به نام خدا

گزارش تمرین کامپیوتری دوم

سیگنال ها و سیستم ها

دکتر اخوان

ناژین نیکخواه بهرامی

۸۱۰۱۰۲۵۳۰

اسفند۱۴۰۳

بخش اول)

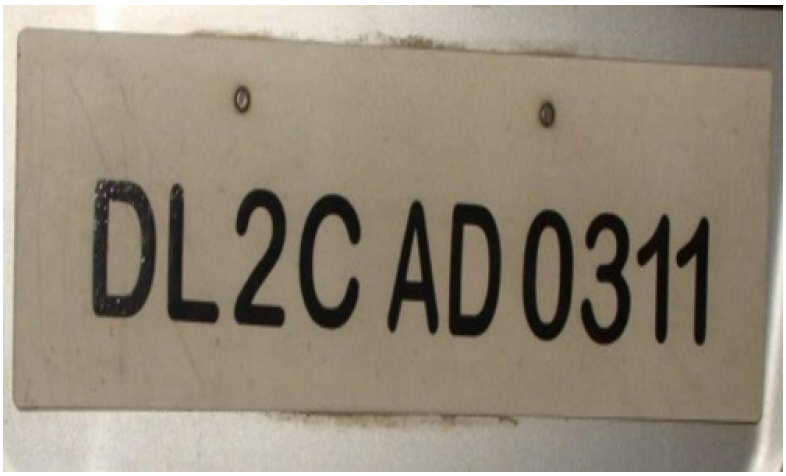
در این بخش قصد داریم با نوشتن تابع های مورد نیاز پلاک انگلیسی را جدا کرده و حروف و اعداد آنرا پیدا کنیم.

۱.ابتدا به کمک تابع آماده متلب فایل را باز کرده و میخوانیم تا بصورت یه ماتریس سه بعدی ذخیره شود.(در یک فایل ذخیره شده است. مثلا این عکس)

```
%% 1.1
[filename, filepath] = uigetfile({'*.jpg;*.png;*.bmp', 'Image Files (*.jpg, *.png,
*.bmp)'}, 'Select an Image');
imagepath = fullfile(filepath, filename);
image = imread(imagepath);
```

۲.برای اینکه بتوانیم در هر ابعادی که عکس ورودی داشته باشیم و آن را بررسی کنیم، سایز عکس را به ۳۰۰ در ۵۰۰ تغییر میدهیم.

```
%% 1.2
image = imresize(image, [300,500]);
figure;
imshow(image);
```



۳.به منظور کاهش پیچیدگی محاسبات و کاهش فضا به یک سوم آن با فرمول زیر بعد سوم را تغییر میدهیم و ذخیره میکنیم.

$$Gray_{channel} = 0.299 \times Red_{channel} + 0.578 \times Green_{channel} + 0.114 \times Blue_{channel}$$

```
%% 1.3
gray_image = mygrayfun(image);
figure;
imshow(gray_image);
```

تابع این قسمت به این صورت است:

```
function[grayscale_pic] = mygrayfun(pic)
    grayscale_pic = 0.299 .* pic(:,:,1) + 0.578 .* pic(:,:,2) + 0.114 .* pic(:,:,3);
end
```

درواقع بعد اول و دوم یعنی طول و عرض ثابت است ولی بعد سوم را تغییر مدهد.

