

تشخیص پلاک انگلیسی:

تشکیل دیتابیس در متلب:

در ابتدای کار لازم است تا داده‌های خود را در متلب لود کنیم و در آن یک دیتابیس به نام TRAIN ایجاد کنیم؛ برای اینکار ابتدا از تابع `dir` استفاده می‌کنیم و به آن آدرس پوشه که داده‌هایمان در آن قرار دارد را می‌دهیم؛ خروجی این تابع، مجموعه‌ای از `struct` است که شامل موارد از جمله نام فایل‌های موجود در پوشه، تاریخ و تعداد بایت آنها است.

Fields	name	folder	date	bytes	isdir	datenum
1	.'	'C:\Users\A...	'22-Nov-20...	0	1	7.3885e+05
2	'.'	'C:\Users\A...	'26-Nov-20...	0	1	7.3885e+05
3	'0.bmp'	'C:\Users\A...	'06-Jan-20...	230	0	7.3634e+05
4	'1.bmp'	'C:\Users\A...	'06-Jan-20...	230	0	7.3634e+05
5	'2.bmp'	'C:\Users\A...	'06-Jan-20...	230	0	7.3634e+05
6	'3.bmp'	'C:\Users\A...	'06-Jan-20...	230	0	7.3634e+05
7	'4.bmp'	'C:\Users\A...	'06-Jan-20...	230	0	7.3634e+05
8	'5.bmp'	'C:\Users\A...	'06-Jan-20...	230	0	7.3634e+05
9	'6.bmp'	'C:\Users\A...	'06-Jan-20...	230	0	7.3634e+05
10	'7.bmp'	'C:\Users\A...	'06-Jan-20...	230	0	7.3634e+05
11	'8.bmp'	'C:\Users\A...	'06-Jan-20...	230	0	7.3634e+05
12	'9.bmp'	'C:\Users\A...	'06-Jan-20...	230	0	7.3634e+05
13	'A.bmp'	'C:\Users\A...	'06-Jan-20...	230	0	7.3634e+05
14	'B.bmp'	'C:\Users\A...	'06-Jan-20...	230	0	7.3634e+05

مجموعه‌ای از `struct` فایل‌ها

در ادامه نام فایل‌های موجود را در متغیری به نام `name` ذخیره می‌کنیم. (همانطور که در تصویر بالا دیده می‌شود، دو عضو ابتدایی، شامل اسامی نیست پس برای اینکار از عضو سوم مجموعه‌ی داده شده استفاده می‌کنیم.) داده‌ی TRAIN را به صورت یک سلول 2 در تعداد فایل‌های دیتابیس، تعریف می‌کنیم و در سطر اول آن داده‌ها که به صورت عکس هستند و در سطر دوم نام متناظر با آن را می‌نویسیم.

	1	2	3	4	5	6	7	8	9	10	11
1	42x24 logical	42x24 logical	42x24 logical	42x24 logical	42x24 logical	42x24 logical	42x24 logical	42x24 logical	42x24 logical	42x24 logical	42x24 logical
2	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	'A'
3											
4											
5											
6											

اعضای نهایی داده TRAIN

```

1  clc;
2  clear;
3  close all;
4
5  directory=dir('Map Set');
6  st={directory.name};
7  name=st(3:end);
8  len=length(name);
9
10 TRAIN=cell(2,len);
11 for i=1:len
12     TRAIN(1,i)=imread(['Map Set','\ ',cell2mat(name(i))]);
13     temp=cell2mat(name(i));
14     TRAIN(2,i)={temp(1)};
15 end
16
17 save('TRAININGSET.mat','TRAIN');
```

در انتها، دیتابیس تشکیل شده در متلب را با دستور `save`، ذخیره می‌کنیم. کد نوشته شده برای توضیحات بالا در روبه‌رو آمده است. (در ابتدای این کد، تمامی متغیرهای تشکیل شده از پیش را `clear` می‌کنیم و فایل‌های باز را می‌بندیم).

آماده‌سازی داده‌ی تست:

انتخاب داده تست: در فایل main.m ابتدا داده‌ی TRAIN را لود می‌کنیم و تعداد اعضای آن را نیز ذخیره می‌کنیم؛ سپس توسط دستور uigetfile، پنجره‌ای باز می‌شود تا از آنجا فایلی که دارای یکپاز پسوندهای داده شده است را انتخاب کنیم؛ محتوا و آدرس این فایل در متغیر pic ذخیره می‌شود و دستور imread، محتوای آن تصویر را خوانده و در متغیر picture ذخیره می‌کند. از figure, subplot و imshow برای نمایش فایل تصویر استفاده می‌شود. در ادامه ابعاد تصویر را به 300 در 500 تغییر می‌دهیم.

