



تمرین کامپیوتری دوم

سیگنال ها و سیستم ها



مجید صادقی نژاد - 810101459

آرمان خورشیدی - 810101417

۱۴۰۳ پاییز

استاد: دکتر اخوان



بخش اول

• سوال اول

این کار به سادگی با قطعه کد زیر قابل انجام است که فقط فایل های png و jpg را می پذیرد (می توانید برای بخش اول از تصاویر موجود در m1 استفاده کنید)

```
clear;clc;close all;

[file,path] = uigetfile({'*.jpg;*.png'}, 'Select your image');
image = imread([path file]);
image = imresize(image, [300 500]);
```

• سوال دوم

همانطور که در تصویر با مشاهده می شود این کار نیز با یک دستور ساده‌ی متلب پیاده سازی می شود

• سوال سوم

تابع مورد انتظار به شکل زیر و دقیقاً طبق خواسته‌ی صورت سوال ساخته می شود.

```
function gray_image = mygrayfun(image)
    gray_image = 0.299 * image(:,:,1) + 0.578 * image(:,:,2) + 0.114 * image(:,:,3);
end
```

• سوال چهارم

پیاده سازی تابع خواسته شده‌ی صورت سوال بسیار ساده و به شکل زیر می باشد برای مقدار threshold دلخواه نیز در کد ما مقدار ۱۰۰ انتخاب شده است

```

function binary_image = mybinaryfun(image, threshold)
    binary_image = image <= threshold ;
end

clear;clc;close all;

[file,path] = uigetfile({'*.jpg;*.png'},'Select your image');
image = imread([path file]);
image = imresize(image, [300 500]);

figure;
subplot(2,2,1)
imshow(image)
title('Base')

subplot(2,2,2)
image = mygrayfun(image);
imshow(image)
title('Gray')

subplot(2,2,3)
image = mybinaryfun(image, 100);
imshow(image)
title('Binary')

```

• سوال پنجم

برای پیاده سازی این سوال ابتدا به تابع سوال ششم نیاز داریم تا بتوانیم تصویر را بخش بندی بکنیم اما پس از بخش بندی تصویر با اعداد مختلف ۱ تا k به راحتی می توان بررسی کرد که آیا تعداد یک عدد خاص در تصویر بخش بندی شده ، کمتر از مقدار آرگومان n است یا خیر و اگر بود تمامی وجود های آن عدد در ماتریکس تصویر را تبدیل به صفر می کنیم بنابراین بخش اصلی این سوال مربوط به سوال بعد است.

```

function cleared_image = myremovecom(image, n)

[L, num] = mysegmentation(image);
cleared_image = false(size(image));

for label = 1:num
    component = (L == label);
    if sum(component(:)) >= n
        cleared_image = cleared_image | component;
    end
end

end

```

• سوال ششم

برای تقسیم بندی یک تصویر باینری که به صورت 0 و 1 در یک ماتریکس ذخیره شده کافیست تا بررسی کنیم هر پیکسل اگر مقدار یک دارد 8 پیکسل اطرافش را از لحاظ 1 بودن چک کنیم و همین طور برای هشت پیکسل اطراف هر کدام از آن 8 پیکسل و ... تا هنگامی که به جایی برسیم یا تمامی خانه های اطراف برابر صفر هستند یا قبل از شماره گذاری شده اند (برای نگهداری پیکسل های بهم متصل از استک استفاده می کنیم) سپس به تمام آن پیکسل ها یک شماره ی طبیعی نسبت می دهیم.

```
function [L, n] = mysegmentation(image)
    [rows, cols] = size(image);
    L = zeros(rows, cols);
    current_label = 0;

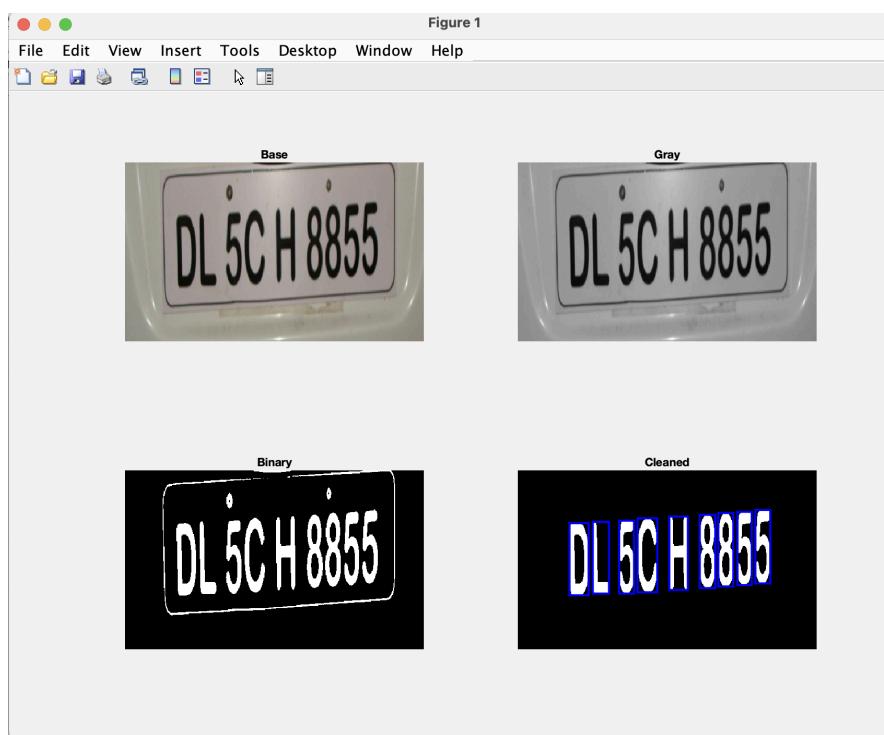
    directions = [-1, 0; 1, 0; 0, -1; 0, 1; -1, -1; -1, 1; 1, -1; 1, 1];
    for r = 1:rows
        for c = 1:cols
            if image(r, c) == 1 && L(r, c) == 0
                current_label = current_label + 1;
                stack = [r, c];
                while ~isempty(stack)
                    pixel = stack(end, :);
                    stack(end, :) = [];
                    r_pixel = pixel(1);
                    c_pixel = pixel(2);

                    if r_pixel < 1 || r_pixel > rows || c_pixel < 1 || c_pixel > cols || L(r_pixel, c_pixel) ~= 0 || image(r_pixel, c_pixel) == 0
                        continue;
                    end

                    L(r_pixel, c_pixel) = current_label;

                    for i = 1:size(directions, 1)
                        new_r = r_pixel + directions(i, 1);
                        new_c = c_pixel + directions(i, 2);
                        stack = [stack; new_r, new_c];
                    end
                end
            end
        end
    end
    n = current_label;
end
```