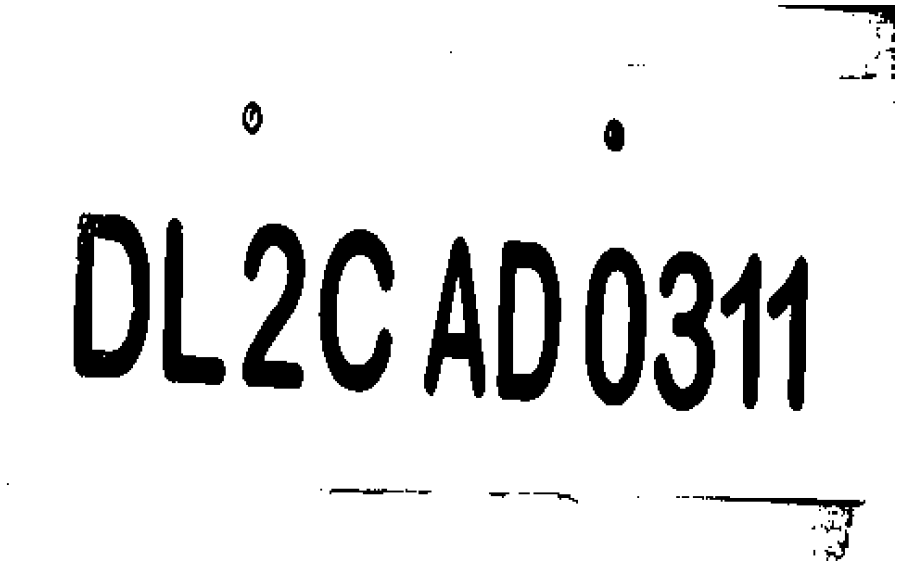
۴.در این قسمت میخواهیم که عکس را سیاه و سفید کنیم که چون عکس خاکستری شده است میتوان با ۰ و ۱ یعنی باینری این کار را انجام داد. برای اینکا میتوان یک ترشهولد گذاشت که بعد بررسی هر تیکه اگر بیشتر از این مقدار بود ۱ و اگر کمتر بود ۰ قرار دهد و با این روش سیه و یفید بسازیم. بعلت اینکه اعداد روی پلاک سیاه اند و بهتر میتوان کاراکتر هارا استخراج کرد از این روش استفاده میکنیم.

```
function[binary_pic] = mybinaryfun(pic, threshold)
    binary_pic = pic > threshold;
end
```

همانطور که مشخص است تابع عکس و میزان ترشهولد را ورودی گرفته و عکس سیاه و سفید شده را خروجی خواهد داد.(*توجه کنید برای تمام این قسمت ها احتیاج به فیگور سازی نیز داشتیم که خروجی عکس بدهد)

```
%% 1.4
binary_image = mybinaryfun(gray_image, 100);
figure;
imshow(binary_image);
```

در کل اعداد ترشهولد را باید باتوجه به قسمت مورد نیاز مثلا اینجا سیاه و سفید کردن یا قسمت های بعدی از بین بردن نویز، با تصحیح و خطا بدست اورد اینجا ۱۰۰ بهتر است.



۵.همانطور که معلوم است این عکس بدست آمده دارای نویز است یعنی مثلا بین چندین عدد ۱۰۱ و ۹۹ و ۹۸ و ... ممکن است خطایی کم باید که در اثر قسمت های دیگر ماشین بدست آمده است. بنابراین باید اطرافش را بررسی و این قسمت هارا حذف کنیم که باعث خطا در تضخیص

کاراکتر نشوند. برای این کار ابتدا باید قسمت به قسمت جلو برویم و ببینیم کجا سیاه است یعنی

۱و برمیگردد اما در غیر این صورت -۱ برمیگرداند که معلوم شود این قسمت مخالف سیاه است. از

این تابع در قسمت بعد استفاده میکنیم.

```
function[i,j] = find_any(matrix)
    for j = 1:size(matrix,2)
        for i = 1:size(matrix,1)
            if(matrix(i,j) == 1)
                return;
            end
        end
    end
    i = -1;
    j = -1;
end
```