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

```
picture=rgb2gray(picture);
threshold = graythresh(picture);
picture = ~imbinarize(picture,threshold);
picture = bwareaopen(picture,200);
background=bwareaopen(picture,3000);
picture2=picture-background;
```

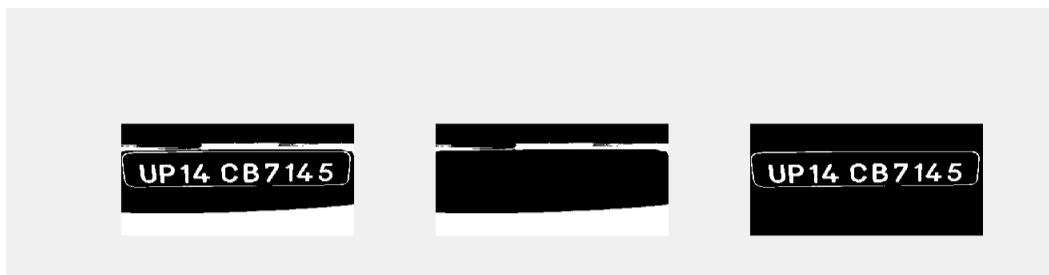
تبدیل تصویر به **bitmap** و واضح‌سازی آن: ابتدا توسط دستور **rgb2gray** تصویر را سیاه و سفید (شامل طیف خاکستری می‌کنیم) و با بدست آوردن **threshold** آن، آن را به تصویری باینری (هر پیکسل سیاه یا سفید) تبدیل می‌کنیم. در ادامه توسط دستور **bwareopen** نقاط سفیدی از تصویر که اندازه‌شان کمتر از 200 پیکسل باشد را سیاه می‌کنیم؛ همچنین از این دستور برای جدا کردن

قاب پلاک استفاده می‌کنیم به این صورت که ابتدا قسمت‌هایی از تصویر که اندازه‌شان کوچکتر از 3000 است (تمامی اجزا جز قاب) جدا می‌کنیم و نتیجه را در **background** می‌ریزیم و سپس آن را از کل تصویر کم می‌کنیم تا تصویر پلاک بدون قاب بدست آید.



نمونه تغییر سایز تصویر

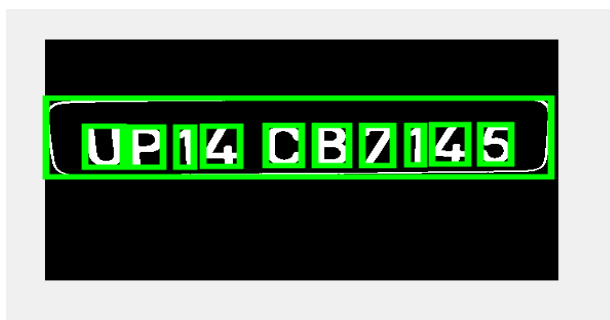




نمونه حذف دور پلاک (که در اینجا فقط قسمتی حذف شده)

لیبل زدن به اجزای به هم پیوسته: از تابع `bwlabel` استفاده می‌کنیم؛ این تابع دو خروجی دارد: خروجی اول شامل اجزای لیبل زده شده و خروجی دوم شامل تعداد این اجزا است. در واقع به ازای هر عضو، پیکسل‌هایی که آن عضو را به هم متصل کرده است را با یک عدد یکسان لیبل می‌زند؛ این عدد از 1 شروع می‌شود و بستگی به تعداد اجزای پیداشده ادامه می‌یابد. در ادامه، توسط دستور `rectangle`، دور اجزای پیدا شده مستطیل‌هایی رسم می‌کنیم تا قابل تشخیص شوند. (این بخش صرفاً برای دیدن ما و اطمینان از صحت کارکرد اضافه شده است.)

```
figure
imshow(image)
[L,Ne]=bwlabel(image);
propied=regionprops(L,'BoundingBox');
hold on
for n=1:size(propied,1)
    rectangle('Position',propied(n).BoundingBox,'EdgeColor','g','LineWidth',1)
end
hold off
```



نمونه اجزا پیدا شده در یک پلاک

تصمیم‌گیری درمورد اجزای پلاک:

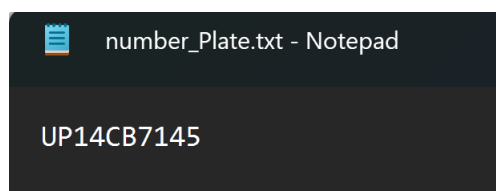
```
final_output=[];
for n=1:Ne
    [r,c] = find(L==n);
    Y=picture2(min(r):max(r),min(c):max(c));
    imshow(Y)
    Y=imresize(Y,[42,24]);
    imshow(Y)
    pause(0.2)

    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
```

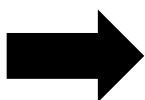
در این بخش به ازای هر جزء پیدا شده در قسمت قبل، شماره تمام ردیف‌ها و ستون‌هایی که پیکسل‌های این جزء در آن واقع شده‌اند در متغیر r و c ذخیره می‌شوند و آن طبق مختصات گوشه‌ی بالا سمت چپ و پایین سمت راست، این جزء را از تصویر اصلی جدا کرده و در متغیر Y می‌ریزیم. حال سایز این متغیر را مطابق با اندازه تصویر دیتابیس (که در اینجا 42 در 24 پیکسل است) تغییر می‌دهیم. متغیر ro را به شکل یک آرایه 1 در تعداد اعضای دیتابیس تعریف می‌کنیم و اعضای آن را با 0 مقداردهی می‌کنیم. به ازای هر عضو دیتابیس، $correlation$ آن را با تصویر Y پیدا کرده و در آرایه‌ی ro ذخیره می‌کنیم؛ حال ماکزیمم مقدار این آرایه را به همراه اندیس آن پیدا می‌کنیم و در صورتی که این مقدار ماکزیمم از یک مقدار مشخص (در اینجا 0.45) بیشتر بود، نام آن عضو را به متغیر $final_output$ اضافه می‌کنیم.