حال می توان تمامی مراحل طی شده را در تصویر زیر بررسی کرد:



```

clear;clc;close all;

[file,path] = uigetfile({'*.jpg;*.png'},'Select your image');
image = imread([path file]);
image = imresize(image, [300 500]);

figure;
subplot(2,2,1)
imshow(image)
title('Base')

subplot(2,2,2)
image = mygrayfun(image);
imshow(image)
title('Gray')

subplot(2,2,3)
image = mybinaryfun(image, 100);
imshow(image)
title('Binary')

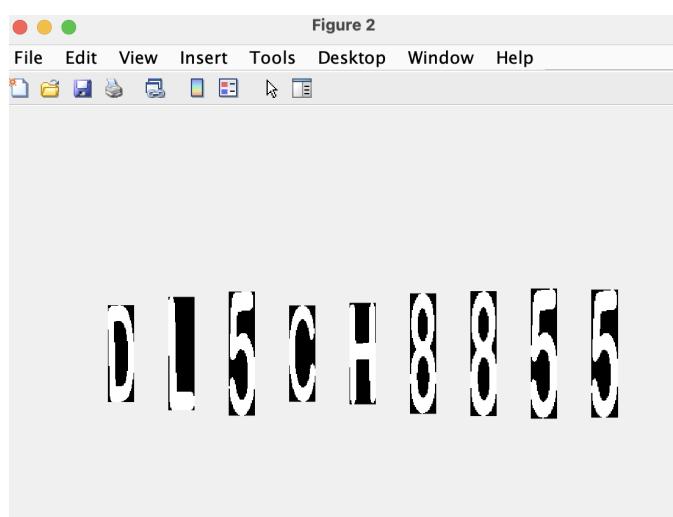
subplot(2,2,4)
image = myremovecom(image,300)-myremovecom(image,2300);
imshow(image)
title('Cleaned')

[L,maxSegment] = mysegmentation(image);
maxSegment = maxSegment+1;
hold on

```

• سوال هفتم

حال بر اساس بخش بندی انجام شده در مرحله‌ی قبل به دور هر یک از کاراکتر‌های بخش بندی شده یک مستطیل می‌کشیم و آن را از عکس می‌بریم و همگی را به ترتیب در یکجا نگه می‌داریم و نمایش می‌دهیم (به صورت جدا از هم) سپس دیتابیس انگلیسی را به کمک سلول‌های متلب لود می‌کنیم که شامل تصاویر حروف و کاراکتر آن هاست و سپس با استفاده از حلقه و همچنین correlation گیری برای هر کاراکتر بریده شد تمام دیتابیس را چک می‌کنیم تا بینیم با کدام حرف یا عدد تطبیق بیشتری دارد



```

[temp,indexs] = sort(label_positions(2,:));
label_positions(1,:) = label_positions(1,indexs);
label_positions(2,:) = label_positions(2,indexs);

files=dir('English_Set');
len=length(files)-2;
TRAIN=cell(2,len);
for i=1:len
    TRAIN{1,i}=imread(['English_Set','/',files(i+2).name]);
    TRAIN{2,i}=files(i+2).name(1);
end

figure
result = [];

for i = 1:maxSegment-1
    [r,c] = find(L == label_positions(1,i));
    character = image(min(r):max(r), min(c):max(c));
    subplot(1,maxSegment-1,i)
    imshow(character)
    decision = cell(2,1);
    decision{1,1} = 0;
    decision{2,1} = '';

    for j = 1:length(TRAIN)
        temp = corr2(imresize(TRAIN{1,j},size(character)), character);
        if temp >= 0.45 && temp >= decision{1,1}
            decision{1,1} = temp;
            decision{2,1} = TRAIN{2,j};
        end
    end
    result = [result decision{2,1}];
end

