



به نام خدا



دانشگاه تهران  
دانشکده مهندسی برق و کامپیوتر  
سیگنال و سیستم

گزارش شماره 1

نام و نام خانوادگی اعضا	مانا محمدی - ریحانه برهخت
شماره دانشجویی	810100096-810100207
تاریخ ارسال گزارش	1401/12/19

## فهرست گزارش

3	مقدمه
4	ماژول سوم
9	ماژول دوم
11	ماژول اول
13	کد نهایی
16	راهنمای اجرای فایل ها

هدف از انجام این پروژه استخراج اعداد و حروف پلاک خودرو های داخل ایران با استفاده از ویدیو و الگوریتم های پردازش تصویر می باشد.

به این منظور ما پروژه را به سه ماژول تقسیم نمودیم:

1- تشخیص ماشین و عکس برداری از پیکسل مناسب آن در ویدیو

2- تشخیص پلاک از روی جلو بندی ماشین

3- تشخیص اعداد و حروف پلاک ماشین از روی تصویر پلاک

برای ساده تر کردن انجام پروژه از ماژول سه شروع به انجام پروژه نمودیم.

## ماژول سوم: تشخیص اعداد و حروف پلاک ماشین از روی تصویر پلاک

### جمع آوری دیتابیس:

برای همبستگی یا کورولیشن گیری و تشخیص حروف ما نیاز به دیتابیس داریم که شامل حروف و اعداد موجود در پلاک ها می باشد. برای جمع آوری دیتابیس دو راه داریم: ۱- عکس گرفتن از پلاک های موجود برای حروف و کراپ و ادیت عکس ها ۲- تایپ اعداد در نهایت عکس های جمع آوری شده برای دیتابیس را تبدیل به عکس های سیاه و سفید با سایز های یکسان کردیم.



### مراحل پیاده سازی کد:

- 1- یک عکس پلاک از کاربر دریافت میکنیم و آن را میخوانیم.
- 2- سایز عکس را به مقدار مورد نظر تغییر میدهیم.
- 3- پلات عکس را قبل و بعد از تغییر سایز رسم میکنیم.
- 4- از دستور `rgb2gray` برای تبدیل فرمت رنگی تصویر به `gray scale` استفاده میکنیم و سپس پلات را رسم میکنیم و تصویر آن را نمایش میدهیم.

```
[file,path]=uigetfile({'*.jpg;*.bmp;*.png;*.tif'}, 'Choose an image');  
s=[path,file];  
picture=imread(s);  
figure  
subplot(1,2,1)  
imshow(picture)  
picture=imresize(picture,[300 500]);  
subplot(1,2,2)  
imshow(picture)
```

(1-3)

```
picture=rgb2gray(picture);  
figure  
subplot(1,2,1)  
imshow(picture)
```

(4)



## 5-im2bw:

تصاویر در متلب انواع مختلفی دارند . توابع مختلفی جهت تبدیل این تصاویر به هم وجود دارند که یکی از آن ها im2bw می باشد که برای تبدیل انواع تصاویر به نوع باینری که تنها دارای نقاط سیاه و سفید است به کار میرود . این تابع تصویر را ابتدا به نوع سیاه و سفید تبدیل میکند و در مرحله بعد آن را به یک تصویر باینری تبدیل میکند. تصویر باینری خروجی bw به ازای تمام پیکسل هایی که مقدار روشنایی آن ها کمتر از مقدار level باشد را مقدار ۰ (سیاه) و برای بقیه پیکسل ها مقدار ۱ (سفید) اختیار خواهد کرد. برای اینکه نقاط را سفید و زمینه را سیاه کنیم از علامت ~ که مانند NOT منطقی عمل میکند استفاده میکنیم.

```
picture = ~im2bw(picture,threshold);
```

(5)

## 6-graythresh:

برای بدست آوردن حد آستانه در im2bw از این تابع با ورودی تصویر مورد نظر استفاده میکنیم و سپس این level را به کار میگیریم. در نهایت دوباره تصویر تغییر یافته را نمایش میدهیم.

```
threshold = graythresh(picture);
```

(6)

نکته: مرحله 6 قبل از مرحله 5 انجام می شود.