در انتها یک فایل تکست باز کرده و مقدار نهایی متغیر $final_output$ را در آن می‌ریزیم.

```
file = fopen('number_Plate.txt', 'wt');
fprintf(file,'%s\n',final_output);
fclose(file);
winopen('number_Plate.txt')
```



خروجی نشان داده شده برای پلاک



نمونه جزء تطبیق داده شده با دیتابیس

تشخیص پلاک فارسی:

تشکیل دیتابیس در متلب:

برای این قسمت مانند پلاک انگلیسی عمل می‌کنیم با این تفاوت که در اینجا دو دیتابیس جداگانه، یکی برای حروف و یکی برای اعداد تشکیل می‌دهیم و همچنین یک دیتابیس کلی که شامل هر دو آنهاست نیز می‌سازیم. کد دیتابیس‌های حروف و عدد مانند قبل است و کد دیتابیس

کلی به شکل
روبه‌رو است.

```
TRAIN=cell(2,len);
for i=1:numLen
    TRAIN(1,i)={imread(['numbers','\ ',cell2mat(numNam(i))])};
    temp=cell2mat(numNam(i));
    TRAIN(2,i)={temp(1)};
end
for i=numLen+1:len
    TRAIN(1,i)={imread(['alphabets','\ ',cell2mat(alphNam(i-numLen))])};
    temp=cell2mat(alphNam(i-numLen));
    TRAIN(2,i)={temp(1)};
end

save('TRAININGSET.mat','TRAIN');
```

	1	2	3	4	5	6	7	8	9	10	11
1	42x24 logic...	42x24 logic...	42x24 logic...	42x24 logic...	42x24 logic...	42x24 logic...	42x24 logic...	42x24 logic...	42x24 logic...	42x24 logic...	42x24 logic...
2	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	'پ'
3											
4											
5											
6											

اعضای نهایی داده TRAIN

جداسازی تصویر پلاک از کل خودرو:

```
picture=rgb2gray(picture2);
figure
subplot(1,2,1)
imshow(picture)

threshold = graythresh(picture);
picture = ~imbinarize(picture,threshold);
subplot(1,2,2)
imshow(picture)

picture=imresize(picture,[300 500]);
picture = bwareaopen(picture,30);
figure
imshow(picture)
[L,Ne]=bwlabel(picture);
```

ایده‌ی کلی این بخش این است که به ازای اجزای تصویر داده شده در دیتابیس کلی به دنبال آن جزء می‌گردیم و در انتها با توجه به چگالی حرف و اعداد در یک قسمت از تصویر، با در نظر گرفتن مقداری خطا، یک مستطیل را به عنوان پلاک جدا می‌کنیم. برای این کار ابتدا مانند پلاک انگلیسی تصویر را از حالت **rgb** به سیاه و سفید مطلق تبدیل کرده، اندازه‌ی آن را به 300 در 500 پیکسل تغییر می‌دهیم و در نهایت با بدست آوردن **threshold** آن، آن را به تصویری باینری (هر پیکسل سیاه یا سفید) تبدیل می‌کنیم. در ادامه توسط دستور **bwareaopen** نقاط سفیدی از تصویر که اندازه‌شان کمتر از 30 پیکسل باشد را سیاه می‌کنیم.



نمونه خاکستری و سپس سیاه و سفید مطلق شده

سپس دیتا مستطیل های بدست آمده را با دیتابیس مقایسه میکنیم و و در صورتی که این مقدار correlation از مقدار مشخص 0.45 بیشتر بود، مختصات X و Y آنرا در آرایه **valids** ذخیره میکنیم. برای بدست آوردن مختصات دقیق تر پلاک، به مینیمم و ماکسیمم X و Y احتیاج داریم. برای بدست آوردن این مقادیر در ابتدا داده هارا سورت میکنیم و میانه هر ردیف که مختصات X و Y ها هستند را بدست می آوریم. سپس اختلاف هر یک از اعضای آرایه **valids** را با میانه بدست می آوریم و یک عدد را به عنوان عدد مرزی بدست می آوریم تا با مقایسه اختلاف هر داده با میانه بتوانیم داده های پرت را حذف کنیم. در نهایت از داده های نهایی مقدار مینیمم و ماکسیمم X و Y را بدست می آوریم و طبق آن مختصات تقریبی پلاک بدست می آید.

```
count = 0;
valids = zeros(4, total);
for n=1:Ne
    [r,c] = find(I==n);
    Y=picture(min(r):max(r),min(c):max(c));
    imshow(Y)
    Y=imresize(Y,[42,24]);
    ro=zeros(1,total);
    for k=1:total
        ro(k)=corr2(TRAIN{1,k},Y);
    end
    [MAXRO,pos]=max(ro);
    if MAXRO>.45
        valids(1, count + 1) = min(r);
        valids(2, count + 1) = max(r);
        valids(3, count + 1) = min(c);
        valids(4, count + 1) = max(c);
        out=cell2mat(TRAIN{2,pos});
        count = count + 1;
    end
end
```

