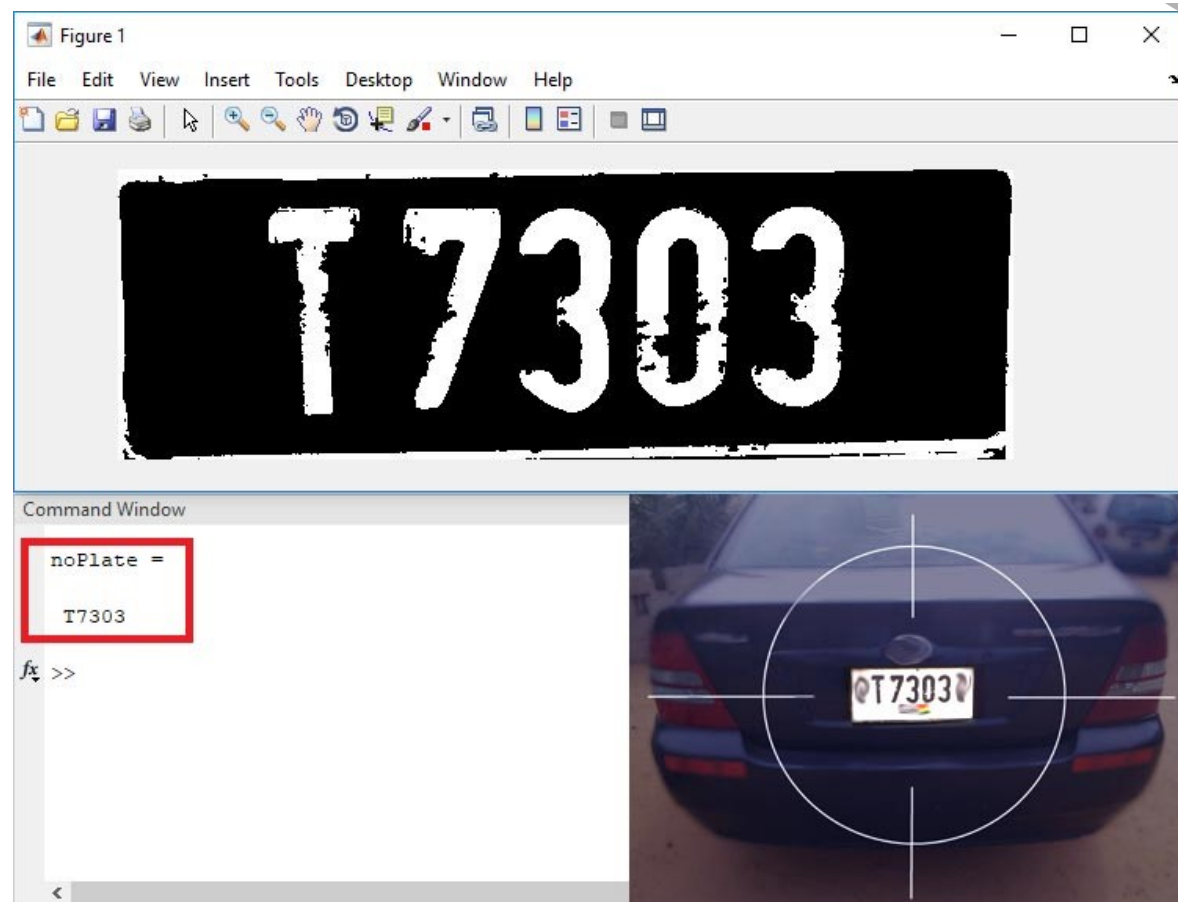
Car Number Plate Detection Using MATLAB and Image Processing

Have you ever wonder that how an ANPR (Automatic Number Plate Recognition) system works? Let me tell you the concept behind it, 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, but here in this tutorial we will use **MATLAB Image Processing to get the vehicle license plate number** into the text format. If you are new with MATLAB or image processing, then check our previous MATLAB projects:

- Getting started with MATLAB: A Quick Introduction (https://circuitdigest.com/tutorial/what-is-matlab-and-how-to-get-started-with-it)
- Getting Started with Image Processing using MATLAB (https://circuitdigest.com/tutorial/getting-started-with-image-processing-using-matlab)

First, let me brief you about the concept we are using for detecting number plates. There are three programs or '.m' files for this project.

- **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.

Now, we will learn about how to code these m-files and what you have to do before start coding. After going through this tutorial, you can find **all the code files and working explanation video at**

the end of this project.

Template Creation

First create a folder for the project (my folder name is *Number Plate Detection*) to save and store the files. We have stored the binary images of all the alphabets and numbers in the sub-folder named as '*alpha*'.