## 7-bwareaopen:

این تابع دو ورودی دارد . یکی تصویر مورد نظر و دیگری یک المان ساختاری که مشخص کننده خصوصیات اجزایی است که باید از تصویر حذف شوند.(بزرگی ، شکل و ...)  
ما میخواهیم اجزایی که کمتر از 1500 پیکسل هستند از تصویر ما حذف شوند.

```
picture = bwareaopen(picture,1500);
```

(7)

8- در این مرحله برای بدست آوردن عکس بدون بک گراند بک گراند را از عکس کم میکنیم تا عکس خام پلاک به ما داده شود.

```
background=bwareaopen(picture,6000);
subplot(1,3,2)
imshow(background)
picture2=picture-background;
```



(8)

(8)

(9)

**Bwlabel**:

این تابع به ما اشیا متصل موجود در بی bw را میدهد. N میتواند 4 یا 8 باشد که در آن 4 نشان دهنده 4 شی متصل و 8 نشان دهنده 8 شی متصل می باشد.

**L:**

این متغیر ماتریسی را به ما میدهد که در آن پیکسل هایی که با 0 لیبل گذاری شده اند پس زمینه هستند. پیکسل هایی که با 1 لیبل گذاری شده اند شی اول را به ما می دهند. پیکسل هایی که با 2 لیبل گذاری شده اند شی دوم را تشکیل میدهند و ...

**Ne:**

این عدد تعداد اشیا متصل در bw را به عنوان یک عدد صحیح غیر منفی به ما می دهد.

```
[L,Ne]=bwlabel(picture2);
```

(9-1)

## regionprops(L,'BoundingBox')

این تابع مجموعه ای از ویژگی ها را برای هر منطقه لیبل گذاری شده در تصویر لیبل L اندازه گیری می کند. این تابع موقعیت و اندازه کوچکترین کادر حاوی ناحیه ای که المان در آن قرار دارد را به صورت بردار 1 در 2 در Q را در جایی که Q ابعاد تصویر است به ما برمیگرداند. اولین المان Q مختصات حداقل گوشه جعبه است. همچنین المان دوم آن اندازه جعبه در امتداد هر بعد است. این تابع در نهایت به ما سایز و موقعیت مستطیلی که دور کاراکتر ها کشیده میشود می دهد.

```
propied=regionprops(L,'BoundingBox');
```

(9-2)

## Rectangle

در نهایت با استفاده از این دستور و حلقه for و اطلاعات جمع آوری شده در مراحل قبل دور هر کدام از کاراکتر های پلاک مستطیل سبز رنگ کشیده می شود.

```
hold on
for n=1:size(propied,1)
    rectangle('Position',propied(n).BoundingBox,'EdgeColor','g','LineWidth',2)
end
hold off
```

(9-3)

-10

در این مرحله با استفاده از حلقه شروع به کرولیشن گیری میکنیم. یعنی میخواهیم متوجه شویم هر کدام از کاراکتر هایی که توسط مراحل قبل تشخیص داده شده است با کدام یک از داده های دیتا ست بیشترین مطابقت را دارد. train حاوی کاراکتر هایی است که در دیتاست وجود دارند که به صورت ماتریس های منطقی 42 در 24 در آمده اند.

## Corr2

این تابع ضریب همبستگی بین دو ماتریس موجود در **train** و تصویر کراپ شده را به ما میدهد. برای بیشترین مقدار ضریب همبستگی به دست آمده در حلقه یک عدد حداقل در نظر گرفته ایم که اگر کرولیشن ماکس از آن کمتر بود این ماتریس را با هیچ کدام از داده های دیتاست مطابقت نمیدهد. (0.45)

اگر ماکسیمم مقدار کرولیشن بیشتر از 0.45 باشد تطابق صورت میگیرد و کاراکتر تشخیص داده می شود .