مقایسه دیتاها با دیتابیس و ذخیره مختصات آنها

```
[~, iy1] = sort(valids(1,:));
minRowSorted = valids(:,iy1);
midIndex = fix((total - count + 1 + total)/ 2);
midMinRow = minRowSorted(1, midIndex);

[~, iy2] = sort(valids(2,:));
maxRowSorted = valids(:,iy2);
midMaxRow = maxRowSorted(2, midIndex);

midMinCol = valids(3, fix(( count + 1 )/ 2));

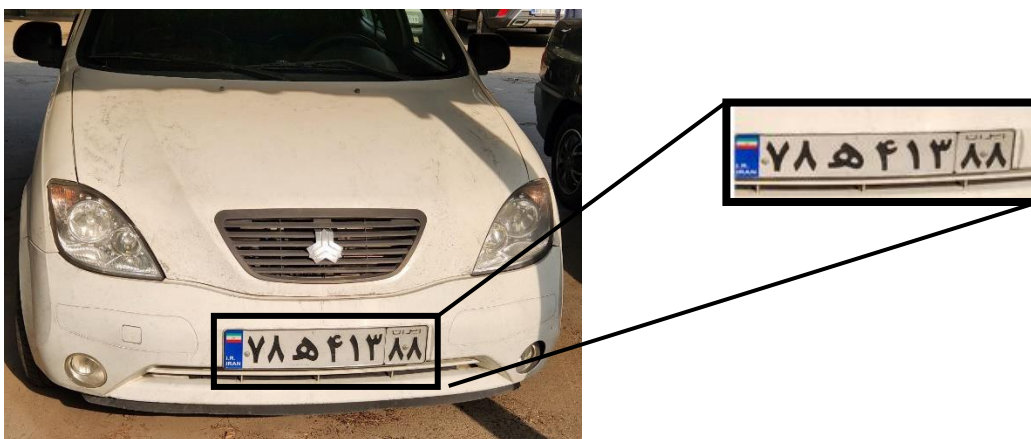
[~, ix] = sort(valids(4,:));
maxColSorted = valids(:,ix);
midMaxCol = maxColSorted(4, midIndex);
```

```

maxC = 0;
minC = 10000;
minR = 10000;
maxR = 0;
for n=1:count
    if (abs(valids(1,n) - midMinRow) < 20 && abs(valids(2,n) - midMaxRow) < 20 ...
        && abs(valids(3,n) - midMinCol) < 200 && abs(valids(4,n) - midMaxCol) < 300)
        minR = min(minR, valids(1, n));
        maxR = max(maxR, valids(2,n));
        minC = min(minC, valids(3,n));
        maxC = max(maxC, valids(4,n));
    end
end
end

```

حذف داده های پرت و ذخیره کردن X و Y مینیمم و ماکسیمم



پلاک جدا شده از تصویر جلوبندی ماشین

پیدا کردن اعداد و حروف پلاک:

```

picture = imread('carPlaque.jpg');
figure
subplot(1,2,1)
imshow(picture)
picture=imresize(picture,[300 500]);
subplot(1,2,2)
imshow(picture)

```

آماده سازی پلاک جدا شده از تصویر: پس از جدا کردن پلاک از تصویر باید حروف و اعداد پلاک را بدست آورده و با مقایسه آنها با دیتابیس مقدار صحیح نهایی را به عنوان خروجی پلاک مورد نظر گزارش کرد. در ابتدای این قسمت با ران کردن برنامه، عکس پلاک بدست آمده به عنوان دیتا به برنامه داده شده است. اندازه‌ی تصویر را به 300 در 500 پیکسل تغییر می‌دهیم.



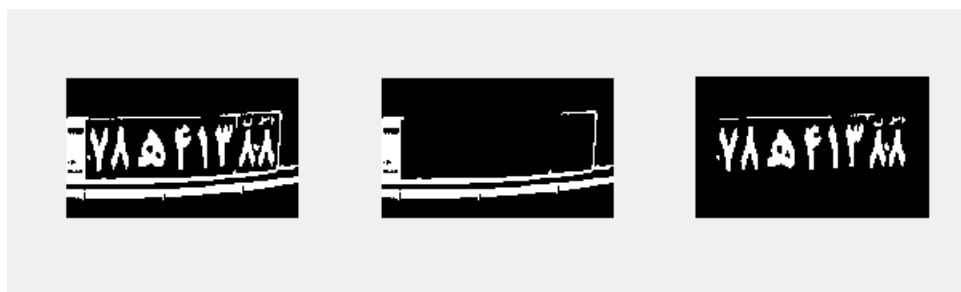
تبدیل تصویر به bitmap و واضح‌سازی آن:

```
picture=rgb2gray(picture);
threshold = graythresh(picture);
picture = bwareaopen(picture,50);
background=bwareaopen(picture,5000);|
picture2=picture-background;
subplot(1,3,3)
imshow(picture2)
```