```

• سوال هشتم

و در نهایت نتیجه به دست آمده در بخش قبل را با استفاده از توابع مطلب در یک فایل می نویسیم
باید توجه کرد که روش correlation گیری بدون خطای نیست و گاهی کاراکتر های ۰ , ۵ را با ۰ اشتباه
می گیرد اما می توان با تعریف threshold برای این عملیات از خطای ها جلوگیری کرد

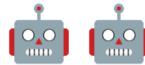
```

disp(result)
file = fopen('result.txt', 'wt');
fprintf(file, '%s\n', result);
fclose(file);

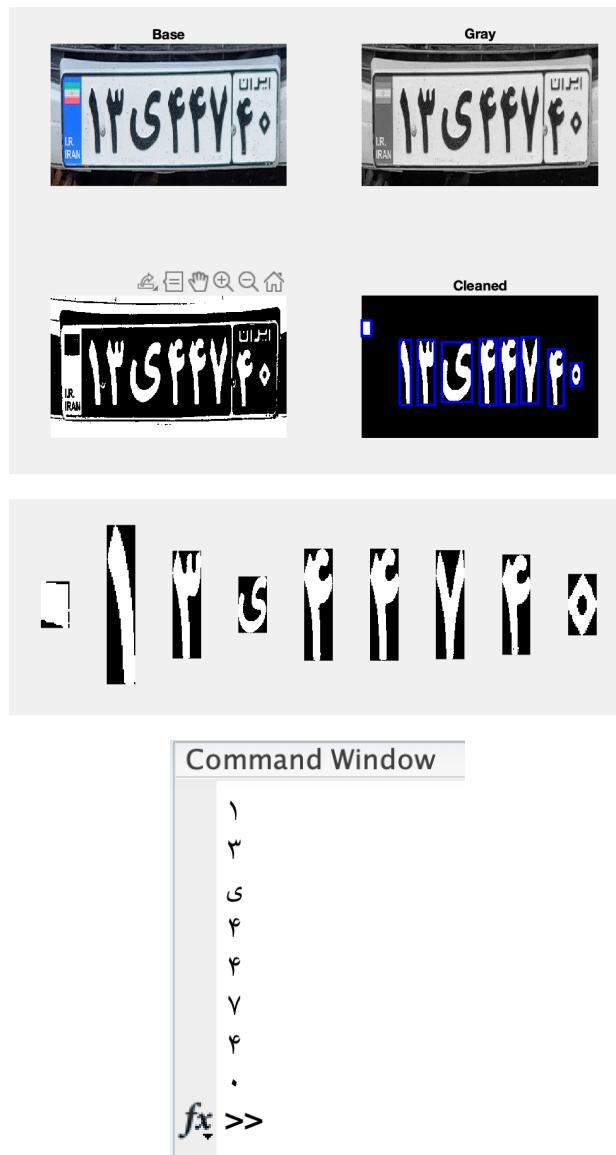
```