تابع های بعدی عکس را میگیرد و دوباره یک ترشهولد نیاز دارد که بتواند پاکسازی کند. یعنی تیکه به تیکه های عکس که از قسمت قبلی گرفته را بررسی میکند و میگردد(همه طرف) که این قسمت سیاه که جدا کردیم آیا به اندازه کافی بزرگ هست که کاراکتر باشد و نگهش داریم یا باید پاک شود. برای این کار باید بریم و اطراف یه تیکه ۱ را بررسی کنیم و دخیره کنیم(بعنوان یک پارت نگهش داریم) و در غیر این ۰ کنیم.

```
function[parts] = detect_parts(pic, threshold)
    height = size(pic,1);
    width = size(pic,2);
    parts = {};
    queue = [];
    pairs = [];
    while(true)
        if(isempty(queue))
            if(size(pairs,1) >= threshold)
                parts{end+1} = pairs;
            end
            pairs = [];
            [i,j] = find_any(pic);
            if(i == -1)
                break;
            end
            queue = [i,j];
            pic(i,j) = 0;
        else
            row = queue(1, 1);
            col = queue(1, 2);
            pairs = [pairs; queue(1, :)];
```

```
            queue(1, :) = [];
            if(row > 1 && pic(row - 1, col) == 1)
                pic(row - 1, col) = 0;
                queue = [queue;[row - 1, col]];
            end
            if(row < height && pic(row + 1, col) == 1)
                pic(row + 1, col) = 0;
                queue = [queue;[row + 1, col]];
            end
            if(col > 1 && pic(row, col - 1) == 1)
                pic(row, col - 1) = 0;
                queue = [queue;[row, col - 1]];
            end
            if(col < width && pic(row, col + 1) == 1)
                pic(row, col + 1) = 0;
                queue = [queue;[row, col + 1]];
            end
            if(row > 1 && col > 1 && pic(row - 1, col - 1) == 1)
                pic(row - 1, col - 1) = 0;
                queue = [queue;[row - 1, col - 1]];
            end
            if(row < height && col > 1 && pic(row + 1, col - 1) == 1)
                pic(row + 1, col - 1) = 0;
                queue = [queue;[row + 1, col - 1]];
            end
            if(row > 1 && col < width && pic(row - 1, col + 1) == 1)
                pic(row - 1, col + 1) = 0;
                queue = [queue;[row - 1, col + 1]];
            end
            if(row < height && col < width && pic(row + 1, col + 1) == 1)
                pic(row + 1, col + 1) = 0;
                queue = [queue;[row + 1, col + 1]];
            end

        end

    end

    pairs = [pairs;queue];
    if(size(pairs,1) > threshold)
        parts{end+1} = pairs;
    end
end




function[filtered_pic] = myremovecom(pic, threshold)
    parts = detect_parts(pic, threshold);
    filtered_pic = zeros(size(pic));
    for i = 1:numel(parts)
        part = parts{i};
        for j = 1:size(part,1)
            filtered_pic(part(j,1),part(j,2)) = 1;
        end
    end
end
```

```
%% 1.5
filtered = myremovecom(~binary_image, 300);
background = myremovecom(~binary_image, 2300);
characters = (filtered - background);
figure;
subplot(1,3,1);
imshow(~filtered);
title('Without noise');
subplot(1,3,2);
imshow(~background);
title('Background');
subplot(1,3,3);
imshow(~characters);
title('characters');
```

در این تابع باید تیکه های بهم چسبیده که گفته شده است را بررسی کند و اگر به اندازه پیوستگی یک کاراکتر نباشد باز هم احتیاج داریم که ترشهولد بدهیم و عکس را از پیش زمینه پلاک برداریم. همانطور که در کلاس نیز گفته شده بود باید این کار را ابتدا طوری کنیم که همه چیز را پاک کند پس باز باید یک ترشهولد بزنیم، بعد از کل تصویر این را بردارد و بعد کاراکتر ها بماند. این هارا کنار هم در فیگور ساپلات میکنیم.

۶.در این قسمت همانطور که در کلاس توضیح داده شد انگار میخواهیم حروف را با اعداد بسازیم.
با پارت هایی که نگه داشته بودیم و عد جایگاهش این کار را میکنیم.

```
function[n, labeled_pic] = mysegmentation(pic)
    parts = detect_parts(pic, 1);
    labeled_pic = zeros(size(pic));
    n = numel(parts);
    for i = n:-1:1
        part = parts{i};
        for j = 1:size(part,1)
            labeled_pic(part(j,1),part(j,2)) = i;
        end
    end
end
```

