

T.C.
BURSA TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ
EEM0458 GÖRÜNTÜ İŞLEME DERSİ
DÖNEM PROJESİ

Fotoğraf Üzerinden Araç Plakası Okuma
Matlab İle Düşük Işıktaki Plaka Tespiti

160109061 Hüseyin EREN

Dersin Sorumlusu : Dr. Öğr. Üyesi Mustafa ÖZDEN

İçindekiler

.....	1
İçindekiler.....	2
İçindekiler	3
Özet	4
Giriş.....	5
Giriş.....	6
Kodlar/Aşamalar.....	7
Template_creation.m.....	8
Plate_detection.m.....	9
Plate_detection.m kodu.....	10
Plate_detection.m kodunun devamı.....	11
Letter_detection.m.....	12
Heuristic.m.....	13
Heuristic.m kodunun devamı.....	14
Heuristic.m kodunun devamı.....	15
Plate_detection.m.....	16
Orijinal Fotoğraf.....	17
Gri Seviyeye Dönüştürme İşlemi.....	18
Binary Seviyeye Dönüştürme İşlemi.....	19
Medyan Filtre İşlemi.....	20
Sobel Filtresi İşlemi.....	21
Kenarları Genişletme İşlemi.....	22
Dikdörtgen Alanları Doldurma İşlemi.....	23

İçindekiler

Plakanın Yerini Tespit Etme İşlemi.....	24
Şekil-9.....	25
Temple_creation.m ve alpha klasörü.....	26
Letter_detection.m.....	27
Letter_detection.m devam.....	28
Plate_detection.m matlab çıktısı.....	29
Letter_detection.m Analizi.....	30
Heuristic Dosyasının Çalışması.....	31
Heuristic Dosyasının Çalışma Aşaması.....	32
Plaka Tespit Etme Çalışmasının Son Aşaması.....	33
Sonuç ve Öneriler.....	34
Kaynakça.....	35

ÖZET

Proje için öncelikle bir konu seçtim.Listede bulunan Fotoğraf Üzerinden Araç Plakası Okuma konusu ilgimi çekti.Ardından bilgi kaynaklarından araştırmalara başladım ve heuristic adlı bir öğrenci çalışma grubunun projesini buldum proje seçtiğim alanda güzel bir örnek olacaktı.Gerekli dosyaları indirdikten sonra üzerinde kodların ve algoritmanın ne işe yaradığı ile ilgili bir çalışma yaptım bu uzun analizler sonucunda ortaya çıkan bilgileri ayrıştırarak belgeledim ve kodların belli bölümlerinde değişiklikler yaptım.Amaç fotoğrafı bulunan araç plakasının matlab kullanarak görüntü işleme alanındaki bilgilerden istifade ederek plakanın çıktısını elde etmektir.Medyan filtre,sobel filtre,binary seviye gibi teorik işlemleri gerçekleştirip sıralı işlemler sonucunda plakanın en istenen modelini ortaya çıkarttım.Bu işlemler 4 adet matlab kodu ile gerçekleştirildi.Heuristic.m uzantılı dosya da diğer 3 kod algoritmaları birleştirilip bir arayüzde kullanıcıya sunulmuştur.

Anahtar Kelimeler : Medyan,binary,korelasyon,indis

Giriş

- Öncelikle, plakaları tespit etmek için kullandığımız konsept hakkında size bilgi vereyim. Bu proje için üç program veya '.m' dosyası var. (1)
- **Şablon Oluşturma** (*template_creation.m*) - Bu, alfanümeriklerin kaydedilmiş görüntülerini çağırmak ve ardından bunları MATLAB belleğine yeni bir şablon olarak kaydetmek için kullanılır. (1)
- **Harf Algılama** (*Letter_detection.m*) - Giriş görüntüsünden karakterleri okur ve en yüksek eşleşen karşılık gelen alfanümerik *olanı* bulur. (1)
- **Plaka Algılama** (*Plate_detection.m*) - Görüntüyü işleyin ve ardından numarayı algılamak için yukarıdaki iki m dosyasını çağırın. (1)
- **mread ()** - Bu komut, görüntüyü hedef klasörden MATLAB'a açmak için kullanılır. (1)
- **rgb2gray ()** –Bu komut, RGB görüntüsünü gri tonlamalı biçime dönüştürmek için kullanılır. (1)
- **imbinarize ()** - Bu komut, 2-D gri tonlamalı görüntüyü ikiye ayırmak için kullanılır veya basitçe görüntüyü siyah beyaz formata dönüştürdüğünü söyleyebiliriz. (1)
- **edge ()** - Bu komut, Roberts, Sobel, Prewitt ve diğerleri gibi çeşitli yöntemler kullanılarak görüntüdeki kenarları tespit etmek için kullanılır. (1)
- **regionprops ()** - Bu komut, görüntü bölgesinin özelliklerini ölçmek için kullanılır. (1)
- **numel ()** - Bu komut, dizi elemanlarının sayısını hesaplamak için kullanılır.(1)
- **imcrop ()** - Bu komut, görüntüyü girilen boyutta kırmak için kullanılır.(1)
- **bwareaopen ()** - Bu komut, küçük nesneleri ikili görüntüden kaldırmak için kullanılır.(1)
- Kodda yukarıdaki komutları kullanarak, giriş görüntüsünü çağırıp gri tonlamaya dönüştürüyoruz. Daha sonra gri tonlama ikili görüntüye dönüştürülür ve ikili görüntülerin kenarı **Prewitt yöntemi** ile tespit edilir.(1)

ANPR (Otomatik Plaka Tanıma) sisteminin nasıl çalıştığını hiç merak ettiniz mi? Size bunun arkasındaki konsepti anlatayım, ANPR sisteminin kamerası araç plakasının görüntüsünü yakalar ve ardından görüntü, görüntünün bir metin formatına alfanümerik bir dönüşümünü sağlamak için çok sayıda algoritma ile işlenir. ANPR sistemi, Petrol Pompaları, Alışveriş Merkezleri, Havaalanları, otoyollar, gişeler, Oteller, Hastaneler, Otoparklar, Savunma ve Askeri kontrol noktaları vb. Birçok yerde kullanılmaktadır.