Now, open the Editor window in the MATLAB, as shown in the below image,
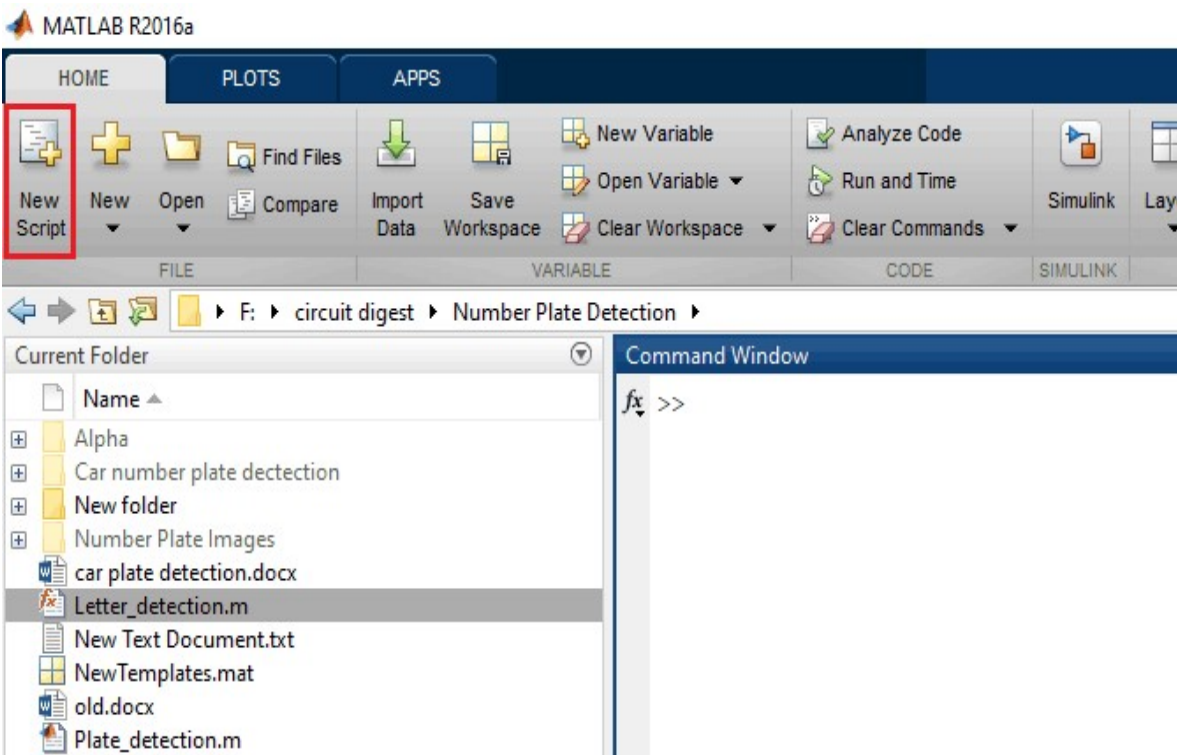
If you are not familiar with the basic terminology of MATLAB (https://circuitdigest.com/tutorial /what-is-matlab-and-how-to-get-started-with-it) I suggest you to check the linked tutorial.

Now, copy and paste the below code in **template_creation.m** file, and save the file in the project folder (*Number Plate Detection*). All the files related to this project including image templates files can be downloaded from here (/sites/default/files/Matlab-Number-Plate-Detection.zip). Also **check the video** given at the end of this project.

```
%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
```

Here, in the above code we are saving the images into a variable by using command **'imread()'**. This function is used to call the images from the folder or from any location of the PC into the MATLAB. Let's take an example from the above code:

```
A=imread('alpha/A.bmp');
```

Where A is the variable, and in '*alpha/A.bmp*', '*alpha*' is the folder name and '*A.bmp*' is the file name.

Then create a matrix of '*letter*' and '*number*' and save it in variable '*NewTemplates*' by using command '*save(filename,variables)*'.

```
%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
```

Now start coding *Letter_detection.m*, in a new editor window.