حالا میتوان شکل حدودی کاراکتر را پیدا کرد.(یعنی ساخته شده نه اینکه جایگزین شود)
برای اینکه بهتر سگمنت بندی کنیم اطراف کاراکتر های ساخته شده مربع میکشیم که در قسمت بعدی که میخواهیم بیشترین همبستگی را بیابیم راحت تر باشد.

```
%% 1.6
[n, labeled] = mysegmentation(characters);
fprintf("Number of segments: %d", n);
figure;
propied=regionprops(labeled,'BoundingBox');
imshow(~characters);
title('detected characters')
hold on
for m=1:size(propied,1)
    rectangle('Position',propied(m).BoundingBox,'EdgeColor','b','LineWidth',2)
end
hold off
```

detected characters

۷.حالا در این قسمت که سگمنت هارا بدست آورده ایم، باید با یک منبع بررسی کنیم که هر شکل بدست آمده چه حرفی است. برای این کار یک دیتا بیس از حروف انگلیسی و اعداد داریم(که آن را میخوانیم)که باید با هر سگمنت(احتیاج به حلقه برای از ابتدا تا انتها محل ذخیره سازی) همبستگی بگیریم و ببینیم در کدام قسمت بیشترین مقدارآن وجود دارد و آن نقطه ماکزیمم را بعنوان هدف بگیریم و ببینیم چه حرف یا عددی بوده است.

```
%% 1.7
folderPath = './Map Set';
fileNames = dir(fullfile(folderPath, '*.bmp'));
numFiles = length(fileNames);

imageData = cell(numFiles, 2);

for i = 1:numFiles
    imagePath = fullfile(folderPath, fileNames(i).name);
    image = imread(imagePath);

    [~, fileName, ~] = fileparts(fileNames(i).name);
    imageData{i, 1} = image;
    imageData{i, 2} = fileName;
end
```

Number of segments: 10>>

.۸

Number of segments: 10Estimated plate number: DL2CAD0311>>

```
file=fopen("output.txt","a");
fprintf(file, "Estimated plate number : %s \n", res);
```

درون فایل نیز ذخیره میکنیم.

image

detected characters

image

detected characters

image

detected characters

بخش دوم)

روش و راه حل دقیقا به همان صورت است اما دیتا بیس برای همبستگی گرفتن فارسی میشود.

باید با فونت نزدیک به فونت پلاک ها تقریبا درست کنیم. همچنین ابعاد نیز باید درست باشد. از همان توابع قسمت قبل استفاده شده است.

۱.باز کردن فایل:

```
%% 2
clc, clearvars, close all;
%% 2.1 opening the files
[filename, filepath] = uigetfile({'*.jpg;*.png;*.bmp', 'Image Files (*.jpg, *.png,
*.bmp)'}, 'Select an Image');
imagepath = fullfile(filepath, filename);
image = imread(imagepath);
```

۲. تغییر ابعد، خاکستری کردن، سیاه و سفید کردن:

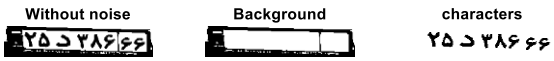
```
%% 2.2 resize the pictures frome any place and distance to ease the process
image = imresize(image, [100,500]);
figure;
imshow(image);
```

```
%% 2.3 making it gray
gray_image = mygrayfun(image);
figure;
imshow(gray_image);
```

```
%% 2.4 making it black and white with binary system 0 1 to only finding the plate and
numbers
binary_image = mybinaryfun(gray_image, 100);
figure;
imshow(binary_image);
```