## Cell2mat

این دستور برای تبدیل یک آرایه سلولی به آرایه عددی به کار می رود . در واقع با این تابع می توانیم یک آرایه سلولی را که خود شامل تعدادی ماتریس است به یک ماتریس تنها تبدیل کنیم.

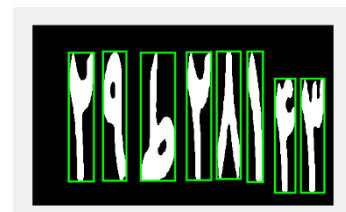
کاراکتر خوانده شده را هر مرحله را در **final-output** می نویسیم و در نهایت آن را نمایش می دهیم.

```
for n=1:Ne
    [r,c] = find(L==n);
    Y=picture2(min(r):max(r),min(c):max(c));
    imshow(Y)
    Y=imresize(Y,[42,24]);
    imshow(Y)
    pause(0.2)

    ro=zeros(1,totalLetters);
    for k=1:totalLetters
        ro(k)=corr2(TRAIN{1,k},Y);
    end

    [MAXRO,pos]=max(ro);
    if MAXRO>.45
        out=cell2mat(TRAIN{2,pos});
        final_output=[final_output out];
    end
end

print(final_output);
```





## ماژول دوم: تشخیص پلاک از روی جلو بندی ماشین

(برای تست این بخش باید فایل..... ران شود)

در این ماژول یک تابع نوشتیم که پلاک را از عکس جلوبندی ماشین تشخیص می دهد.

1-در ابتدا مانند ماژول 3 به ترتیب از توابع :

rgb2gray - graythresh - im2bw - bwareaopen  
bwlabel

استفاده کردیم تا بتوانیم نویزها را حذف کنیم و عکس را تبدیل به ماتریس کنیم(صفحات

5-7).

```
function Plate = DetectPlate(image)

%   I = imread('pelak3.jpg');

    picture=rgb2gray(image);

    figure()
    imshow(picture)

    threshold = graythresh(picture);
    picture =im2bw(picture,threshold);
    figure()
    imshow(picture)

    picture = bwareaopen(picture,15000);
    figure();
    imshow(picture)

    [L,Ne]=bwlabel(picture);
```



2-در این مرحله ابتدا یک structure تعریف کردیم که اشیایی که مشخصاتی شبیه به پلاک دارند در آن قرار گیرند تا بتوانیم در مرحله بعد شی ای که بیشترین شباهت را به پلاک دارد از بین آن ها تشخیص دهیم. مشخصات معیار ما در این مرحله ابعاد و مختصات پلاک است. از متغیر num برای شمردن تعداد اشیایی که در structure ما قرار میگیرند

استفاده می کنیم و مقدار اولیه آن را 0 قرار می دهیم .

```
Plate = struct('x',zeros(Ne,1), 'y', zeros(Ne,1), 'width', zeros(Ne,1), 'height', zeros(Ne,1));  
num=0;
```