Bu Plaka tespiti için birçok görüntü işleme aracı mevcuttur, ancak burada bu eğitimde **araç plaka numarasını** metin formatına **almak için MATLAB Görüntü İşleme'yi** kullanacağız.(1)

Kodlar/Aşamalar

4 adet kodumuz var ve her biri bir aşamayı sağlayıp heuristic.m arayüzünde çalıştırılmaktadır.

- Heuristic.m
- Letter_detection.m
- Plate_detection.m
- Template_creation.m

Template_creation.m

imread: imread ile üzerinde çalışılmak istenilen fotoğraf bir değişkene atanarak matlab workspace için tanımlanır ve ardından imshow ile pencerede gösterilebilir. Workspace'te imgenin boyutu, çözünürlüğü gibi bilgileri görebilirsiniz.

- Alpha klasöründe bulunan harf fotoğraf dosyaları imread komutu ile harflere atanır.
- Alpha klasöründe bulunan doğal sayılar (0-9) imread komutu ile fotoğraflar ile eşleştirilir.
- Alfabe ve numara dizisi oluşturulur.
- NewTemplates değişkenine harf ve sayılar atanır.

```
%160109061-HUSEYİN-EREN
%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
```

Plate_detection.m

- İlk olarak örnek bir resim yüklenir. **im = imread('1.jpg');**
- Fotoğraflar ekranda gösterilir. **imshow(im)**
- Dosyaya Orijinal Resim başlıklı olarak isim verilir. **title("\color{red}Orijinal Resim")**
- Resim gri seviyeye dönüştürülür. **imgray = rgb2gray(im);**
- Dosyaya Gri Seviyeye Dönüştürme İşlemi adlı isim verilir. **title("\color{red}Gri Seviyeye Dönüştürme İşlemi")**
- Gri seviyeye dönen resim binary Seviyeye dönüştürülür. **imbin = imbinarize(imgray);**
- Dosyaya Binary Seviyeye Dönüştürme İşlemi adı verilir. **("\color{red}Binary Seviyeye Dönüştürme İşlemi")**
- Binary seviye işlemine uğrayan fotoğrafa resimdeki gürültüleri gidermek için Medyan filtre işlemi yapılır. **imgraynew = medfilt2(imgray);**
- Dosyaya Medyan Filtre İşlemi başlığı verilir. **title("\color{red}Medyan Filtre İşlemi")**
- En son haline resimdeki kenarları bulmak için sobel filtresi uygulanır. **im = edge(imgraynew, 'sobel');**
- Dosya Sobel Filtresi İşlemi olarak adlandırılır. **title("\color{red}Sobel Filtresi İşlemi")**
- Bulunan kenarlar kalınlaştırıp daha belirgin hale getirilir. **im=imdilate(im,strel('diamond',2));**
- İşleme kenarları genişletme işlemi olarak ad verilir. **title("\color{red}Kenarları Genişletme İşlemi")**
- Buğulu kenarlardan oluşan dikdörtgen alanları dolduruluyor. **im2=imfill(im,'holes');**
- Dosyaya Dikdörtgen Alanları Doldurma İşlemi adı verilir. **title("\color{red}Dikdörtgen Alanları Doldurma İşlemi")**
- Bulunan en büyük dikdörtgenli alanı binary resimden kırılır. **(im=imopen(im,strel('rectangle',[2 2]));)**

- Korelasyon işlemi gerçekleştirilir.
- Korelasyon bilinen bir şeklin varlığını tespit için kullanılır.
- Korelasyon işleminin uygun hale getirilmesi için resimdeki siyah alanlar beyaza beyaz alanlar siyaha dönüşürülmüştür.Sebebi ise korelasyon yapılması için kullanılan veriye benzetilmesidir

```

%160109061-HUSEYIN-EREN
close all;
clear all;
im = imread('1.jpg');
figure
imshow(im), title("\color{red}Oriijinal Resim")
imgray = rgb2gray(im);
figure
imshow(imgray), title("\color{red}Gri Seviyeye Dönüştürme İşlemi")
imbin = imbinarize(imgray);
figure
imshow(imbin) , title("\color{red}Binary Seviyeye Dönüştürme İşlemi")
imgraynew = medfilt2(imgray);
figure
imshow(imgraynew) , title("\color{red}Medyan Filtre İşlemi")
im = edge(imgraynew, 'sobel');
figure
imshow(im) , title("\color{red}Sobel Filtresi İşlemi")
im=imdilate(im,strel('diamond',2));
figure
imshow(im) , title("\color{red}Kenarları Genişletme İşlemi")
im2=imfill(im,'holes');
figure
imshow(im2) , title("\color{red}Dikdörtgen Alanları Doldurma İşlemi")
im=imopen(im,strel('rectangle',[2 2]));

%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
figure
imshow(im) , title("\color{red}Plakanın Yerini Tespit Etme İşlemi")
im = bwareaopen(~im, 500); %remove some object if it width is too long or too small than 500

```

Plate_detection.m kodu

```

im = bwareaopen(~im, 500); %remove some object if it width is too long or too small than 500
figure
imshow(im) ,
|
[h, w] = size(im);%get width

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
end

```

Plate_detection.m kodunun devamı

Letter_detection.m

- Temple_creation dosyasında oluşturulan NewTemplates matrisi indisler else if komutunda karşılaştırılarak alfabe ve 0-9'a doğal sayılar letter değişkenine aktarılır.
- İmresize ile kullanılan interpolasyon seçilip boyutlandırılmıştır.
- İnd'deki sayıların karşılaştırılmasına göre letter'da değişken (Harf/Sayı) almıştır.İleride tekrar değinilecektir.

```
%160109061-HUSEYİN-EREN
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';
```

```
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='0';
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
```

Heuristic.m

Bu dosyamız programımızın tüm yan dosyalarının harmanlandığı ve genel arayüz ile çalıştırdığımız kod dosyasıdır.

