

گزارش کار تمرین کامپیوتری دوم سیگنال و سیستم

اعضای گروه:

محمد مهدی صمدی (۸۱۰۱۰۱۴۶۵)

سپهر جمالی (۸۱۰۱۰۱۴۰۰)

بخش اول

در ابتدا عکس لود میشود. سپس ریسایز، خاکستری و باینری میشود.

```
1 function final_output=p1()
2     figure;
3     %1
4     [file,path]=uigetfile({'*.jpg;*.bmp;*.png;*.tif'},'Choose an image');
5     s=[path,file];
6     picture=imread(s);
7     subplot(4, 2, 1);
8     imshow(picture);
9     title("raw picture");
10
11     %2
12     picture=imresize(picture,[300 500]);
13     subplot(4, 2, 2);
14     imshow(picture);
15     title("resized picture");
16
17     %3
18     gray_picture = mygrayfun(picture);
19     subplot(4, 2, 3);
20     imshow(gray_picture);
21     title("grayscale picture");
22
23     %4
24     thr = 0.4;
25     binary_picture = mybinaryfun(gray_picture,thr);
26     subplot(4, 2, 4);
27     imshow(binary_picture);
28     title("black and white picture");
29
```

در این بخش کد کامپوننت های اضافه حذف شده، بکگراند تشخیص داده و حذف شده و در نهایت عکسی شامل تنها اعداد و حروف پلاک خواهیم داشت.

```

30 %5
31 cleaned_image = myremovecom(~binary_picture,500);
32 subplot(4, 2, 5);
33 imshow(cleaned_image);
34 title("after removing small components");
35 cleaned_image_background = myremovecom(~binary_picture,2500);
36 subplot(4, 2, 6);
37 imshow(cleaned_image_background);
38 title("background of the picture");
39 cleaned_image_letters = cleaned_image - cleaned_image_background;
40 subplot(4, 2, 7);
41 imshow(cleaned_image_letters);
42 title("only letters and numbers");
43
44 %6
45 [labeled_image, num_components] = mysegmentation(cleaned_image_letters);
46 %%%
47
48 %7
49 load TRAININGSET_ENGLISH;
50 totalLetters=size(TRAIN_ENGLISH,2);
51
52 component_stats = regionprops(labeled_image, 'BoundingBox');
53 bboxes = cat(1, component_stats.BoundingBox);
54 [~, sort_idx] = sort(bboxes(:, 1));
55
56
57
58 subplot(4, 2, 8);
59 imshow(cleaned_image_letters);
60 for n=1:num_components
61     rectangle('Position',component_stats(n).BoundingBox,'EdgeColor','m','LineWidth',1)
62 end
63 title("rectangles around components");

```

در ادامه به ازای هر کامپوننت باقیمانده (ممکن است حتی عدد یا رقم نباشد)، با تمام دیتا های داخل دیتاست کورلیشن میگیریم و اگر از ترشلد مشخص شده ای بالا تر باشد ان را تشخیص میدهیم. در انتها پلاک تشخیص داده شده در فایل نوشته میشود.

```

58 subplot(4, 2, 8);
59 imshow(cleaned_image_letters);
60 for n=1:num_components
61     rectangle('Position',component_stats(n).BoundingBox,'EdgeColor','m','LineWidth',1)
62 end
63 title("rectangles around components");
64
65 final_output=[];
66 t=[];
67
68 for i=1:num_components
69     n = sort_idx(i);
70     [r, c] = find(labeled_image == n);
71     Y=labeled_image(min(r):max(r),min(c):max(c));
72     Y=imresize(Y,[42,24]);
73     ro=zeros(1,totalLetters);
74     for k=1:totalLetters
75         ro(k)=corr2(TRAIN_ENGLISH{1,k},Y);
76     end
77     [MAXRO,pos]=max(ro);
78     if MAXRO>.45
79         out=TRAIN_ENGLISH{2,pos};
80         final_output=[final_output out];
81     end
82 end
83
84 %8
85 file = fopen('English Plate.txt', 'wt');
86 fprintf(file,'%s\n',final_output);
87 fclose(file);