مانند قسمت قبل، تصویر را از حالت **rgb** به سیاه و سفید مطلق تبدیل کرده و در نهایت با بدست آوردن **threshold** آن، آن را به تصویری باینری (هر پیکسل سیاه یا سفید) تبدیل می‌کنیم. در ادامه توسط دستور **bwareaopen** نقاط سفیدی از تصویر که اندازه‌شان کمتر از 50 پیکسل باشد را سیاه می‌کنیم. سپس برای جدا کردن قاب پلاک ابتدا قسمت‌هایی از تصویر که اندازه‌شان کوچکتر از 5000 است (تمامی اجزا جز قاب) جدا می‌کنیم و نتیجه را در **background** می‌ریزیم و سپس آن را از کل تصویر کم می‌کنیم تا تصویر پلاک بدون قاب بدست آید.



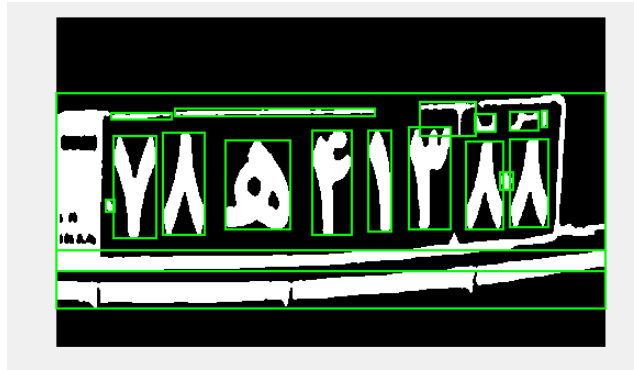
نمونه خاکستری و سپس سیاه و سفید مطلق



نمونه حذف دور پلاک

لیبل زدن به اجزای به هم پیوسته: در این قسمت با استفاده از تابع **bwlabel** اجزای متصل در تصویر باینری دو بعدی را لیبل می‌زنیم. با استفاده از خروجی‌های این تابع که شامل تعداد و اجزای دیتای لیبل زده شده است و همچنین استفاده از دستور **rectangle** داده‌های انتخابی را مشخص می‌کنیم. (این بخش صرفاً برای دیدن ما و اطمینان از صحت کارکرد اضافه شده است.)

```
figure
imshow(picture)
[L,Ne]=bwlabel(picture);
propied=regionprops(L,'BoundingBox');
hold on
for n=1:size(propied,1)
    rectangle('Position',propied(n).BoundingBox,'EdgeColor','g','LineWidth',1)
end
hold off
```

نمونه اجزا پیدا شده در یک پلاک

تصمیم‌گیری در مورد اجزای پلاک:

```
count = 0;
n = 1;
upY = 0;
previousXend = 0;
while n <= Ne
    if count == 8
        break;
    end
    [r,c] = find(I==n);
    if (max(r)-min(r))* (max(c)-min(c)) < 200 || max(r) < upY || min(c) < previousXend
        n = n + 1;
        continue;
    end
```

در این بخش به ازای هر جزء پیدا شده در قسمت قبل، شماره تمام ردیف‌ها و ستون‌هایی که پیکسل‌های این جزء در آن واقع شده‌اند در متغیر r و c ذخیره می‌شوند. برای حذف داده‌های پرت از مجموعه داده هامن، چند شرط را در ابتدا بررسی می‌کنیم:

1. مساحت اجزای بدست آمده از مقدار مشخصی کمتر نباشد.

2. در نظر نگرفتن قسمت "ایران" پلاک که برای اینکار y بالای دومین رقم از سمت راست پلاک را در متغیر upY ذخیره می‌کنیم و در صورتی که y جزهای پس از آن بالای upY بود، آن را در نظر نمی‌گیریم.

3. x ابتدایی هر جز حتما از x انتهایی جزء قبلی آن بیشتر باشد، زیرا در غیر اینصورت نمیتواند عضوی از پلاک باشد.

```

if count == 2
[rNext,cNext] = find(L==n+1);
if min(cNext) < max(c) && max(cNext) < max(c)
    if min(rNext) > min(r)
        row = r;
    else
        row = rNext;
    end
if max(rNext) > max(r)
    row2 = rNext;
else
    row2 = r;
end
Y=picture2(min(row):max(row2),min(c):max(c));
Y=imresize(Y,[42,24]);
imshow(Y)
pause(10)
n = n + 1;
else
Y=picture2(min(r):max(r),min(c):max(c));
imshow(Y)
Y=imresize(Y,[42,24]);
imshow(Y)
pause(0.2)
end
total = totalAphs;
TRAIN = ALPHTRAIN;
else
Y=picture2(min(r):max(r),min(c):max(c));
total = totalNums;
TRAIN = NUMTRAIN;
end
end