- logo = imread('btu.jpg'); ile BTU okul logosu arayüzün ortasına aktarılmıştır.

```
function varargout = Heuristic(varargin)
%HEURISTIC MATLAB code file for Heuristic.fig
%
% HEURISTIC, by itself, creates a new HEURISTIC or raises the existin
% singleton*.
%
% H = HEURISTIC returns the handle to a new HEURISTIC or the handle t
% the existing singleton*.
%
% HEURISTIC('Property','Value',...) creates a new HEURISTIC using the
% given property value pairs. Unrecognized properties are passed via
% varargin to Heuristic_OpeningFcn. This calling syntax produces a
% warning when there is an existing singleton*.
%
% HEURISTIC('CALLBACK') and HEURISTIC('CALLBACK',hObject,...) call th
% local function named CALLBACK in HEURISTIC.M with the given input
% arguments.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only on
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
%
% Edit the above text to modify the response to help Heuristic
%
% Last Modified by GUIDE v2.5 12-Dec-2019 13:42:03
%
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Heuristic_OpeningFcn, ...
                  'gui_OutputFcn',  @Heuristic_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% End initialization code - DO NOT EDIT

% --- Executes just before Heuristic is made visible.
function Heuristic_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin   unrecognized PropertyName/PropertyValue pairs from the
%            command line (see VARARGIN)
%
% Choose default command line output for Heuristic
handles.output = hObject;
%
% Update handles structure
guidata(hObject, handles);
%
% UIWAIT makes Heuristic wait for user response (see UIRESUME)
% uiwait(handles.figure1);
%
% --- Outputs from this function are returned to the command line.
function varargout = Heuristic_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
axes(handles.axes3)
logo = imread('btu.jpg');

imshow(logo)
% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
```

```

[filename pathname]=uigetfile({'*.jpg','*.png'},'Dosya Seçimi');
global aaa
aaa=strcat(pathname,filename);
axis off
axes(handles.axes1)
imshow(aaa)

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
axis off
axes(handles.axes2)
axis off
global aaa
a=imread(aaa);
im = imread(aaa);
imgray = rgb2gray(im);
imbin = imbinarize(imgray);
imgraynew = medfilt2(imgray);
im = edge(imgraynew, 'sobel');
im=imdilate(im,strel('diamond',2));
im2=imfill(im,'holes');
im=imopen(im,strel('rectangle',[2 2]));
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');
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
set(handles.edit3,'visible','on');
edit3= sprintf('%s', noPlate);
set(handles.edit3, 'String',edit3);

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)

```

```

function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%         str2double(get(hObject,'String')) returns contents of edit2 as a double