Letter Detection

Here we are creating the second code file named **Letter_detection.m.** Now, copy and paste the below code in that file and save the file in the project folder with name *Letter_detection.* This file can be downloaded from here (/sites/default/files/Matlab-Number-Plate-Detection.zip), this attached zip files also contains other files related to this Number plate detection project.

```matlab
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(find(rec==max(rec)));

% 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
```

```matlab
Editor - F:\circuit digest\Number Plate Detection\Letter_detection.m

  Plate_detection.m  ×  Letter_detection.m  ×  template_creation.m  ×  +

 1      function letter=readLetter(snap)
 2
 3 -    load NewTemplates
 4 -    snap=imresize(snap,[42 24]);
 5 -    rec=[ ];
 6
 7 -    for n=1:length(NewTemplates)
 8 -        cor=corr2(NewTemplates{1,n},snap);
 9 -        rec=[rec cor];
10 -    end
11
12 -    ind=find(rec==max(rec));
13 -    display(ind);
14
15      % Alphabets listings.
16 -    if ind==1 || ind==2
17 -        letter='A';
18 -    elseif ind==3 || ind==4
19 -        letter='B';
20 -    elseif ind==5
21 -        letter='C';
22 -    elseif ind==6 || ind==7
23 -        letter='D';
24 -    elseif ind==8
25 -        letter='E';
26 -    elseif ind==9
27 -        letter='F';
28 -    elseif ind==10
29 -        letter='G';
30 -    elseif ind==11
31 -        letter='H';
32 -    elseif ind==12
33 -        letter='I';
34 -    elseif ind==13
35 -        letter='J';
36 -    elseif ind==14
37 -        letter='K';
38 -    elseif ind==15
39 -        letter='L';
40 -    elseif ind==16
41 -        letter='M';
42 -    elseif ind==17
43 -        letter='N';
44 -    elseif ind==18 || ind==19
45 -        letter='O';
46 -    elseif ind==20 || ind==21
47 -        letter='P';
48 -    elseif ind==22 || ind==23
49 -        letter='Q';
50 -    elseif ind==24 || ind==25
51 -        letter='R';
52 -    elseif ind==26
53 -        letter='S';
54 -    elseif ind==27
55 -        letter='T';
56 -    elseif ind==28
57 -        letter='U';
58 -    elseif ind==29
59 -        letter='V';
60 -    elseif ind==30
61 -        letter='W';
62 -    elseif ind==31
63 -        letter='X';
64 -    elseif ind==32
65 -        letter='Y';
66 -    elseif ind==33
67 -        letter='Z';
68          %*-*-*-*-*
69      % Numerals listings.
70 -    elseif ind==34
71 -        letter='1';
72 -    elseif ind==35
73 -        letter='2';
74 -    elseif ind==36
75 -        letter='3';
76 -    elseif ind==37 || ind==38
77 -        letter='4';
78 -    elseif ind==39
79 -        letter='5';
80 -    elseif ind==40 || ind==41 || ind==42
81 -        letter='6';
82 -    elseif ind==43
83 -        letter='7';
84 -    elseif ind==44 || ind==45
85 -        letter='8';
86 -    elseif ind==46 || ind==47 || ind==48
87 -        letter='9';
88 -    else
89 -        letter='0';
90 -    end
91 -    end
```

Here, 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.

Now, after completing with this open a new editor window to start code for the main program.