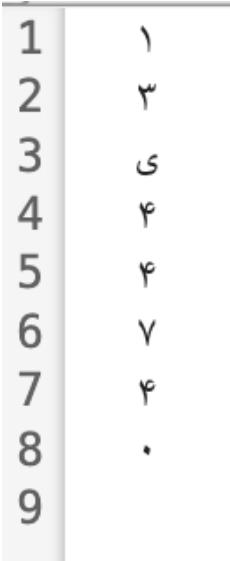


بخش دوم



در این بخش دقیقاً کد بخش اول را می‌نویسم با چند تفاوت جزئی برای مثال از تابع segmentation خود مطلب استفاده می‌کنیم زیرا عملکرد سریع تری دارد اما از تابع باینری ساز ساخته شده در بخش قبل استفاده می‌کنیم زیرا توانایی تنظیم دستی threshold را دارد به جای دیتابیس انگلیسی، دیتابیس فارسی را لود می‌کنیم (فایل دیتابیس شامل حروف و اعداد خواسته شده در پوششی تمرین کامپیووتری در دسترس است) و مقداری کمی (در حد ۱۰ عدد) مقادیر توابع bwareaopen و باینری ساز را تغییر می‌دهیم تا نتیجه مطلوب تر شود باقی کار دقیقاً مانند بخش قبل است. حتی در بخش خطای هم همانند بخش قبل ممکن است کاراکترهایی اضافه خواننده شوند که از آن جایی که با هیچ کدام از اعضای دیتابیس همپوشانی ندارند در نتیجه‌ی نهایی تاثیری ندارند (پوششی p2 مربوط به فایل‌های تستی است که می‌توان بر روی این فایل اجرا کرد در ذیل یک نمونه کارکرد کد این بخش آمده است)





```
clear;clc;close all;

[file,path] = uigetfile({'*.jpg;*.png'},'Select your image');
image = imread([path file]);
image = imresize(image, [300 500]);

figure;
subplot(2,2,1)
imshow(image)
title('Base')

subplot(2,2,2)
image = rgb2gray(image);
imshow(image)
title('Gray')

subplot(2,2,3)
image = imbinarize(image);
image = ~image;
imshow(image)
title('Binary')

subplot(2,2,4)
image = bwareaopen(image,400) - bwareaopen(image,5000);
imshow(image)
title('Cleaned')

[L,maxSegment] = bwlabel(image);
maxSegment = maxSegment+1;
hold on

label_positions = zeros(2,maxSegment-1);
for i = 1:maxSegment-1
    [r,c] = find(L == i);
    mc = min(c,[],'all');
    mr = min(r,[],'all');
    Mc = max(c,[],'all');
    Mr = max(r,[],'all');
    label_positions(1,i) = i;
    label_positions(2,i) = mc;
    rectangle('Position',[mc,mr,Mc-mc,Mr-mr], 'EdgeColor','b','LineWidth',2)
end
```

```

[temp,indexs] = sort(label_positions(2,:));
label_positions(1,:) = label_positions(1,indexs);
label_positions(2,:) = label_positions(2,indexs);

files=dir('Persian_Set');
len=length(files)-2;
TRAIN=cell(2,len);
for i=1:len
    TRAIN{1,i}=imread(['Persian_Set','/',files(i+2).name]);
    TRAIN{2,i}=files(i+2).name(1);
end

figure
result = [];
file = fopen('result.txt', 'wt');

for i = 1:maxSegment-1
    [r,c] = find(L == label_positions(1,i));
    character = image(min(r):max(r), min(c):max(c));
    subplot(1,maxSegment-1,i)
    imshow(character)
    decision = cell(2,1);
    decision{1,1} = 0;
    decision{2,1} = '';
    for j = 1:length(TRAIN)
        temp = corr2(imresize(TRAIN{1,j},size(character)), character);
        if temp >= 0.45 && temp >= decision{1,1}
            decision{1,1} = temp;
            decision{2,1} = TRAIN{2,j};
        end
    end
    disp(decision{2,1});
    fprintf(file, '%c\n',decision{2,1});
    result = [result decision{2,1}];
end

fclose(file);
|