```

با توجه به اینکه سومین جز پلاک، یک حرف است، بنابراین زمانی که counter که شمارش آن از 0 شروع میشود به 2 برسد، شرط هایی که در ادامه گفته خواهد شد را بررسی کرده و در دیتابیس alphabet به دنبال حروف میگردیم. در غیر اینصورت در دیتابیس numbers به دنبال اعداد می گردیم. از آنجایی که نقطه های حروف به عنوان جزئی جدا در نظر گرفته میشوند، وقتی به یک حرف میرسیم مختصات جز بعدی را ذخیره کرده و با مقایسه این مختصات با مختصات جز فعلی میتوان فهمید که جز بعدی نقطه است یا خیر. در صورت نقطه داشتن حرف، مختصات جدید حرف را بدست آورده و شمارنده را یکی جلو میبریم که دیگر نقاط حرف را محاسبه نکند. در غیر اینصورت مختصات جز فعلی را بدست می آوریم. چون که نقاط حروف میتوانند بالا، وسط و پایین حرف باشند، باید شروطی را برای مختصات نقاط بررسی کنیم که این شرط ها عبارتند از: 1. اگر مینیمم X جز بعدی از مینیمم X جز فعلی بیشتر باشد، به معنای آن است که نقطه حرف بالای آن است، بنابر این مینیمم X جز فعلی را در نظر میگیریم، در غیر اینصورت مینیمم X جز بعدی را در نظر میگیریم. 2. اگر ماکسیمم X جز بعدی از ماکسیمم X جز فعلی بیشتر باشد، به معنای آن است که نقطه حرف پایین آن است، بنابر این ماکسیمم X جز بعدی را در نظر میگیریم، در غیر اینصورت ماکسیمم X جز فعلی را در نظر میگیریم.

```

imshow(Y)
Y=imresize(Y,[42,24]);
imshow(Y)
pause(0.2)

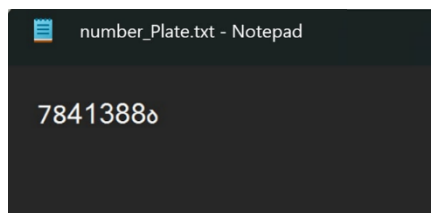
ro=zeros(1,total);
for k=1:total
    ro(k)=corr2(TRAIN{1,k},Y);
end
[MAXRO,pos]=max(ro);
if MAXRO>.45
    out=cell2mat(TRAIN(2,pos));
    final_output=[final_output out];
    count = count + 1;
    previousXend = max(c);
    if count == 7
        lastY = min(r);
    end
end
end

```

متغیر Y که برابر تصویر جز بررسی شده است را مطابق با اندازه تصویر دیتابیس (که در اینجا 42 در 24 پیکسل است) تغییر می دهیم. متغیر ro را به شکل یک آرایه 1 در تعداد اعضای دیتابیس تعریف می کنیم و اعضای آن را با 0 مقداردهی می کنیم. به ازای هر عضو دیتابیس، correlation آن را با تصویر Y پیدا کرده و در آرایه ی ro ذخیره می کنیم؛ حال ماکزیمم مقدار این آرایه را به همراه اندیس آن پیدا می کنیم و در صورتی که این مقدار ماکزیمم از یک مقدار مشخص (در اینجا 0.45) بیشتر بود، نام آن عضو را به متغیر final_output اضافه کرده و counter را یکی اضافه میکنیم.

در انتها یک فایل تکست باز کرده و مقدار نهایی متغیر `final_output` را در آن می‌ریزیم.

```
file = fopen('number_Plate.txt', 'wt');
fprintf(file, '%s\n', final_output);
fclose(file);
winopen('number_Plate.txt')
```



خروجی نشان داده شده برای پلاک



نمونه جزء تطبیق داده شده با دیتابیس

روش دیگر جداسازی پلاک از تصویر:

ایده‌ی کلی این است که مکان‌های احتمالی وجود پلاک را پیدا کنیم؛ برای اینکار تصویر پلاک را سیاه و سفید (طیف خاکستری) می‌کنیم و با توجه به تفاوت رنگ پیکسل‌های مجاور مکان احتمالی پلاک را تشخیص می‌دهیم؛ چون اعداد پلاک به رنگ سیاه و زمینه‌ی آن سفید است این تفاوت رنگ به طور واضحی در قسمت پلاک وجود دارد. پس از این عملیات با توجه به مکان قرارگیری پلاک در اکثر خودروها، پلاک را از بین گزینه‌های احتمالی تشخیص می‌دهیم.