Number Plate Detection

Here is the third and final code file named **Plate_detection.m** copy and paste the below code in this file and save into the project folder. For quick start you can download all the code files with image templates from here (/sites/default/files/Matlab-Number-Plate-Detection.zip).

```
close all;
clear all;

im = imread(' 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);
im = bwareaopen(~im, 500);
 [h, w] = size(im);

imshow(im);

Iprops=regionprops(im,'BoundingBox','Area', 'Image');
count = numel(Iprops);
noPlate=[];
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);
        noPlate=[noPlate letter]
    end
end
```

```
Editor - F:\circuit digest\Number Plate Detection\Plate_detection.m

Plate_detection.m  ×  Letter_detection.m  ×  template_creation.m  ×  +

1 -     close all;
2 -     clear all;
3
4 -     im = imread('Number Plate Images/image1.png');
5 -     imgray = rgb2gray(im);
6 -     imbin = imbinarize(imgray);
7 -     im = edge(imgray, 'prewitt');
8
9       %Below steps are to find location of number plate
10 -    Iprops=regionprops(im,'BoundingBox','Area', 'Image');
11 -    area = Iprops.Area;
12 -    count = numel(Iprops);
13 -    maxa= area;
14 -    boundingBox = Iprops.BoundingBox;
15 - ⊟for i=1:count
16 -       if maxa<Iprops(i).Area
17 -           maxa=Iprops(i).Area;
18 -           boundingBox=Iprops(i).BoundingBox;
19 -       end
20 -   └end
21
22 -    im = imcrop(imbin, boundingBox);%crop the number plate area
23 -    im = bwareaopen(~im, 500); %remove some object if it width is too long or too small than 500
24
25 -     [h, w] = size(im);%get width
26
27 -    imshow(im);
28
29 -    Iprops=regionprops(im,'BoundingBox','Area', 'Image'); %read letter
30 -    count = numel(Iprops);
31 -    noPlate=[]; % Initializing the variable of number plate string.
32
33 - ⊟for i=1:count
34 -       ow = length(Iprops(i).Image(1,:));
35 -       oh = length(Iprops(i).Image(:,1));
36 -       if ow<(h/2) & oh>(h/3)
37 -           letter=Letter_detection(Iprops(i).Image); % Reading the letter corresponding the binary image 'N'.
38 -           noPlate=[noPlate letter] % Appending every subsequent character in noPlate variable.
39 -       end
40 -   └end
```
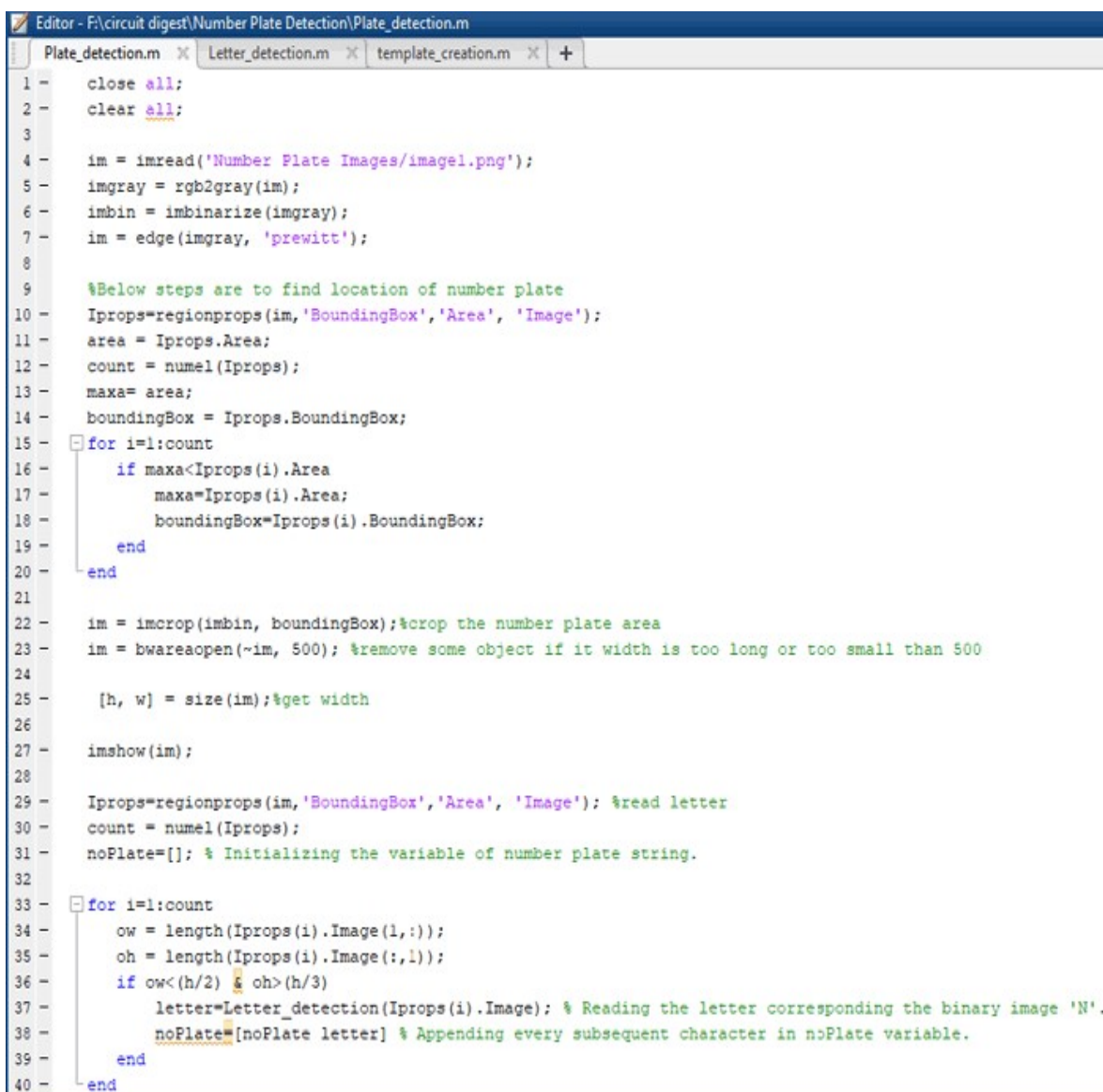
Basic commands used in above code are mentioned below:

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.**

Then the below code is used to **detect the location of the number plate in the entire input image**,

```
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
```

After that **crop the number plate and remove the small objects from the binary image** by using command *'imcrop()'* and *'bwareaopen()'* respectively.