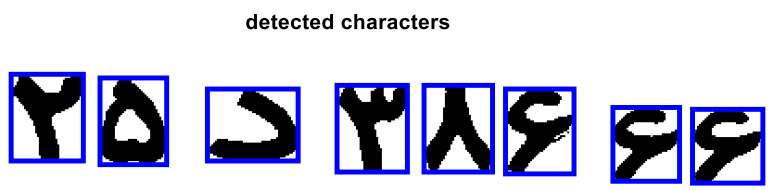
۳. از بین بردن نویز ها و جدا کردن پارت ها و پیش زمینه:

```
%% 2.5 removing noise and the dots and finding the numbers by suntracking the
plate(using threshold)
filtered = myremovecom(~binary_image, 300);
background = myremovecom(~binary_image, 2300);
characters= (filtered - background);

figure;

subplot(1,3,1);
imshow(~filtered);
title('Without noise');

subplot(1,3,2);
imshow(~background);
title('Background');

subplot(1,3,3);
imshow(~characters);
title('characters');
```



۴.سگمنت بندی کردن، کادربندی و ساختن کاراکتر ها:

```
%% 2.6 segmentation
[n, labeled] = mysegmentation(characters);
fprintf("Number of segments: %d", n);
figure;
propied=regionprops(labeled,'BoundingBox');
imshow(~characters);
title('detected characters')
hold on
for m=1:size(propied,1)
    rectangle('Position',propied(m).BoundingBox,'EdgeColor','b','LineWidth',2)
end
hold off
```



۵.همبستگی با ست ساخته شده:

```
%% 2.7 finding map set(Persian alphabet and numbers)
folderPath = './PersianMapSet';
fileNames = dir(fullfile(folderPath, '*.png'));
numFiles = length(fileNames);

imageData = cell(numFiles, 2);

for i = 1:numFiles
    imagePath = fullfile(folderPath, fileNames(i).name);
    image = imread(imagePath);

    [~, fileName, ~] = fileparts(fileNames(i).name);
    imageData{i, 1} = image;
    imageData{i, 2} = fileName;
end
```

```
%% 2.8
res = '';
for i = 1:n
    [row, col] = find(labeled == i);
    y = characters(min(row):max(row), min(col):max(col));
    y = imresize(y,[100,80]);
    corr_max = -2;
    i_max = 0;
    for j = 1:numFiles
        corr_j = corr2(y,imageData{j,1});
        if(corr_j > corr_max)
            corr_max = corr_j;
            i_max = j;
        end
    end
    if(corr_max > 0.3)
        res = strcat(res,imageData{i_max,2});
    end
end
fprintf("The plates numbers are: %s", res);
```

```
Number of segments: 8The plates numbers are: 25DD38666>>
```

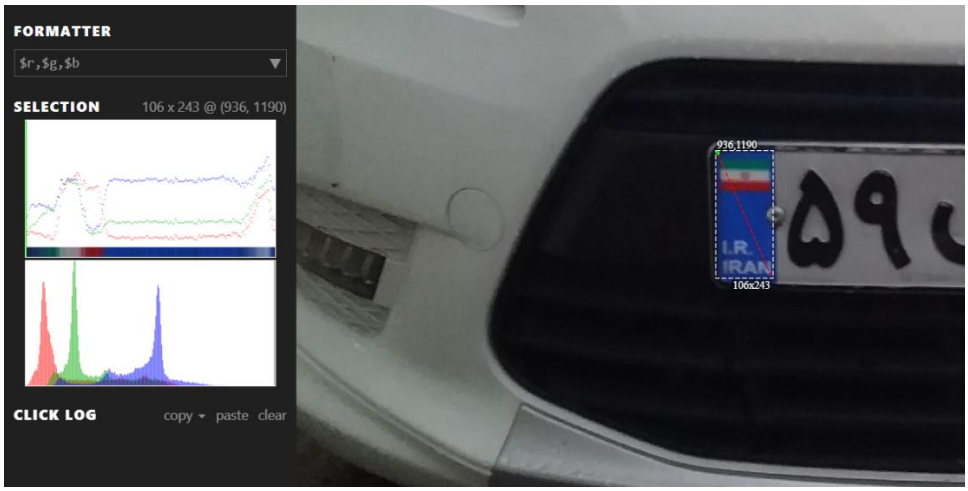
*خروجی به صورت حروف انگلیسی است اما درون ست خودم حروف به صورت خوانا اسم گذاری شده اند.