```

% --- Executes during object creation, after setting all properties.

```

function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

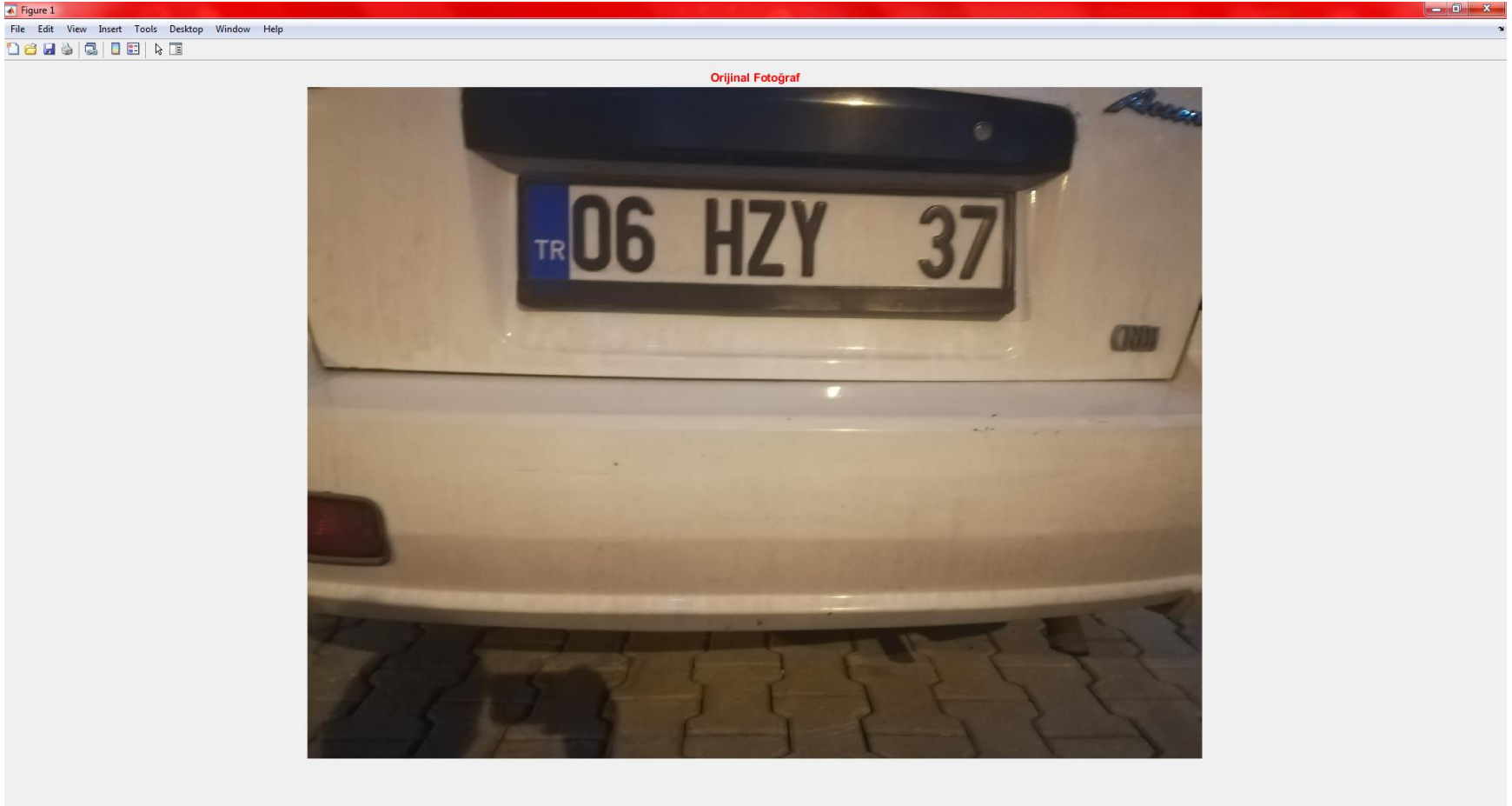
% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

Plate_detection.m

- Kodu çalıştırdığımızda yüklenen fotoğraf 8 aşamadan geçmektedir.
- Bu aşamalar fotoğraflar ile aşama aşama ilerleyerek açıklanacaktır.
- Orijinal Fotoğraf Yüklenmesi
- Gri seviyeye dönüştürme
- Binary seviyeye dönüştürme
- Medyan filtre işlemi
- Sobel filtre işlemi
- Kenarları genişletme işlemi
- Dikdörtgen alanları doldurma işlemi
- Plakanın yerini tespit etme işlemi

- `im = imread('1.jpg');` // İşlemi ile fotoğrafımızı okuyoruz
- `imshow(im), title("\color{red}Orijinal Fotoğraf")` // Orijinal Fotoğrafı Matlab'a yüklüyoruz. (Şekil-1)