```

بخش سوم

شیوه جدا سازی پلاک از جلو بندی به این صورت است که ابتدا تصویر به حالت gray در می آوریم و سپس آن را با یک threshold مناسب به طوری که اکثر اعداد پلاک از دست نزوند به صورت باینری در می آوریم حال با استفاده از تابع blabel کامپوننت های سفید تصویر باینری که شامل اعداد پلاک هم می شود را استخراج می کنیم و سپس با استفاده از تابع regionprops ، مختصات و ابعاد کامپوننت های یافت شده را پیدا می کنیم حال به فیلتر این کامپوننت ها می پردازیم در ابتدا کامپوننت هایی که به صورت افقی (طول از عرض بیشتر) هستند را حذف می کنیم زیرا می دانیم اعداد پلاک به صورت عمودی هستند سپس به دنبال حداقل سه کامپوننت عمودی می گردیم که به صورت افقی روی یک خط باشند (horizontally aligned) زیرا می دانیم اعداد پلاک چنین خاصیتی دارند در نهایت مختصات کامپوننت های پیدا شده که چنین خاصیتی دارند برایمان می ماند پس از مختصات آن ها مقداری به چپ و مقداری به راست می رویم و مختصات به دست آمده را از تصویر اصلی می برمی پس از این تصویر پلاک را استخراج کرده ایم و باقی کد دقیقا مطابق بخش قبل است. کد پیاده سازی این بخش به صورت زیر است:

```

[file, path] = uigetfile({'*.jpg;*.png'}, 'Select your image');
main_image = imread([path file]);
main_image = imresize(main_image, [600 1000]); % you should change these numbers for different picture kinds
image = rgb2gray(main_image);
image = mybinaryfun(image, 108);
image = bwareaopen(image, 200); % you should change these numbers for different picture kinds

figure
subplot(1,2,1)
imshow(main_image)
subplot(1,2,2)
imshow(image)
hold on

[labeled_img, num] = bwlabel(image);
stats = regionprops(labeled_img, 'BoundingBox', 'Area');

y_coords = arrayfun(@(x) x.BoundingBox(2), stats);
ly_coords_sorted, sortIndex = sort(y_coords);
sorted_stats = stats(sortIndex);
cords = {};

for k = 1:num
    bbox = sorted_stats(k).BoundingBox;
    if bbox(3) < bbox(4)*1.1
        rectangle('Position', bbox, 'EdgeColor', 'r', 'LineWidth', 2);
        cords{end+1} = bbox ;
    end
end
hold off

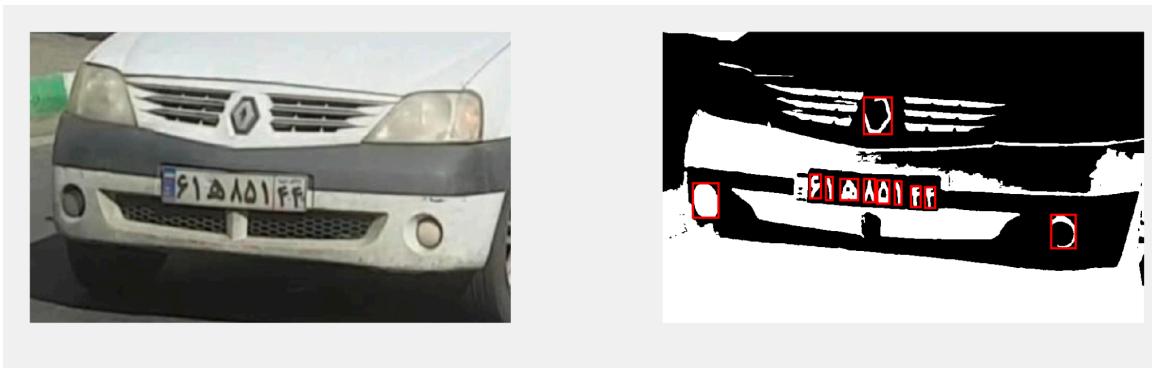
tresh = 10;

for i = 1:length(cords)-2
    if abs(cords{i}(2) -cords{i+1}(2)) < tresh && abs(cords{i}(2) -cords{i+2}(2)) < tresh && abs(cords{i+1}(2) -cords{i+2}(2)) < tresh
        plate_cord = [cords{i}(1)-cords{i}(3), cords{i}(2)-2*tresh,12.5*cords{i}(3),cords{i}(4)+4*tresh];
        break ;
    end
end

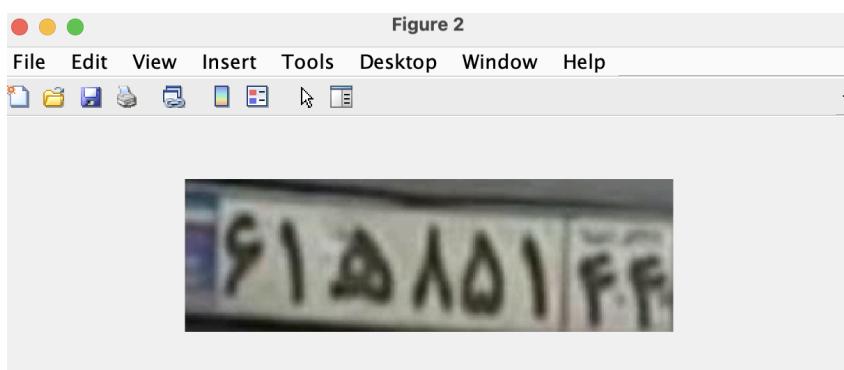
plate = imcrop(main_image,plate_cord);
figure
imshow(plate)

```

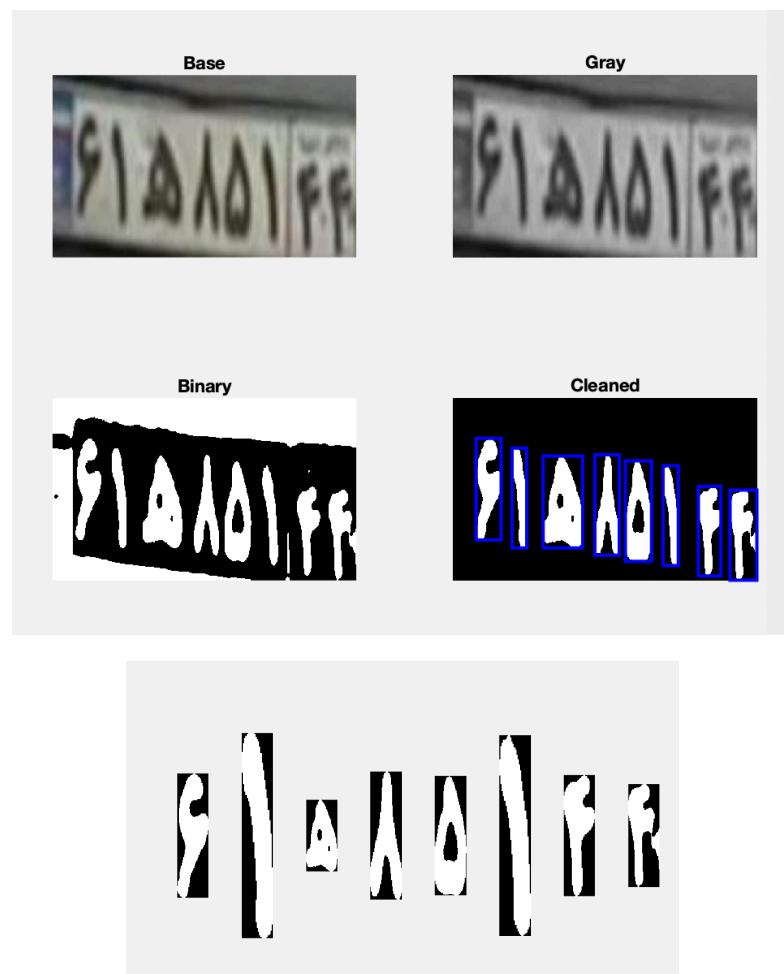
همچنین نتیجه به شکل زیر است:



تصویر خودرو باینری می شود و کامپوننت های عمودی استخراج می شوند سپس کامپوننت های که به صورت افقی در یک خط باشند شناسایی می شوند و محدوده‌ی اطراف آن ها (با مقداری منطقی حاشیه از چپ و راست به اندازه سه یا چهار کاراکتر عدد) از روی عکس اصلی می بریم
نتیجه‌ی نهایی به شکل زیر است:



و پس از آن مراحل استخراج پلاک آغاز می شود:

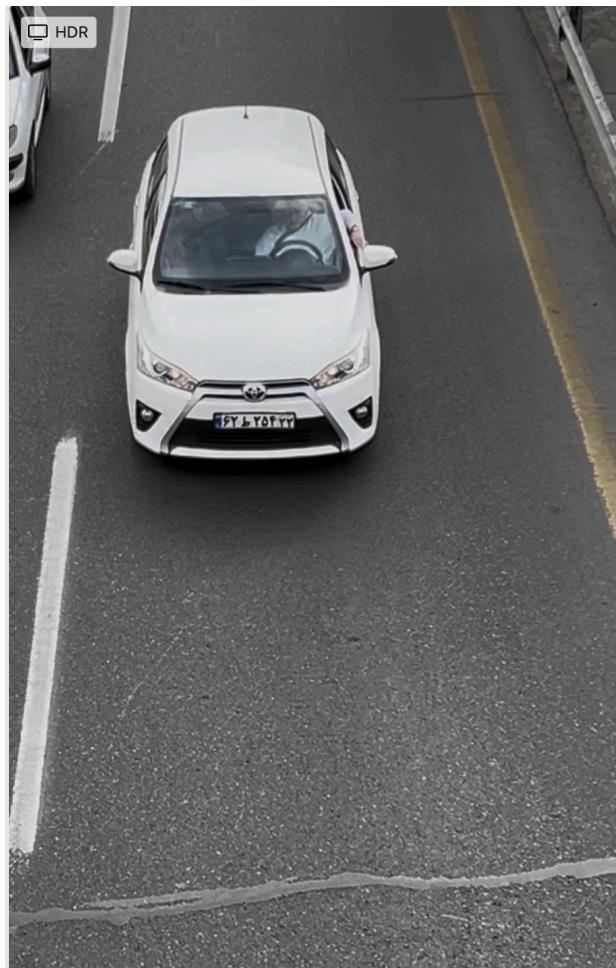


Command Window

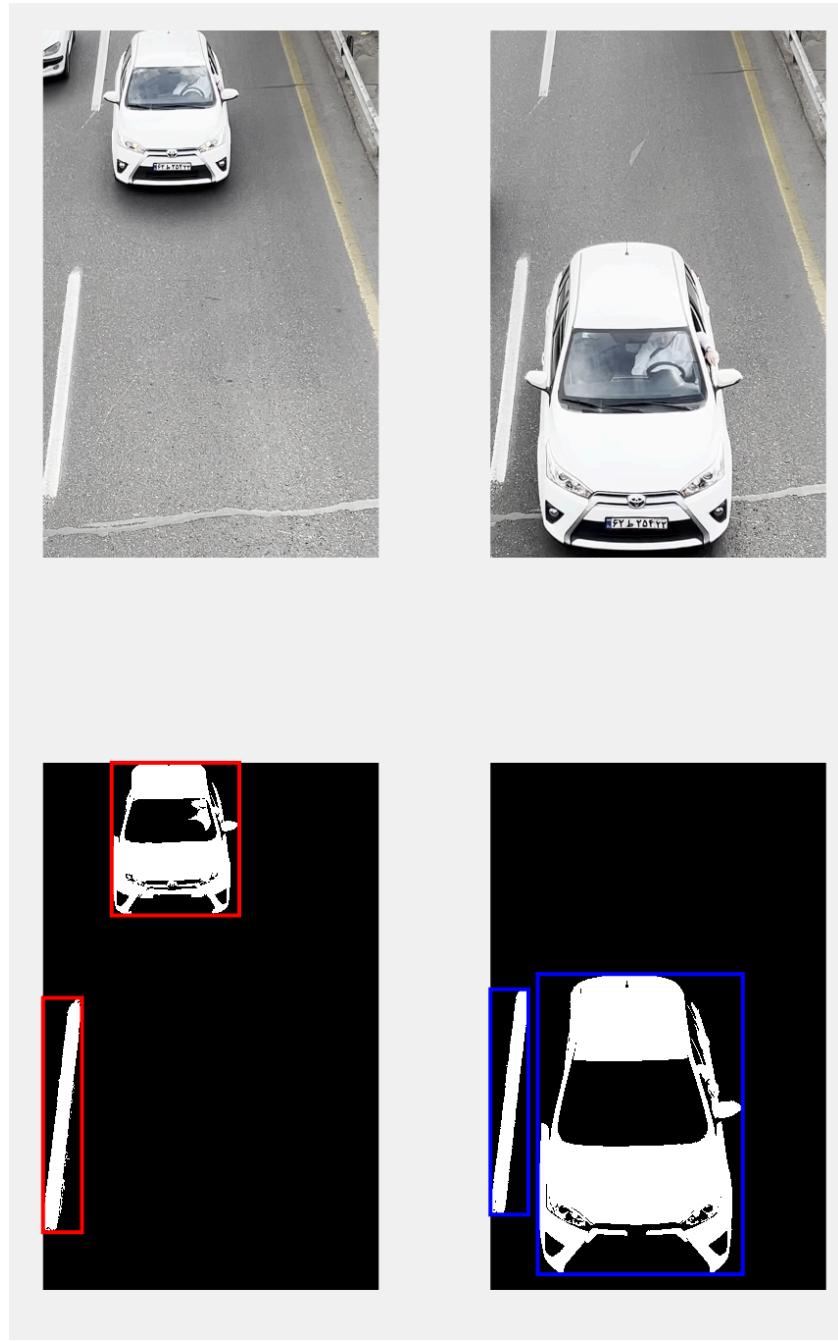
```
6  
1  
5  
8  
5  
1  
4  
4  
fx >>
```