توجه: درباره ست فارسی ساخته شده بعلت ایجاد خطا و درواقع احتیاج به ایجاد چندین ترشهولد برای نقاط حروفی مثل الف، ق، ف، ع ..... و شکل هم بودن بعظی حروف در نوشتار مثل ع و غ یا ب و ث و پ یا د و ذ ..... فقط شکل آنها قرار داده شده است. همچنین اعداد فونت خاصی از فارسی اند. برای ترشهولد ها جابه جایی انجام میدهیم که اندازه شود و بتواند اعدادی مثل ۳و۲ یا ۰ فارسی را درست نشان دهد اما باز هم باتوجه به نوع عکس ممکن است احتیاج به تغییر داشته باشد.

بخش سوم)

در این سوال ابتدا باید روشی ایجاد کنیم که بتوان پلاک را از روی ماشین برداشت و بعد دوباره با همان روش های سوالات قبل یعنی تغییر ابعاد عکس(چون ممکن است ماشین در هر فاصله ای از ما باشد پس اندازه پلاک کوچک و بزرگ میشود)، خاکستری کردن بعد آرجی بی برای کاهش حجم، سیاه و سفید کردن یا باینری کردن(در این پروژه ما شبیه پلاک واقعی اعداد را مشکی و پیش زمینه را سفید میکنیم) عکس خاکستری برای بررسی راحت تر با استفاده از ترشهولد، تیکه یه تیکه بررسی کردن قسمت های مختلف و از بین بردن نویز و نکه داشتن قسمت های مورد نیاز، بدست آوردن کاراکتر ها و ساختن آنها و کادربندی و بعد همبستگی هر سگمنت با ست فارسی در دسترس ما و بعد اعلام خروجی.

حالا برای مرحله اول باید به دنبال یک خاصیت از پلاک باشیم که در پلاک های فارسی قسمتی از آنها دارای نمونه ابی رنگ و پرچم ایران است. در واقع این قسمت را ملاک میگذاریم و همبستگی با رنگ ابی(بیشترین)، رنگ قزمز و سبز(پرجم) میگیریم. بنابراین در این قسمت میتوان پلاک را تشخیص داد. بعد از آن دوباره مراحل گفته شده برای کاراکتر های پلاک انجام میدهیم.