Şekil-1

- `imshow(imgray), title("\color{red}Gri Seviyeye Dönüştürme İşlemi")`
- `imbin = imbinarize(imgray);`

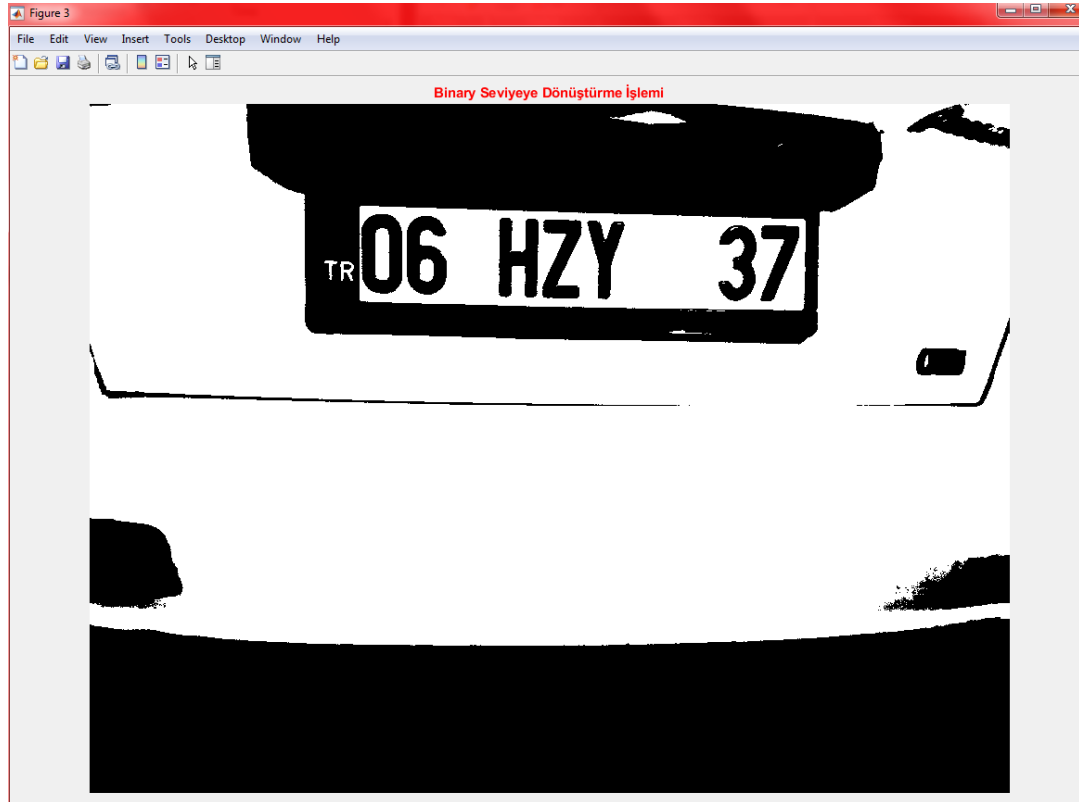
Bu kod işlemleri ile orijinal fotoğrafı gri seviyeye dönüştürüyoruz. (Şekil-2)



Şekil-2

- `imshow(imbin) , title("\color{red}Binary Seviyeye Dönüştürme İşlemi")`
- `imgraynew = medfilt2(imgray);`
- `figure`

Bu kodlar ile fotoğrafı Binary Seviyeye Dönüştürüyoruz.(Şekil-3)



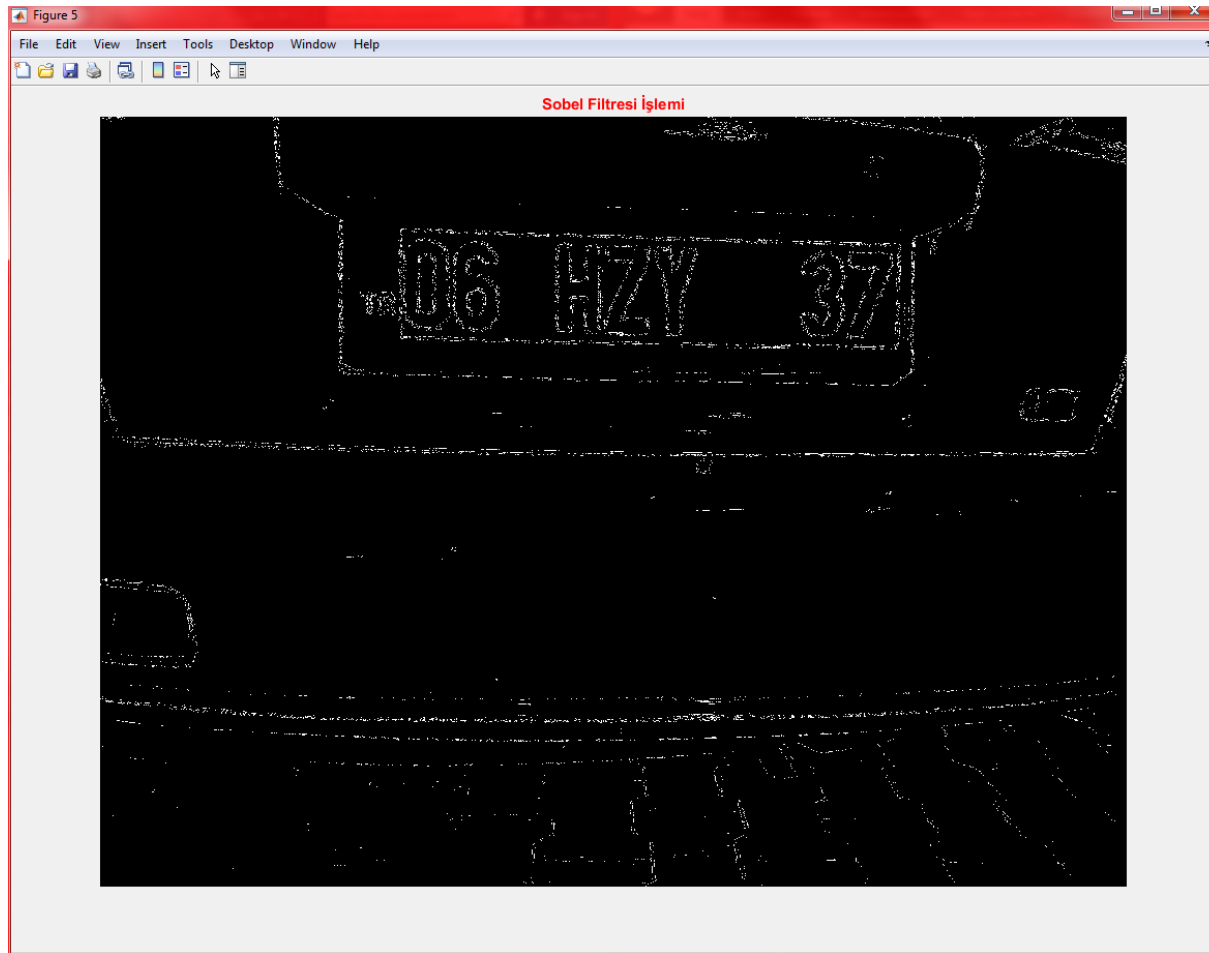
Şekil-3

- `imshow(imgraynew) , title("\color{red}Medyan Filtre İşlemi")`
 - `im = edge(imgraynew, 'sobel');`
 - `Figure`
- Bu kod işlemleri ile fotoğraftaki gürültüleri gidermek için Medyan filtre işlemi yapılır.(Şekil-4)



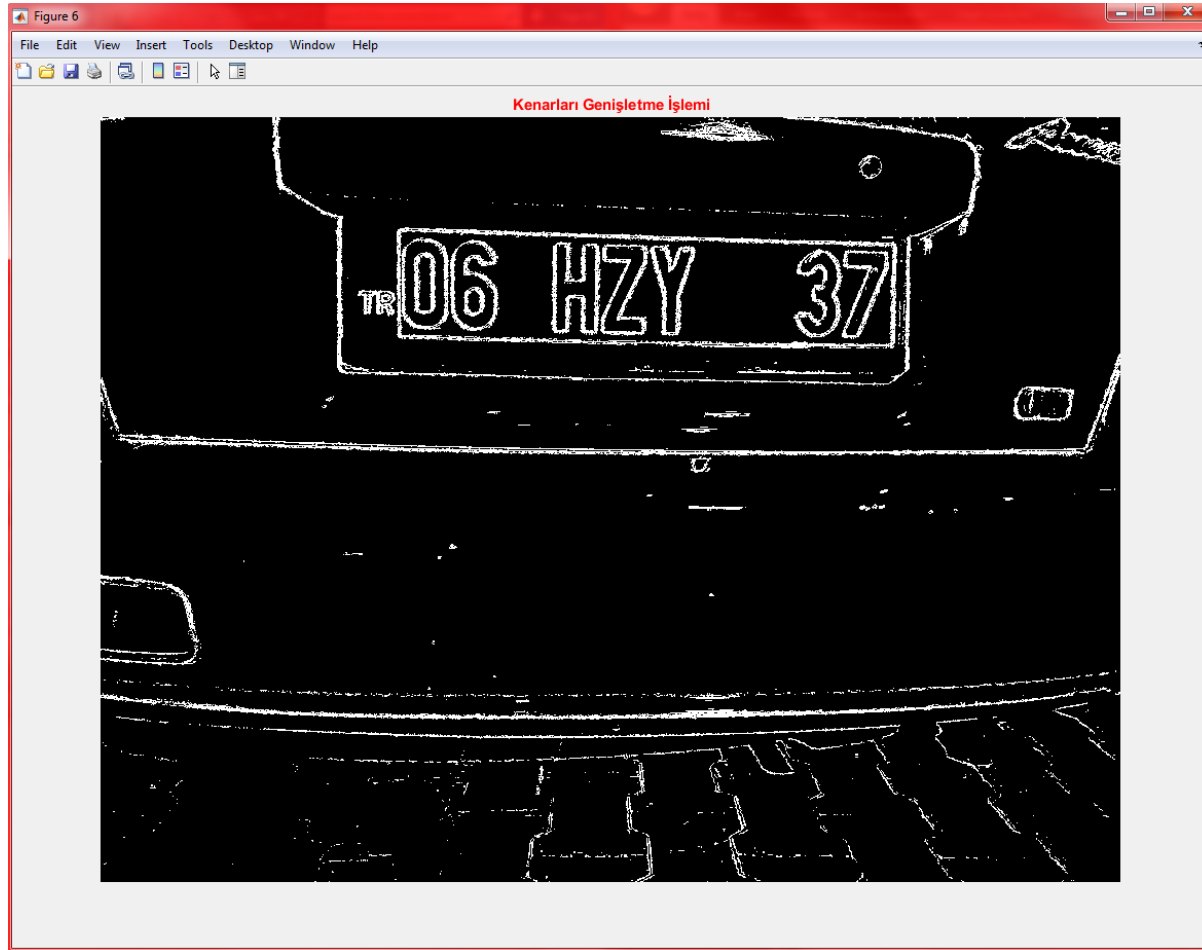
Şekil-4