```

تابع find components

```

1 function [labeled_img, num_labels] = find_components(binary_img)
2     [rows, cols] = size(binary_img);
3     labeled_img = zeros(rows, cols);
4     label_count = 1;
5     connectivity_dirs = [-1,0; 1,0; 0,-1; 0,1; -1,-1; -1,1; 1,-1; 1,1];
6
7     for r = 1:rows
8         for c = 1:cols
9             if binary_img(r, c) == 1 && labeled_img(r, c) == 0
10                 stack = [r, c];
11
12                 while ~isempty(stack)
13                     [current_r, current_c] = deal(stack(end, 1), stack(end, 2));
14                     stack(end, :) = [];
15
16                     if current_r > 0 && current_r <= rows && current_c > 0 && current_c <= cols
17                         if binary_img(current_r, current_c) == 1 && labeled_img(current_r, current_c) == 0
18                             labeled_img(current_r, current_c) = label_count;
19                             for d = 1:size(connectivity_dirs, 1)
20                                 neighbor_r = current_r + connectivity_dirs(d, 1);
21                                 neighbor_c = current_c + connectivity_dirs(d, 2);
22                                 stack = [stack; neighbor_r, neighbor_c];
23                             end
24                         end
25                     end
26                 end
27
28                 label_count = label_count + 1;
29             end
30         end
31     end
32
33     num_labels = label_count - 1;
34 end

```

تابع باینری کردن عکس خاکستری (عکس دو بعدی):

```
1 function binary_picture = mybinaryfun(gray_picture,thr)
2     [row, col, ~] = size(gray_picture);
3     binary_picture = zeros(row, col);
4     for i = 1:row
5         for j = 1:col
6             if gray_picture(i, j) > thr
7                 binary_picture(i, j) = 1;
8             else
9                 binary_picture(i, j) = 0;
10            end
11        end
12    end
13    %imshow(binary_picture)
14 end
```

تابع خاکستری کردن عکس رنگی (تبدیل عکس از ۳ بعد به ۲ بعد):

```
1 function gray_picture = mygrayfun(image)
2     [row, col, ~] = size(image);
3     gray_picture = zeros(row, col);
4     for i = 1:row
5         for j = 1:col
6             gray = 0.299 * image(i, j, 1) + 0.587 * image(i, j, 2) + 0.114 * image(i, j, 3) ;
7             gray_picture(i, j) = double(gray) / 255;
8         end
9     end
10    %imshow(gray_picture)
11 end
```

تابع حذف کتامپونن های اضافه:

```
1 function cleaned_image = myremovecom(binary_image,n)
2     [labeled_image,labels] = find_components(binary_image);
3     for i=1 : labels
4         if sum(labeled_image(:) == i) < n
5             labeled_image(labeled_image == i) = 0;
6         end
7     end
8     cleaned_image = labeled_image > 0;
9     %imshow(cleaned_image)
10 end
```

تابع سگمنت کردن:

```
1 function [labeled_image, num_components] = mysegmentation(binary_image)
2     [labeled_image,labels] = find_components(binary_image);
3     num_components = labels;
4     disp(num_components)
5     %imshow(labeled_image)
6 end
```