بخش چهارم 😊😊

یک ویدیوی کوتاه از خودرو آپلود می‌کنیم فایل ویدیو با نام video.mov در پوشه‌ی پروژه موجود است و به شکل زیر می‌باشد.



حال برای پیدا کردن سرعت متوسط دو فریم از ویدیو استخراج می‌کنیم در هر دو فریم باید تصویر بدن خودرو کامل در تصویر باشد تا بتوان فاصله‌ی بین آن دو را حساب کنید از آن جا که fps ویدیو را می‌دانیم می‌توان زمان را نیز محاسبه کرد و خط سفید راهنمای روی آسفالت یک مقیاس است که اندازه‌ی آن به متر را می‌دانیم پس می‌توان با تشخیص بدن‌ی ماشین در دو فریم و محاسبه‌ی فاصله‌ی آن‌ها سرعت را به دست آورد برای این کار همانند بخش‌های قبل با استفاده از توابع آماده‌ی مطلب ابتدا تصویر را باینری کرده و بخش‌های زائد آن را حذف می‌کنیم سپس بدن‌ی خودرو و خط را در هر دو فریم تشخیص می‌دهیم که مختصات و ابعاد آن‌ها را بیابیم نتیجه به شکل زیر است:



حال با تعداد اندکی محاسبه می توان می سرعت متوسط خودرو بر حسب پیکسل بر ثانیه و سپس با استفاده از ابعاد مقیاس بر حسب متر بر ثانیه به دست آورد در نهایت متوجه می شویم سرعت متوسط خودرو برابر ۱۱ متر بر ثانیه است:

Command Window

speed is about 11 m/s !