پیدا کردن تفاوت رنگ پیکسل‌های هر سطر و ستون و فیلتر کردن آنها:

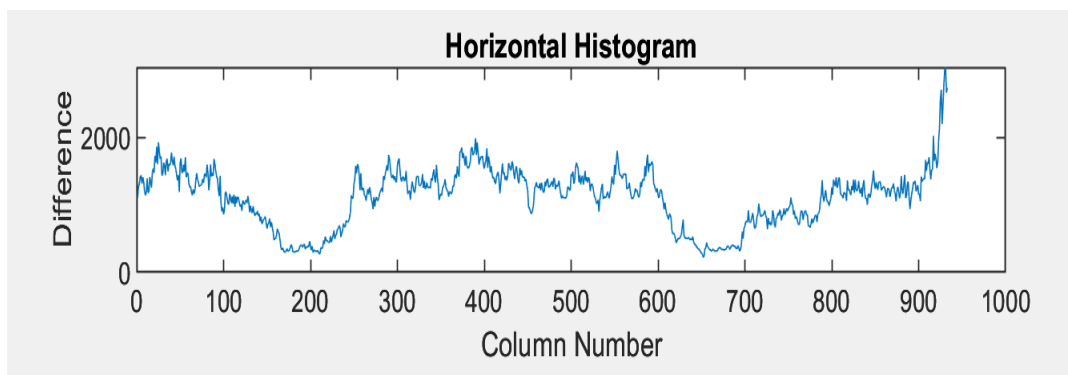
```
rgbImage = image;
grayPic = rgb2gray(image);
[rowSize, colSize] = size(grayPic);

image = grayPic;

colDifSum = 0;
AllColsDifSum = 0;
for i = 1:colSize
    colDifSum = 0;
    for j = 2:rowSize
        if abs(uint32(image(j-1, i)) - image(j, i)) > 20
            colDifSum = colDifSum + abs(uint32(image(j-1, i)) - image(j, i));
        end
    end
    col(i) = colDifSum;
    AllColsDifSum = AllColsDifSum + colDifSum;
end
colDifavg = AllColsDifSum / colSize;
```

ابتدا تصویر را خاکستری می‌کنیم و تعداد ستون و ردیف‌های آن را ذخیره می‌کنیم. همچنین نمونه‌ی رنگی تصویر را در متغیر `rgbImage` نگهداری می‌کنیم. حالا به ازای هر ستون موجود در تصویر اختلاف رنگ هر دو عضو بالایی و پایینی را پیدا می‌کنیم و در صورتی که این مقدار از مقدار قابل توجهی (در اینجا 20) بیشتر بود، آن را با عددی که نشان‌دهنده‌ی مجموع اختلاف‌ها در آن ستون است، جمع می‌کنیم. در نهایت مقادیر بدست آمده برای هر ستون را باهم جمع می‌کنیم و بر تعداد ستون‌ها تقسیم می‌کنیم تا میانگین مجموع اختلاف‌های پیکسل‌ها برای هر ستون بدست آید.

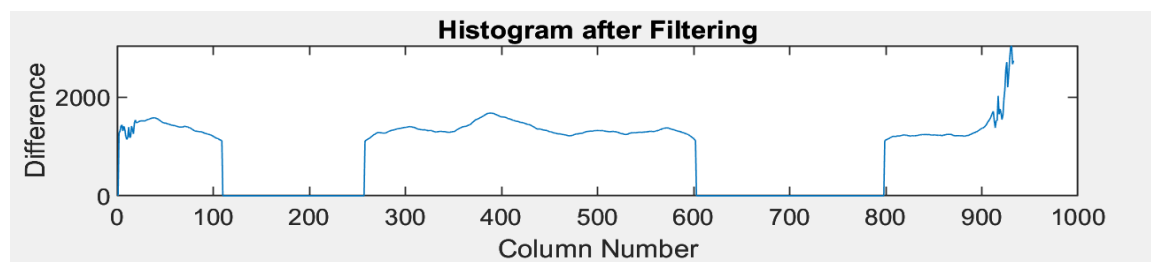
با رسم نموداری که مقدار تفاوت موجود در هر ستون را بر اساس شماره ستون نمایش می‌دهد، متوجه می‌شویم که مقادیر ما مقداری نویز دارد. این نمودار در زیر نمایش داده شده است.



نمودار مقدار اختلاف در ستون‌ها بر اساس شماره ستون

```
colSum = 0;
difCol = col;
for i = 21:(colSize-21)
    colSum = 0;
    for j = (i-20):(i+20)
        colSum = colSum + col(j);
    end
    difCol(i) = colSum / 41;
end
```

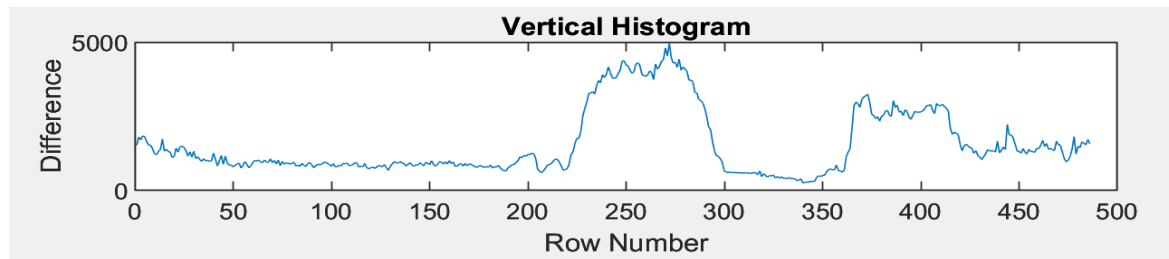
برای رفع نویزها و یکپارچه‌تر کردن نمودار، برای هر ستون میانگین اختلاف ستون‌ها را تنها برای ستون‌های مجاورش (20 ستون سمت چپ و 20 ستون سمت راست) محاسبه می‌کنیم و سپس نمودار را بر حسب متغیر difCol رسم می‌کنیم.