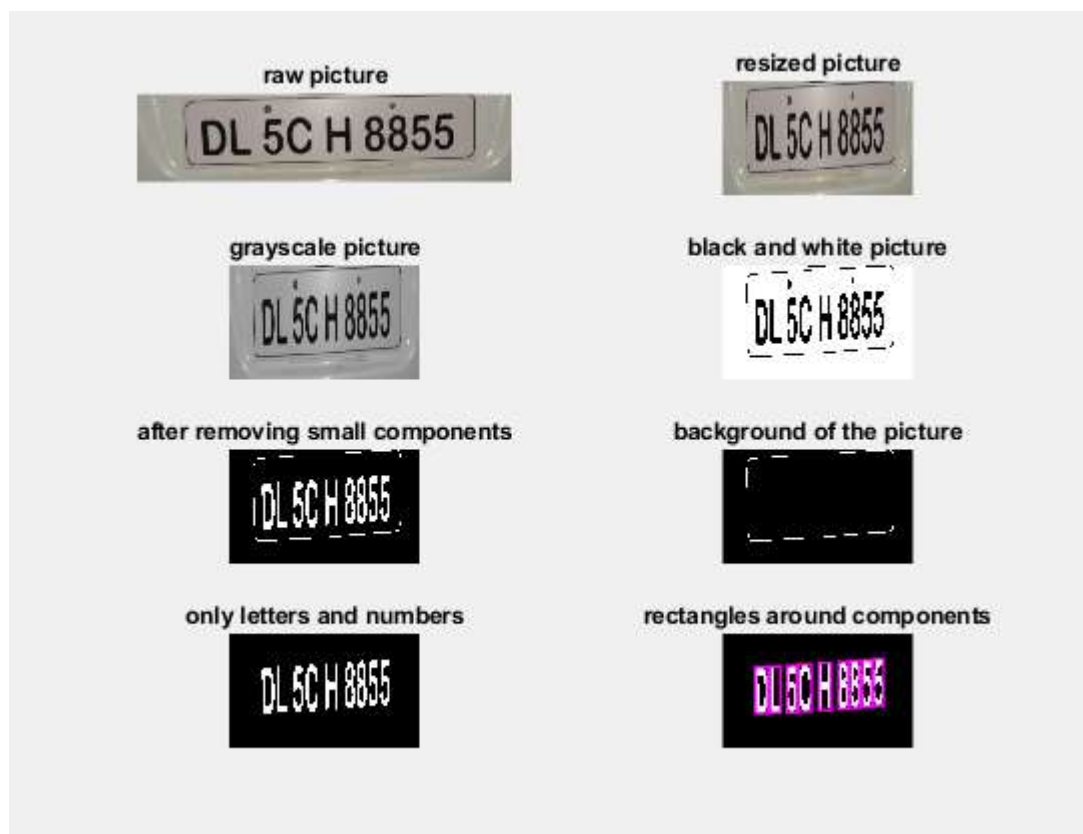
تابع ساختن دیتاست انگلیسی در قالب فایل دات مت:

```
1 function train_p1()
2     files=dir('Map Set (English)');
3     len=length(files)-2;
4     TRAIN_ENGLISH=cell(2,len);
5
6     for i=1:len
7         TRAIN_ENGLISH{1,i}=imread([files(i+2).folder,'\ ',files(i+2).name]);
8         TRAIN_ENGLISH{2,i}=files(i+2).name(1);
9     end
10
11     save TRAININGSET_ENGLISH TRAIN_ENGLISH;
12
```

اسکرپت ران کردن کل بخش اول:

```
1 clc
2 clearvars
3 close all
4
5 train_p1();
6 plate = p1();
7 plate
```

نتیجه اجرای کد بخش اول:



بخش دوم)

این بخش تفاوتی با بخش اول ندارد فقط دیتاست فارسی نیاز داریم

دیتاست فارسی که ما استفاده کردیم:



کلیت این بخش همانند بخش اول است. همچنین برای هر بخش که نیاز بود از توابع آماده متلب استفاده کردیم.