Then, the below code is used to **process that cropped license plate image and to display the detected number in the image and text format** (in the command window).

```
Iprops=regionprops(im,'BoundingBox','Area', 'Image');
count = numel(Iprops);
noPlate=[];

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);
        noPlate=[noPlate letter]
    end
end
```

Working of Vehicle License Plate Number Detection System using MATLAB

In the *template_creation.m* file we have design the code to save all the binary images of alphanumerics into a directory or file named as '*NewTemplates*'. Then that directory is called in the *Letter_detection.m* as you can see in the below

```
1    function letter=readLetter(snap)
2
3 -    load NewTemplates
4 -    snap=imresize(snap,[42 24]);
5 -    rec=[ ];
6
7 -    for n=1:length(NewTemplates)
8 -        cor=corr2(NewTemplates{1,n},snap);
9 -        rec=[rec cor];
10 -   end
```
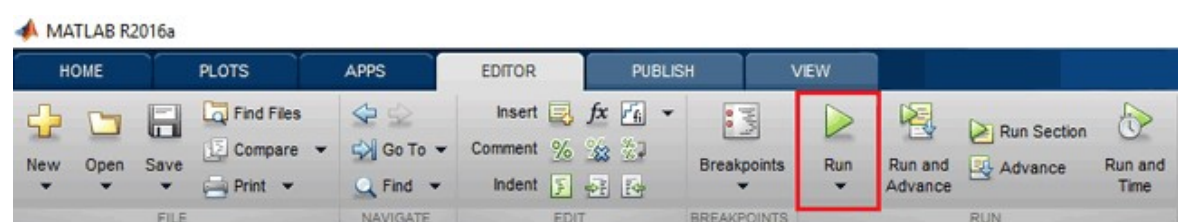
Then in the *Plate_detection.m* code file the *Letter_detection.m* code file is called when we process the image as shown in image below,
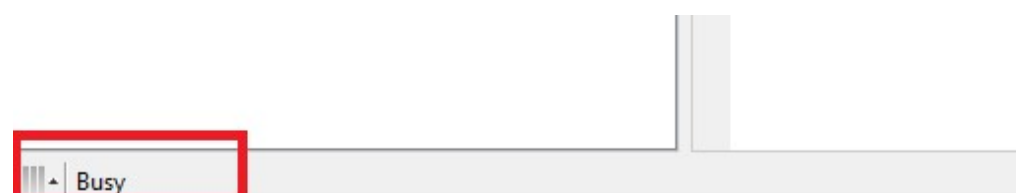
```
33 -    for i=1:count
34 -        ow = length(Iprops(i).Image(1,:));
35 -        oh = length(Iprops(i).Image(:,1));
36 -        if ow<(h/2) & oh>(h/3)
37 -            letter=Letter_detection(Iprops(i).Image);
38 -            noPlate=[noPlate letter]
39 -        end
40 -    end
```
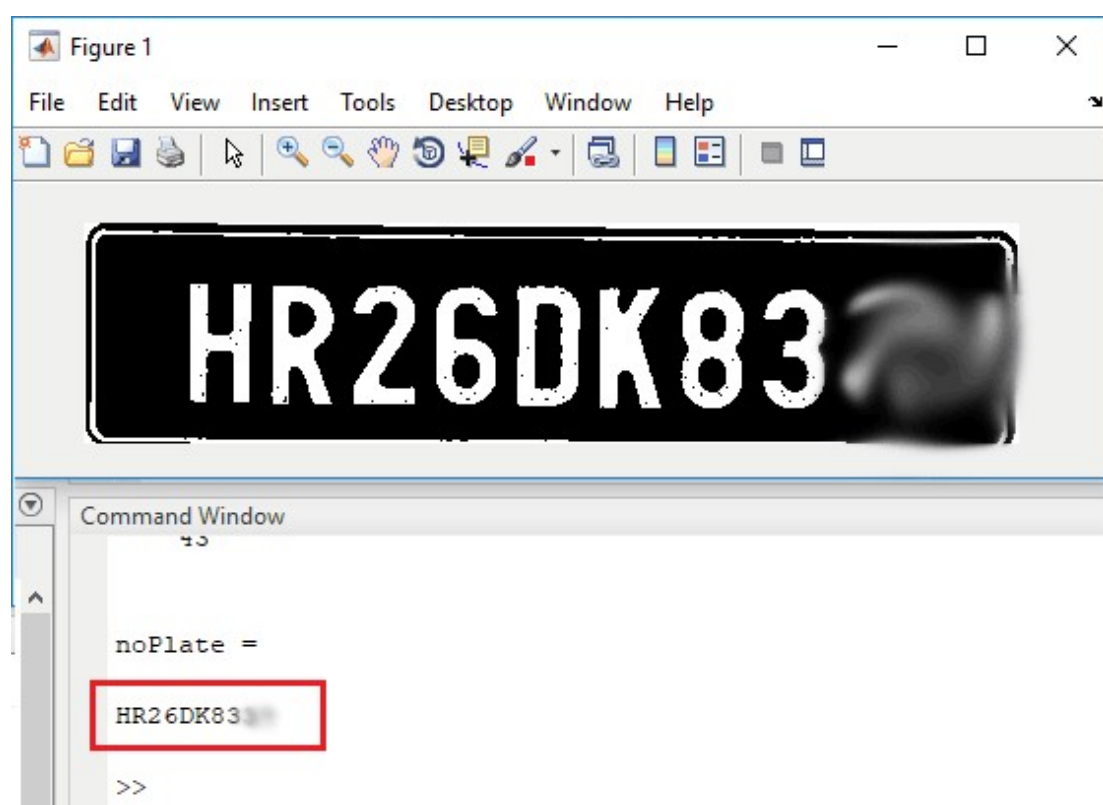
Now, click on the 'RUN' button to run the .m file