3-در این مرحله یک حلقه نوشتیم تا بتوانیم اشیای مستطیل شکلی را که نسبت ابعاد آن ها مانند نسبت ابعاد پلاک است را بیابیم و در استراکچر بریزیم. از آنجایی که در پلاک بین قسمت کد شهر و پلاک اصلی خط وجود دارد ممکن است برنامه دچار خطا شود و هرکدام از این قسمت ها را یک object مجزا تشخیص دهد. در نتیجه چون ابعاد کد شهر کوچکتر از پلاک اصلی می باشد از فیلتر حذف خواهد شد. پس برای اینکه برنامه پلاک کامل را برای ما تشخیص دهد طبق ابعاد پلاک معیار فارسی طول پلاک را 5.5 برابر عرض آن در نظر گرفته و ناحیه انتخابی خودمان را از نقطه شروع پلاک با ابعاد عرض تشخیص می دهیم و 5.5 برابر آن را به عنوان به طول قرار می دهیم.

```
for n=1:size(propied,1)  
    if (propied(n).BoundingBox(3) >590) && (propied(n).BoundingBox(3) < 1360) && (propied(n).BoundingBox(4) >120) && (propied(n).BoundingBox(4) < 300)  
  
        num=num+1;  
        temp = [propied(n).BoundingBox(1), propied(n).BoundingBox(2), propied(n).BoundingBox(4)*5.5, propied(n).BoundingBox(4)];  
        Plate.x(num) = propied(n).BoundingBox(1);  
        Plate.y(num) = propied(n).BoundingBox(2);  
        Plate.width(num) = propied(n).BoundingBox(4)*5.5;  
        Plate.height(num) = propied(n).BoundingBox(4);  
        rectangle('Position',temp,'EdgeColor','g','LineWidth',2);  
    end  
end
```



## ماژول اول: تشخیص جلوبندی ماشین و عکس برداری از پیکسل مناسب آن در ویدیو

1- در این قسمت ابتدا توسط `videoReader` ویدیو را دریافت می کنیم .

```
obj = VideoReader('traffic.mp4');
```

2- سپس توسط حلقه `while` ویدیو را فریم به فریم میخوانیم و اطلاعات هر فریم از جمله `colormap` را در استراکچری که قبل از حلقه تعریف کردیم مینویسیم.

```
while obj.CurrentTime<=27

    s(k).cdata = readFrame(obj);
    s(k).cdata = imrotate(s(k).cdata , -90)
    %imshow(s(k).cdata);
    k= k+1;
end

s = struct('cdata',zeros(obj.Height, obj.Width, 3),'colormap', []);
```

3- سپس در حلقه `for` از پیکسل دوم شروع می کنیم و هر پیکسل را از پیکسل قبلی آن کم میکنیم و در استراکچر میریزیم تا بتوانیم بک گراند یا همان اجسام ثابت را تشخیص دهیم و ماشین را که در حال حرکت می باشد از بقیه اجسام موجود در ویدیو جدا کنیم.

```
dif(i).cdata = s(i).cdata - s(i-1).cdata;
```

4- سپس توسط توابع `graythresh` , `im2bw` تفاضل دو پیکسل را باینری می کنیم.

```
threshold = graythresh(dif(i).cdata);
dif(i).cdata = im2bw(dif(i).cdata,threshold);
```

5-strel: این تابع تولیدکننده یک المان ساختاری می باشد که از آن در imopen استفاده می کنیم و نویز های موجود در تصویر را تا حد امکان پاک می کنیم و در temp قرار می دهیم.

```
SE_Filter = strel('disk',1);  
temp = imopen(dif(i).cdata , SE_Filter);
```

6- در نهایت باید جلوبندی ماشین را از تصویر بدست آمده از مرحله قبل بدست آوریم. برای این کار بازه ای که ماشین در آن قرار دارد را برابر با بازه ای که در آن ماتریس ما 1 است قرار می دهیم و بازه افقی و عمودی آن را بدست می آوریم.

```
[nonZeroRows,nonZeroColumns] = find(temp);  
topRow = min(nonZeroRows(:));  
bottomRow = max(nonZeroRows(:));  
leftColumn = min(nonZeroColumns(:));  
rightColumn = max(nonZeroColumns(:));  
  
dif(i).cdata = s(i).cdata(topRow:bottomRow, leftColumn:rightColumn);  
imshow(dif(i).cdata)
```

## کد نهایی

در نهایت هدف ما این است که ویدیویی به برنامه بدهیم و برنامه پلاک ماشین در حال عبور را تشخیص دهد. برای این کار باید به ترتیب خروجی ماژول سه را به دو و دو را به یک بدهیم .