- `imshow(im)` , `title("\color{red}Sobel Filtresi İşlemi")`
- `im=imdilate(im,strel('diamond',2));`
- `figure`
- Sobel filtresi işlemi ile resimdeki kenarları bulma işlemi gerçekleştirilir. (Şekil-5)
- Bulunan kenarlar kalınlaştırıp daha belirgin hale getirilir.



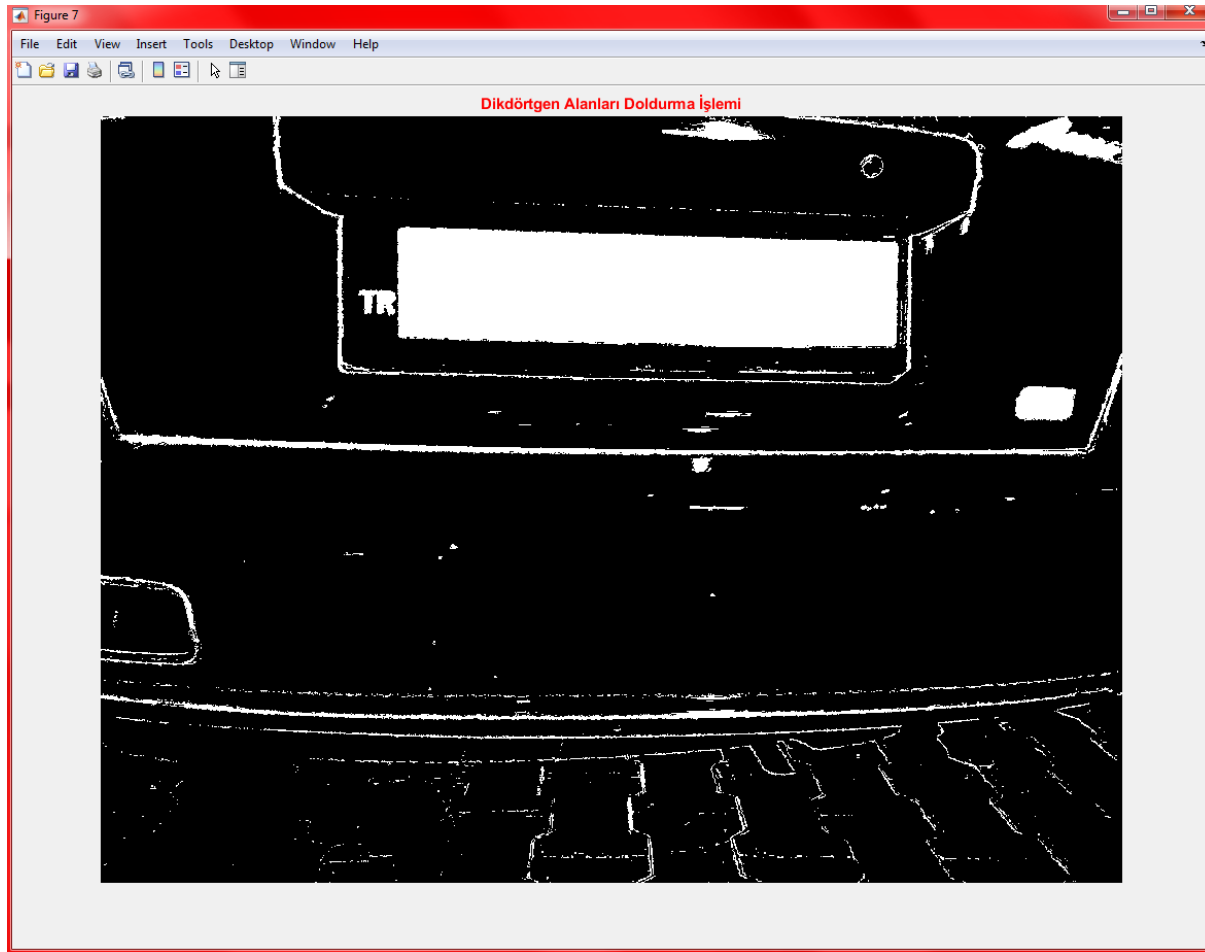
Şekil-5

- `imshow(im)` , `title("\color{red} Kenarları Geniřletme İřlemi")`
 - `im2=imfill(im,'holes');`
 - `figure`
- Görüntüyü bölümleyebilmek için bu alanları doldurmamız gerekir. Bunun için `imfill` işlemini gerçekleştirilmiştir. (Şekil-6)



Şekil-6

- `imshow(im2)` , `title("\color{red}Dikdörtgen Alanları Doldurma İşlemi")`
- `im=imopen(im,strel('rectangle',[2 2]));`
- Bulunan en büyük dikdörtgenli alanı binary resimden kırpılır. (Şekil-7)



Şekil-7