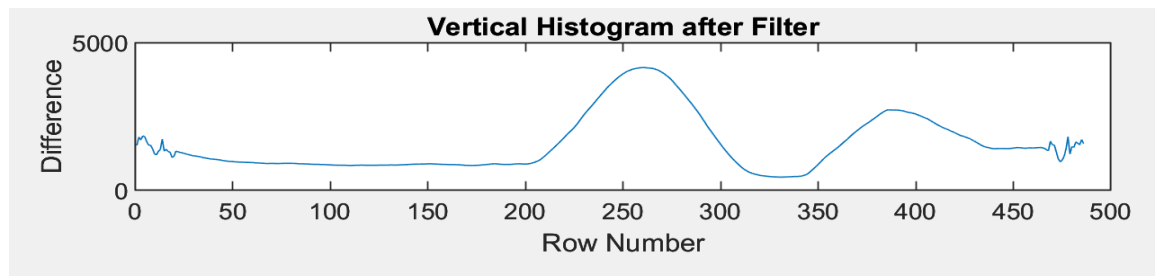
نمودار مقدار اختلاف در ستون‌ها بر اساس شماره ستون پس از اعمال فیلتر

همانطور که مشاهده می‌شود، مقدار اختلاف در ستون‌های میانی یعنی در وسط تصویر از نظر افقی زیاد است که این مکان، همان مکان قرارگیری پلاک است.

عملیات گفته شده در عینا برای سطرهای تصویر اجری می‌کنیم که نمودار های آن به صورت زیر خواهد بود.



نمودار مقدار اختلاف در سطرها بر اساس شماره سطر



نمودار مقدار اختلاف در سطرها بر اساس شماره سطر پس از اعمال فیلتر

حذف قسمتهایی که احتمال وجود پلاک در آنها نیست:

```
for i = 1:colSize
    if(difCol(i) < colDifavg)
        difCol(i) = 0;
        for j = 1:rowSize
            image(j, i) = 0;
            rgbImage(j, i) = 255;
        end
    end
end
```

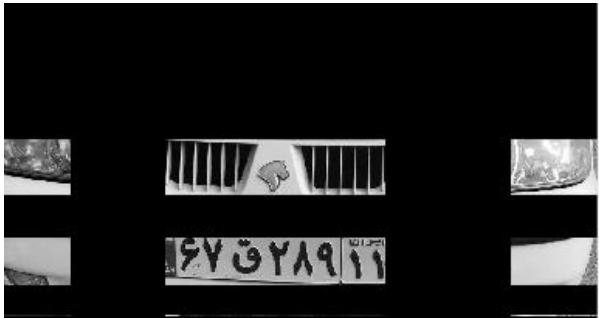
در این قسمت تمامی ستونها و ردیفهایی که مقدار اختلاف آنها کمتر از میانگین است را در تصویر سیاه و سفید درآورده و در تصویر رنگی به آن مقدار 255 که نشان دهنده حداکثر رنگ قرمز است می دهیم (دلیل این کار در انتها ذکر می شود). کد روبهرو مربوط به حذف ستونها است.



تصویر اولیه خاکستری



تصویر پس از حذف ستونها



تصویر پس از حذف سطرها و ستونها



تصویر رنگی پس از حذف ستونها و سطرها

تشخیص پلاک از بین گزینه‌های احتمالی:

```
for i = 1:rowSize
    for j = 1:colSize
        if (i < rdivide((2*rowSize),3) || j < rdivide(colSize,4) || j > rdivide(3*colSize,4)) && image(i,j) ~= 0
            image(i,j) = 0;
            rgbImage(i,j) = 255;
        end
    end
end
```

برای اینکار تمامی پیکسل‌هایی که در $2/3$ بالایی تصویر و $1/4$ راست و $1/4$ چپ تصویر هستند را حذف می‌کنیم؛ با اینکار قسمت‌هایی که از تصویر که در محدوده‌ی پلاک نیستند حذف می‌شوند. البته مشکل این روش این است که ممکن است موقعیت پلاک بالاتر از $1/3$ پایین تصویر باشد.



تصویر خاکستری سیاه کردن تمامی قسمت‌ها به جز پلاک



تصویر rgb قرمز کردن تمامی قسمت‌ها به جز پلاک

```

R = rgbImage(:,:,1);
G = rgbImage(:,:,2);
B = rgbImage(:,:,3);
mask = R ~= 255;
R(~mask) = 255;
G(~mask) = 255;
B(~mask) = 255;
J = cat(3,R,G,B);
imshow(J);
imwrite(J, "carPlaque.jpg");

```

حال برای جدا کردن پلاک از تصویر رنگی، تمامی قسمت‌هایی که بخش Red آنها عددی به جز 255 است را حذف می‌کنیم و تصویر باقی‌مانده را در carPlaque.jpg ذخیره می‌کنیم.

Figure 1-10



تصویر نهایی پلاک جداشده