همچنین کد این بخش به شکل زیر است:

```
clear; clc; close all;

select1 = 6;
select2 = 17;
vidObj = VideoReader('video.mov');
figure
subplot(2,2,1)
frame1 = read(vidObj, select1);
imshow(frame1)
subplot(2,2,2)
frame2 = read(vidObj, select2);
imshow(frame2)

image1 = rgb2gray(frame1);
image1 = mybinaryfun(image1,200);
image1 = ~image1 ;
image1 = bwareaopen(image1, 10000);
[L1, n1] = bwlabel(image1);
stats1 = regionprops(L1, 'BoundingBox', 'Area');

subplot(2,2,3)
imshow(image1)
hold on
for i = 1:n1
    bbox = stats1(i).BoundingBox;
    rectangle('Position', bbox, 'EdgeColor', 'r', 'LineWidth', 2);
end
hold off

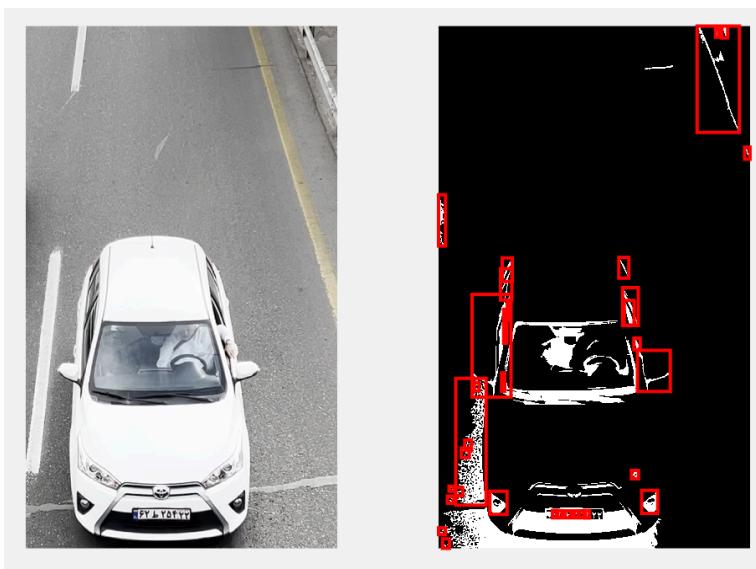
image2 = rgb2gray(frame2);
image2 = mybinaryfun(image2,200);
image2 = ~image2 ;
image2 = bwareaopen(image2, 10000);
[L2, n2] = bwlabel(image2);
stats2 = regionprops(L2, 'BoundingBox', 'Area');

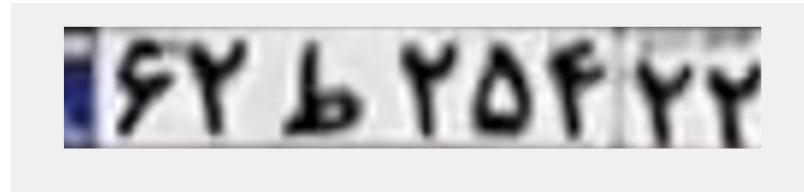
subplot(2,2,4)
imshow(image2)
hold on
for i = 1:n2
    bbox = stats2(i).BoundingBox;
    rectangle('Position', bbox, 'EdgeColor', 'b', 'LineWidth', 2);
end
hold off

scale = stats2(2).BoundingBox(4)/stats1(2).BoundingBox(4);
scale = scale -1;
measure_m_px = 3 / stats2(1).BoundingBox(4);
distance_px = ceil(stats2(2).BoundingBox(2) + scale*stats1(2).BoundingBox(4) - stats1(2).BoundingBox(2));
distance_m =ceil(distance_px * measure_m_px);
time_s = (select2-select1+1)* (1/vidObj.FrameRate);
speed_m_s = ceil(distance_m / time_s);

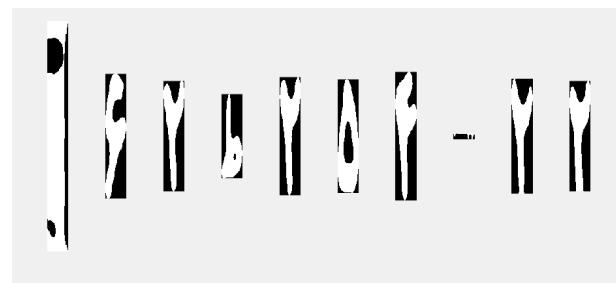
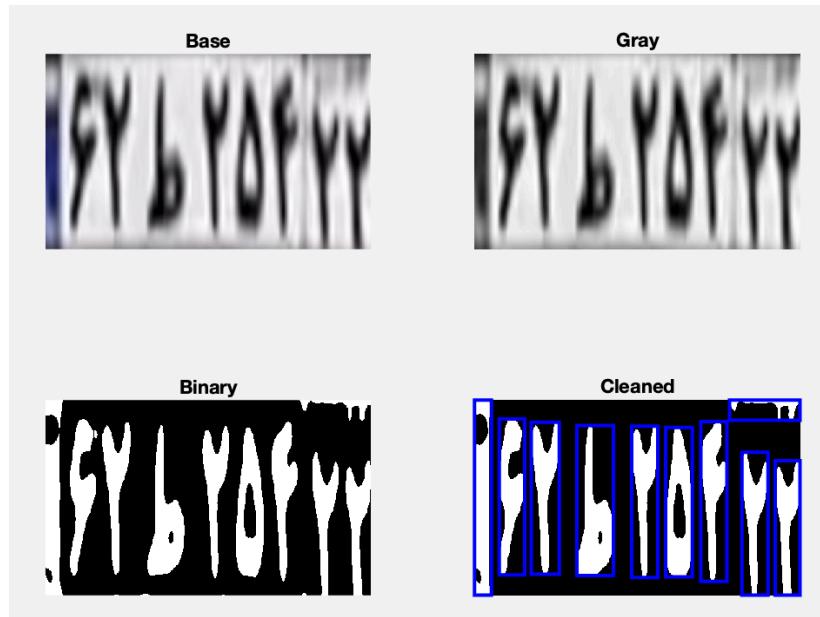
disp(['speed is about ', num2str(speed_m_s), ' m/s !'])
```

حال باقی کار دقیقاً مانند بخش های قبل است ابتدا با کدی که در بخش ۳ نوشته شد پلاک خودرو استخراج می شود:





سپس با استفاده از کد بخش ۲ اعداد پلاک تشخیص داده می شوند:



و در نهایت نتیجه‌ی نهایی به شکل زیر است:

```
Command Window
speed is about 11 m/s !
۱
۴
۲
ط
۲
۵
۴
۲
۲
fx >>
```