- `im = imcrop(imbin, boundingBox); %crop the number plate area`
- `figure`
- `imshow(im) , title("\color{red}Plakanın Yerini Tespit Etme İşlemi")` (Şekil-8)



Şekil-8



Şekil-9

Heuristic.m Arayüzünde kullanılacaktır.

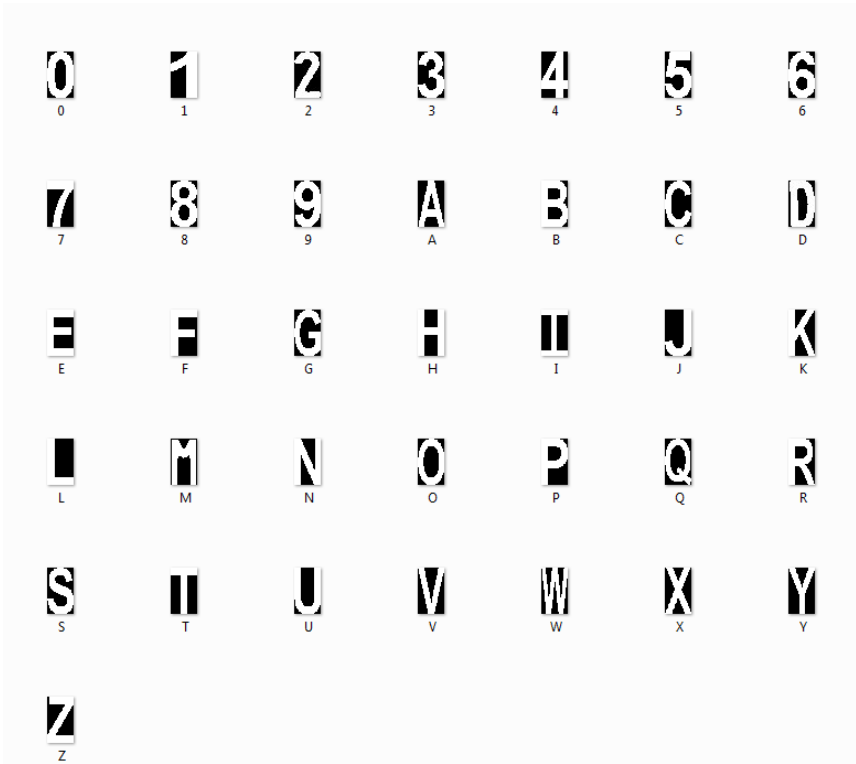
```
%CREATE TEMPLATES
%Alphabets
A=imread('alpha/A.bmp');B=imread('alpha/B.bmp');C=imread('alpha/C.bmp');
D=imread('alpha/D.bmp');E=imread('alpha/E.bmp');F=imread('alpha/F.bmp');
G=imread('alpha/G.bmp');H=imread('alpha/H.bmp');I=imread('alpha/I.bmp');
J=imread('alpha/J.bmp');K=imread('alpha/K.bmp');L=imread('alpha/L.bmp');
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
```

•Kodların bu bölümünde Alpha adlı klasörde bulunan 0-9 ve A-Z ye kadar olan bit işlem resim uzantılarını Matlab’a tanıtip ileride kullanmak üzere hafızaya atmıştır.