در این جا نیز ابتدا عکس ریسایز، خاکستری و باینری شده. در ادامه کامپوننت های اضافی حذف میشوند. بکگرانند نیز همینطور.

```
1 function final_output=p2(picture)
2     figure;
3
4     subplot(4, 2, 1);
5     imshow(picture);
6     title("raw picture");
7     %%%
8     picture=imresize(picture,[300 500]);
9     subplot(4, 2, 2);
10    imshow(picture);
11    title("resized picture");
12    %%%
13    gray_picture = rgb2gray(picture);
14    subplot(4, 2, 3);
15    imshow(gray_picture);
16    title("grayscale picture");
17    %%%
18    threshold = graythresh(gray_picture);
19    picture = imbinarize(gray_picture,threshold);
20    subplot(4, 2, 4);
21    imshow(picture);
22    title("black and white picture");
23
24    %%%
25    pic = bwareaopen(~picture,1000);
26    subplot(4, 2, 5);
27    imshow(pic);
28    title("after removing small components");
29    background=bwareaopen(~picture,10000);
30
31    subplot(4, 2, 6);
32    imshow(background);
33    title("background of the picture");
```

```

40     %%
41     [labeled_image, num_components] = bwlabel(picture2);
42     %%
43     load TRAININGSET_FARSI;
44
45     totalLetters=size(TRAIN_FARSI,2);
46
47     component_stats = regionprops(labeled_image, 'BoundingBox');
48     bboxes = cat(1, component_stats.BoundingBox);
49     [~, sort_idx] = sort(bboxes(:, 1));
50
51     subplot(4, 2, 8);
52     imshow(picture2);
53     for n=1:num_components
54         rectangle('Position',component_stats(n).BoundingBox,'EdgeColor','m','LineWidth',1)
55     end
56     title("rectangles around components");
57
58     final_output=[];
59     t=[];
60
61     for i=1:num_components
62
63         n = sort_idx(i);
64         [r, c] = find(labeled_image == n);
65         Y=labeled_image(min(r):max(r),min(c):max(c));
66
67         ro=zeros(1,totalLetters);
68         for k=1:totalLetters
69             [row, col] = size(TRAIN_FARSI{1,k});
70             Y2 = imresize(Y, [row, col]);
71             ro(k)=corr2(TRAIN_FARSI{1,k},Y2);
72

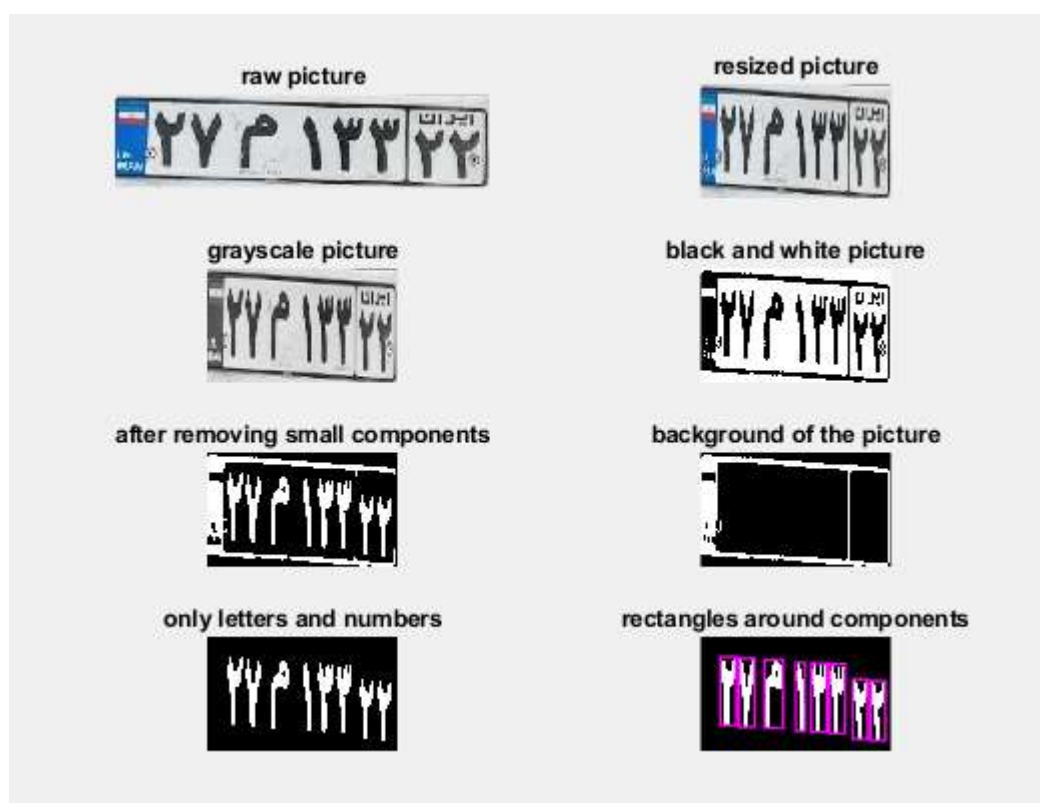
```

```

61     for i=1:num_components
62
63         n = sort_idx(i);
64         [r, c] = find(labeled_image == n);
65         Y=labeled_image(min(r):max(r),min(c):max(c));
66
67         ro=zeros(1,totalLetters);
68         for k=1:totalLetters
69             [row, col] = size(TRAIN_FARSI{1,k});
70             Y2 = imresize(Y, [row, col]);
71             ro(k)=corr2(TRAIN_FARSI{1,k},Y2);
72
73         end
74         [MAXRO,pos]=max(ro);
75         if MAXRO>.45
76             out=TRAIN_FARSI{2,pos};
77             final_output=[final_output out];
78         end
79     end
80     %8
81     fileID = fopen('Farsi Plate.txt', 'w', 'n', 'UTF-8');
82     fprintf(fileID, '%s\n', final_output);
83     fclose(fileID);