MATLAB may take few seconds to respond, wait until it shows busy message in the lower left corner as shown below,

As the program start you will get the number plate image popup and the number in the command window. The output for my image will look like the image given below;

Complete working of Vehicle License Number Plate Detection System is demonstrated in the Video below, and **all the code files with image templates can be downloaded from here (/sites /default/files/Matlab-Number-Plate-Detection.zip).**
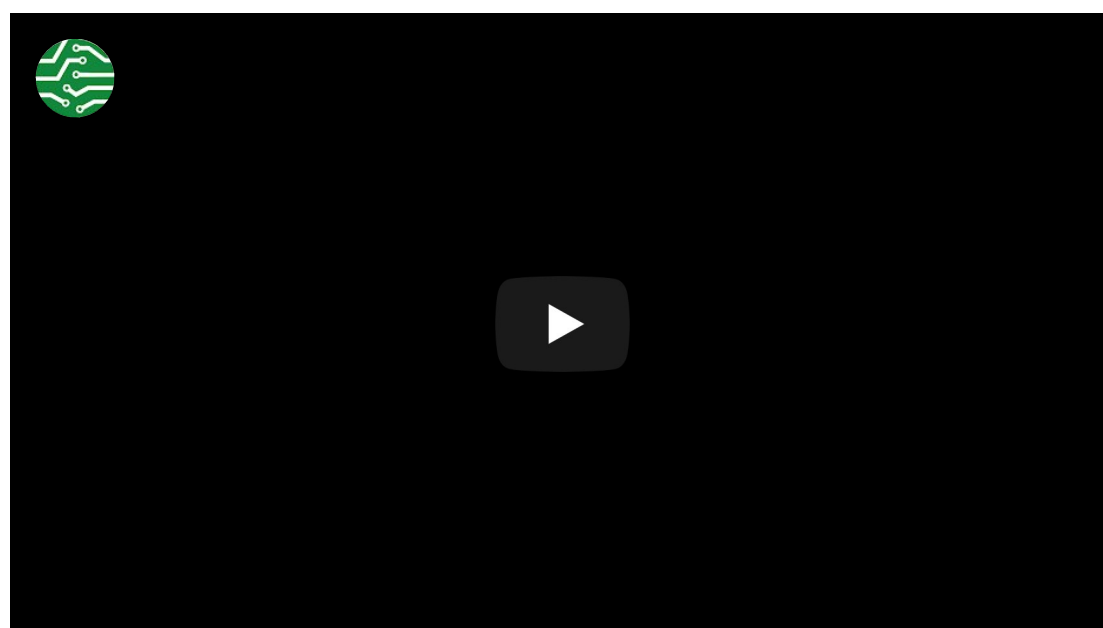
Also, check all MATLAB Projects here (http://circuitdigest.com/tags/matlab/).