```

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';

```

```

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';

```

```

79 -     letter='9';
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

```

```
>> Plate_detection
```

```
ind =
```

```
49
```

```
noPlate =
```

```
'0'
```

```
ind =
```

```
40
```

```
noPlate =
```

```
'06'
```

```
ind =
```

```
11
```

```
noPlate =
```

```
'06H'
```

```
ind =
```

```
33
```

```
noPlate =
```

```
'06HZ'
```

Command Window

```
'06H'
```

```
ind =
```

```
33
```

```
noPlate =
```

```
'06HZ'
```

```
ind =
```

```
32
```

```
noPlate =
```

```
'06HZY'
```

```
ind =
```

```
36
```

```
noPlate =
```

```
'06HZY3'
```

```
ind =
```

```
43
```

```
noPlate =
```

```
'06HZY37'
```

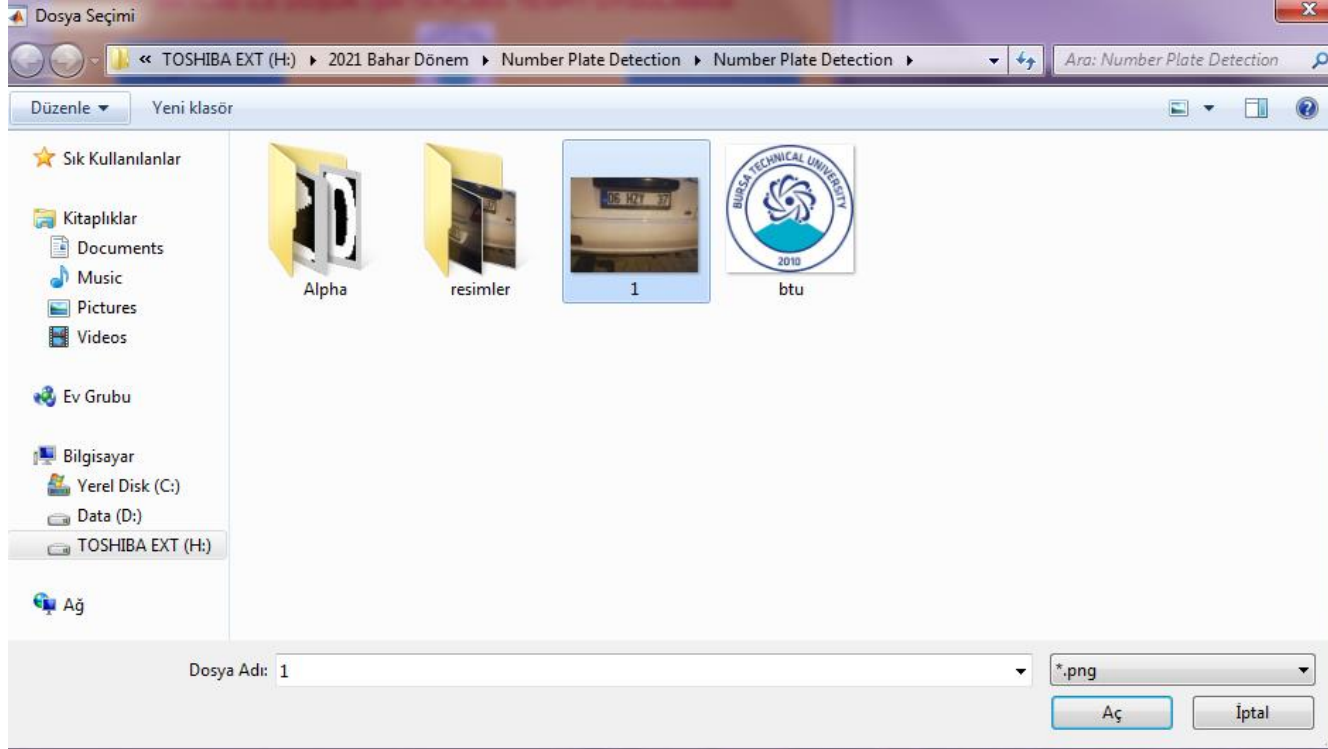
Letter_detection.m Analizi

- Letter detection bölümündeki else if yapısı ile plaka fotoğrafı taranmakta eşleşen harf ve sayılar
- Noplate'e sırasıyla eklenmekte ve ortaya adım adım plaka çıkmaktadır.
- Mesela 49 çıkan ind **0**'a tekabül etmekte devamında 40 **6**'ya 11 **H**'a 33 **Z**'ye 32 **Y**'ye 36 **3**'e 43 ise **7**'ye tekabül etmektedir.
- Koyu harflerle yazılan bu ind'ler birleştirildiğinde plakayı oluşturmaktadır.
- **0-6-H-Z-Y-3-7**
- **06HZY37**

HEURISTIC DOSYASININ ÇALIŞMASI



Arayüz ile birlikte imge yükle butonuna basarak tespit etmek istediğimiz plaka fotoğrafını analiz için matlab a aktarırız ve diğer çalışma dosyalarının da etkisiyle plaka tespit et butonuna bastığımızda plaka analiz edilerek analizin son aşama fotoğrafı ile birlikte alt kutucukta plakanın ne olduğu karşımıza çıkar.



- İmge Yükle butonuna basıldıktan sonra analizi yapılması istenen 1 adlı plaka fotoğrafını seçip aç'a basarız.

Plaka Tespit Etme Çalışmasının Son Aşaması



•Plate_detection,template_creation,letter_detection kod dosyalarının kullanımı ile matlab arayüzüne (Heuristic.m)'e aktarılan analiz ve bilgiler burada birleştirilip kullanıcıya en kolay şekilde kullanımına sunulmuştur.

SONUÇ VE ÖNERİLER

Projede amaçlanan fotoğraf üzerinden matlab kullanarak görüntü işleme metodu ile plaka tespiti gri seviyeye dönüştürme, binary seviyeye dönüştürme, medyan filtre işlemi, sobel filtre işlemi gibi bir çok işlemde geçen orijinal fotoğraf son aşamada matlab vasıtasıyla çıktı olarak alınmış arayüzde orijinal fotoğraf, şekil-9 ve plate detection olarak gösterilmiştir. Bu arayüz için 3 tane ek kod kaynağı oluşturulmuştur. Bu ek kod kaynakları Letter detection, plate detection, template creation'dır. Bunlar sırasıyla giriş görüntüsü karakterleri okuma ve en yüksek karşılık gelen alfanümerik olanı bulma, görüntüyü işleme ve sonrasında algılamak için letter_detection ve template_creation çağırma gibi işlevleri olan ek kod dosyaları ile amaçlanan eylem gerçekleştirilmiştir. Bu 3 ek kod kaynağı tek bir kod dosyası içinde yazılıp derleme işlemi hızlandırılabilir.

KAYNAKÇA

(1) Khatri P. Matlab ve Görüntü İşleme Kullanarak Araba Plaka Algılama
<https://circuitdigest.com/> 26.11.2018

ÜNAL A. ,DEMİRAL S. (2019) Konya Teknik Üniversitesi ,Elektrik-Elektronik Mühendisliği.
<https://www.mathworks.com/products/matlab.html>