1- برای این امر ابتدا ماژول سه را تبدیل به یک تابع کردیم و آن را `find car` نامیدیم. ورودی این تابع ویدیو می باشد.(صفحه 4-8)

```
function dif = FindCar(obj)
s = struct('cdata',zeros(obj.Height, obj.Width, 3),'colormap', []);

k=1;
while hasFrame(obj)
s(k).cdata = readFrame(obj);
%s(k).cdata = imrotate(s(k).cdata , -90);
k= k+1;
end

dif = struct('cdata',zeros(obj.Height, obj.Width, 3),'colormap', []);
figure();

for i=1 : size(s,2)-1
imshow(s(i).cdata);
imshow(s(i).cdata);
dif(i).cdata = s(i+1).cdata - s(i).cdata;
imshow(dif(i).cdata)

threshold = graythresh(dif(i).cdata);
dif(i).cdata = im2bw(dif(i).cdata,threshold);

SE_Filter = strel('disk',1);
temp = imopen(dif(i).cdata , SE_Filter);

[nonZeroRows,nonZeroColumns] = find(temp);
topRow = min(nonZeroRows(:));
bottomRow = max(nonZeroRows(:));
leftColumn = min(nonZeroColumns(:));
rightColumn = max(nonZeroColumns(:));

s(i).cdata = s(i).cdata(topRow:bottomRow, leftColumn:rightColumn);
imshow(s(i).cdata)
end
end
```

2- سپس ماژول 2 را به یک تابع تبدیل کردیم و آن را `detectPlate` نامیدیم. ورودی این تابع یک تصویر می باشد که همان تصویر جلوبندی ماشین است.(صفحه 9-10)

3- در این مرحله یک تابع مینویسیم که به ما تصویر پلاک را می دهد.`final-output` ماتریسی است که به عنوان پلاک نهایی در نظر می گیریم. در این تابع مشخصات `object`

هایی را که به عنوان پلاک در نظر گرفته شده اند را چک می کند و بهترین پلاک را برای ما کراپ می کند .

```
function [final_output , final_accuracy] = DetectRead(I)
    imresize(I,[4000,2000]);
    figure(2);
    imshow(I);

    final_output = [];
    final_accuracy = 0;

    MyPlate = DetectPlate(I);

    if(size(MyPlate.x)~=0)
        for i=1 : size(MyPlate)

            if(MyPlate.width(i) == 0)
                break;
            end
            MyIm = imcrop(I, [MyPlate.x(i), MyPlate.y(i), MyPlate.width(i), MyPlate.height(i)]);
            figure(6);
            imshow(MyIm);

            [num, acc] = ReadPlate(MyIm);
            if(acc > final_accuracy)
                final_output = num;
                final_accuracy = acc;
            end
        end
    end
end
```

4-در نهایت ماژول سوم را تبدیل به تابعی کردیم که عکس پلاک کراپ شده را می گیرد و آن را می خواند.

```
obj = VideoReader('car3.mp4');

Cars = FindCar(obj);

Best_Output = 0;
Best_Accuracy = 0;

for i=1 : size(Cars, 2)
    figure(1)
    imshow(Cars(i).cdata);
    [output, acc] = DetectRead(Cars(i).cdata);
    if acc > Best_Accuracy
        Best_Output = output;
    end
end

Best_Output
```

## راهنمای اجرای فایل ها

1- برای تست تشخیص شماره پلاک از روی عکس پلاک باید فایل `ReadCode` ران شود.

2- برای تست تشخیص پلاک از روی جلوبندی باید `FindPlate` ران شود.

3- برای تست تشخیص شماره پلاک از روی جلوبندی باید `final` ران شود که در آن از توابع `DetectRead, DetectPlate, ReadPlate` استفاده شده.

4- برای تست تشخیص ویدیو باید `FinalCode` ران شود که در آن از `FindCar` و `DetectRead` استفاده شده.