Tags   MATLAB (/TAGS/MATLAB)   IMAGE PROCESSING (/TAGS/IMAGE-PROCESSING)
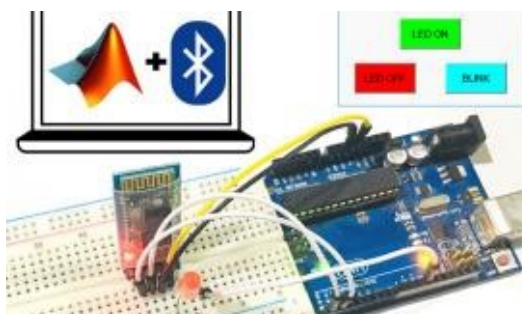
RELATED CONTENT
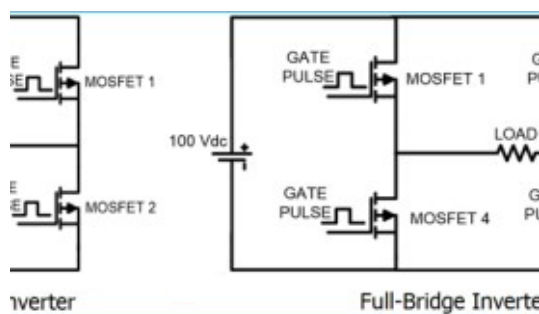
(/electronic-circuits/single-phase-half-bridge-and-full-bridge-inverter-circuit-using-matlab)

Single Phase Half Bridge and Full Bridge Inverter using MATLAB (/electronic-circuits /single-phase-half-bridge-and-full-bridge-inverter-circuit-using-matlab)
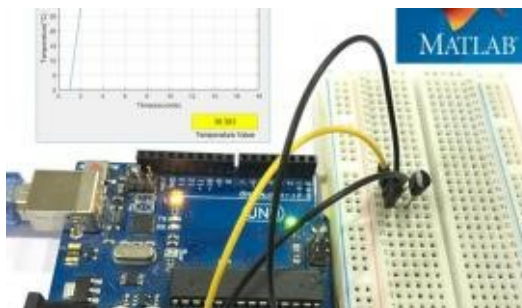

(/microcontroller-projects/how-to-use-bluetooth-with-matlab-for-wireless-communication)

How to use Bluetooth with MATLAB for Wireless Communication (/microcontroller-projects/how-to-use-bluetooth-with-matlab-for-wireless-communication)

(/tutorial/getting-started-with-image-

(/microcontroller-projects/how-to-plot-real-

processing-using-matlab)
Getting Started with Image Processing
using MATLAB (/tutorial/getting-started-
with-image-processing-using-matlab)

time-temperature-graph-using-matlab)
How to Plot Real Time Temperature Graph
using MATLAB (/microcontroller-projects
/how-to-plot-real-time-temperature-graph-
using-matlab)

(/tutorial/getting-started-with-simulink-in-
matlab)
Getting Started with Simulink in MATLAB:
Designing a Model (/tutorial/getting-
started-with-simulink-in-matlab)

(/microcontroller-projects/serial-
communication-between-matlab-and-
arduino)
Serial Communication between MATLAB
and Arduino (/microcontroller-projects
/serial-communication-between-matlab-
and-arduino)

(/microcontroller-projects/stepper-motor-
control-using-matlab-and-arduino)
Stepper Motor Control using MATLAB and
Arduino (/microcontroller-projects/stepper-
motor-control-using-matlab-and-arduino)
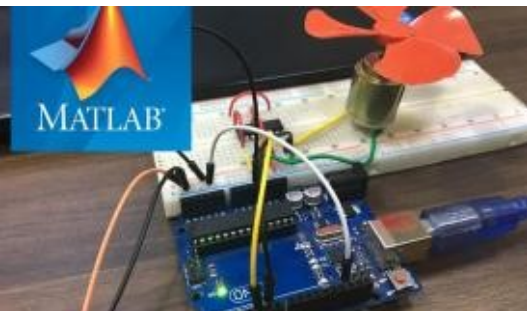
(/microcontroller-projects/matlab-dc-
motor-control-using-arduino)
DC Motor Control Using MATLAB and
Arduino (/microcontroller-projects/matlab-
dc-motor-control-using-arduino)

Get Our Weekly Newsletter!

Subscribe below to receive most popular news, articles and DIY projects from Circuit Digest

Email Address *

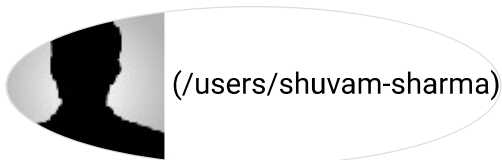**Name**

Country

United States of America

Subscribe

‹ PREVIOUS POST

Getting Started with Image
Processing using MATLAB
(https://circuitdigest.com/tutorial
/getting-started-with-image-
processing-using-matlab)

NEXT POST ›

Pull Up and Pull Down Resistor
(https://circuitdigest.com/tutorial
/pull-up-and-pull-down-resistor)

COMMENTS

(/users/shuvam-sharma)