از فایل ۳ داده شده برای عکس ها استفاده شده است که خودش از داخل آن ۳ تا عکس را بررسی میکند:

```
%% 3
clc,clearvars,close all;
%% photo of the car and doing the same work like p2(finding the plate and correlation
with the set)
template = imread('origin.jpg');
filedest = "p3/";
for k = 1:3
    image = imread(filedest +k+ ".jpg");

    image = imresize(image, [200, 300]);
    template = imresize(template, [20, 10]);

    template = im2double(template);
    image = im2double(image);

    correlationMapR = normxcorr2(template(:,:,1), image(:,:,1));
    correlationMapG = normxcorr2(template(:,:,2), image(:,:,2));
    correlationMapB = normxcorr2(template(:,:,3), image(:,:,3));
    wR = 1;
    wG = 1;
    wB = 1;
    correlationMap = (wR * correlationMapR + wG * correlationMapG + wB *
correlationMapB) / (wR + wG + wB);

    [maxCorrelationValue, maxIndex] = max(correlationMap(:));
    [maxRow, maxCol] = ind2sub(size(correlationMap), maxIndex);
    templateHeight = size(template, 1);
    templateWidth = size(template, 2);
    matchedRegionRow = maxRow - templateHeight + 1;
    matchedRegionCol = maxCol - templateWidth + 1;


    figure;
    subplot(1,2,1);
    imshow(image);
    hold on;
    rectangle('Position', [matchedRegionCol, matchedRegionRow, templateWidth,
templateHeight], 'EdgeColor', 'r', 'LineWidth', 2);
    rectangle('Position', [matchedRegionCol-10, matchedRegionRow-10, 7 *
templateHeight, 2 * templateHeight], 'EdgeColor', 'g', 'LineWidth', 2);
    hold off;
    plate = imcrop(image, [matchedRegionCol-10, matchedRegionRow-10, 7 *
templateHeight, 2 * templateHeight]);
    subplot(1,2,2);
    imshow(mybinaryfun(mygrayfun(imresize(plate, [100, 500])), 0.4));
    figure;
    image = imresize(plate, [100,500]);
    subplot(1,3,1);
    title('original');
    imshow(image);
```

```
gray_image = mygrayfun(image);
subplot(1,3,2);
title('grayscaled');
imshow(gray_image);
binary_image = mybinaryfun(gray_image, 0.4);
subplot(1,3,3);
title('binarized');
imshow(binary_image);
filtered = myremovecom(~binary_image, 150);
background = myremovecom(~binary_image, 2300);
characters= (filtered - background);

figure;

subplot(1,3,1);
imshow(~filtered);
title('Without noise');

subplot(1,3,2);
imshow(~background);
title('Background');

subplot(1,3,3);
imshow(~characters);
title('characters');
[n, labeled] = mysegmentation(characters);
fprintf("Number of segments: %d ", n);
figure('Position', [0 0 900 400]);
propied=regionprops(labeled,'BoundingBox');
imshow(~characters);
title('detected characters')
hold on
for m=1:size(propied,1)
    rectangle('Position',propied(m).BoundingBox,'EdgeColor','b','LineWidth',2)
end
hold off
folderPath = './PersianMapSet';
fileNames = dir(fullfile(folderPath, '*.png'));
numFiles = length(fileNames);

imageData = cell(numFiles, 2);

for i = 1:numFiles
    imagePath = fullfile(folderPath, fileNames(i).name);
    image = imread(imagePath);

    [~, fileName, ~] = fileparts(fileNames(i).name);
    imageData{i, 1} = image;
    imageData{i, 2} = fileName;
end
res = '';
for i = 1:n
    [row, col] = find(labeled == i);
    y = characters(min(row):max(row), min(col):max(col));
    y = imresize(y,[100,80]);
```

```
    corr_max = -2;
    i_max = 0;
    for j = 1:numFiles
        corr_j = corr2(y,imageData{j,1});
        if(corr_j > corr_max)
            corr_max = corr_j;
            i_max = j;
        end
    end
    if(corr_max > 0.3)
        res = strcat(res,imageData{i_max,2});
    end
end
fprintf("The plates numbers are: %s \n", res);

end
```

Without noise

Background

characters

detected characters

Number of segments: 8 The plates numbers are: 59YY84410

Without noise

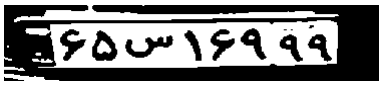
Background

characters

۴۶ ۳۷۸ ق ۶۲

detected characters

Number of segments: 8 The plates numbers are: 62GH27846



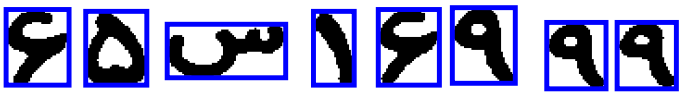
Without noise

Background

characters

۶۵س۱۶۹۹۹

detected characters



Number of segments: 8 The plates numbers are: 65SS16999

بخش چهارم)

ایده این بخش درواقع پیدا کردن ماشین یا پلاک است. یعنی کد باید بتواند تمپلیت یک ماشین یا یک پلاک را روی فیلم مورد نظر بیندازد و بتواند شی را پیدا کند. بعد از آن دوتا فیگور بگیرد و مقایسه کند و بعد از بتواند با میانگین گرفتن از جابه جایی فاصله این ها در زمان مشخص شده، سرعت را بیابد. این قسمت پیدا کردن پلاک همان بخش ۳ است.

در واقعیت در مرحله ی اول ویدیو پردازش می شود و سرعت استخراج می شود. در مرحله ی بعد یک فریم عکس از ویدیو جدا می شود. در مرحله ی بعد جلوبندی و قاب پلاک خودرو از عکس جدا می شود و در نهایت حروف و اعداد پلاک استخراج می شوند.