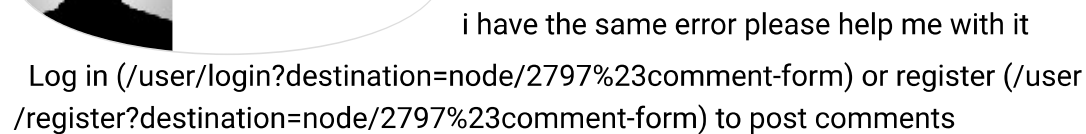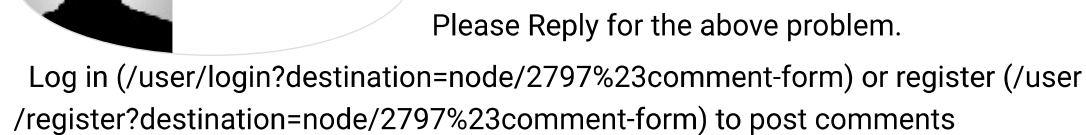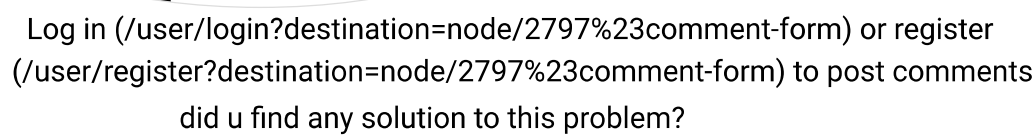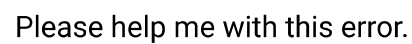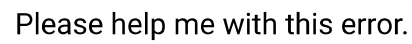**Shuvam Sharma (/users/shuvam-sharma)**
Apr 01, 2019

When i am running letter detection, it is

Log in (/user/login?destination=node/2797%23comment-form) or register (/user
/register?destination=node/2797%23comment-form) to post comments
showing :

Brace indexing is not supported for variables of this type.

Error in Letter_detection (line 8)
cor=corr2(NewTemplates{1,n},snap);

Will you please help me for the same..

Monisha (/users/monisha)
(/users/monisha)
Apr 04, 2019

Brace indexing is not supported for variables of this

type.

Error in Letter_detection (line 8)
cor=corr2(NewTemplates{1,n},snap);

Please help me with this error.

Monisha (/users/monisha)
(/users/monisha)
Apr 04, 2019

Brace indexing is not supported for variables of this

type.

Error in Letter_detection (line 8)
cor=corr2(NewTemplates{1,n},snap);

Please help me with this error.

Shuvam Sharma (/users/shuvam-sharma)
(/users/shuvam-sharma)
Apr 13, 2019

did u find any solution to this problem?

Monisha (/users/monisha)
(/users/monisha)
Apr 06, 2019

Please Reply for the above problem.

unknown (/users/unknown-0)
(/users/unknown-0)
May 04, 2019

i have the same error please help me with it

TI TRAINING VIDEOS

-
-
-
-
-
-
-
-
-
-
-

Connect with us on social media and stay updated with latest news, articles and projects!

f
(https://www.facebook.com/circuitdigest)

(https://twitter.com/circuitdigest)

(https://www.youtube.com/circuitdigest)

(https://www.instagram.com/circuitdigest/)

(https://in.pinterest.com/circuitdigest/)

in
(https://www.linkedin.com/company/circuit-digest/)

(https://www.youtube.com/UCy3CUAIYgZdAOG9k3IPdLmw)

CATEGORIES

Embedded Electronics (https://circuitdigest.com/embedded)

Power Electronics (https://circuitdigest.com/power-electronics)

Analog Electronics (https://circuitdigest.com/analog-electronics)

Internet of Things (https://circuitdigest.com/internet-of-things)

Audio Electronics (https://circuitdigest.com/audio-electronics)

POPULAR

ROBOTICS (/ROBOTICS-PROJECTS)

555 CIRCUITS (/555-TIMER-CIRCUITS)

ARDUINO PROJECTS (/ARDUINO-PROJECTS)

RASPBERRY PI PROJECTS (/SIMPLE-RASPBERRY-PI-PROJECTS-FOR-BEGINNERS)

ELECTRONICS NEWS (HTTPS://CIRCUITDIGEST.COM/NEWS)

ELECTRONICS FORUM (HTTPS://CIRCUITDIGEST.COM/FORUMS)

CALCULATORS (HTTPS://CIRCUITDIGEST.COM/CALCULATORS)

NEWSLETTER

Sign Up for Latest News

Enter your email

Subscribe

Hardware Startup? (/hardware-start-up-listing)

Privacy Policy (http://circuitdigest.com/privacy-policy) | Cookie Policy (https://circuitdigest.com/cookie-policy) | Terms of Use (https://circuitdigest.com/terms-of-use) | Contact Us (http://circuitdigest.com/contact) | Advertise (http://circuitdigest.com/advertise)