```


نتیجه اجرای کد بخش دوم:



بخش سوم)

در ابتدا به کمک بخش آبی کنار پلاک (سمت چپ پلاک)، یک **bounding box** حدودی برای پلاک در میاریم و سپس عکس کراپ شده توسط آن را به کد بخش دوم میدهیم.

دو عکس از آن قسمت آبی با سایز و کیفیت های مختلف پیدا کردیم و با کمک آن ها این کار را انجام دادیم (دو عکس در فایل زیپ آپلودی هستند).

این تابع عکس ورودی را گرفته و بعد از ریسایز آن، در حلقه فور با تعداد زیادی از اسکیل های مختلف آن عکس آبی کورلیشن میگیرد و بالاترین کورلیشن را ذخیره میکند. در ادامه همین کار برای عکس دوم نیز انجام میشود (به همان صورت قبل)

```

1 function bounding_box = p3(picture_full)
2     RESIZE_WIDTH = 800;
3     ERR_MARGIN = 10;
4     BLUE2PLATE_RATIO = 14;
5     CORRELATION_THRESHOLD = 0.5;
6
7     blue_part_org = imread('blue_part.png');
8
9     picture = imresize(picture_full, [NaN, RESIZE_WIDTH]);
10    ratio = size(picture_full, 1) / size(picture, 1);
11
12    [corr_mix, corr_max, bbox] = rgb_corr2(blue_part_org, picture);
13
14    for r=0.2:0.2:10
15        % resizing the picture by a scale
16        blue_part = imresize(blue_part_org, r);
17        if size(picture, 1) > size(blue_part, 1)
18            [new_corr_mix, new_corr_max, new_bbox] = rgb_corr2(blue_part, picture);
19            if new_corr_max > corr_max
20                if new_corr_max > CORRELATION_THRESHOLD
21                    corr_max = new_corr_max;
22                    corr_mix = new_corr_mix;
23                    bbox = new_bbox;
24                end
25            end
26        end
27    end
28
29    blue_part_org2 = imread('blue_part2.png');
30
31    picture = imresize(picture_full, [NaN, RESIZE_WIDTH]);
32    ratio = size(picture_full, 1) / size(picture, 1);
33
34    for r=0.2:0.2:10
35        % resizing the picture by a scale
36        blue_part2 = imresize(blue_part_org2, r);
37        if size(picture, 1) > size(blue_part2, 1)
38            [new_corr_mix, new_corr_max, new_bbox] = rgb_corr2(blue_part2, picture);
39            if new_corr_max > corr_max
40                if new_corr_max > CORRELATION_THRESHOLD
41                    corr_max = new_corr_max;
42                    corr_mix = new_corr_mix;
43                    bbox = new_bbox;
44                end
45            end
46        end
47    end
48

```

سپس در صورتی که پلاک یافت نشد پیغامی پرینت میشود ولی اگر پیدا شد عکس اصلی، عکس اصلی به همراه باندینگ باکس‌ها و در آخر عکس کراپ شده پلاک نمایش داده میشوند.

```

50
51     if corr_max < CORRELATION_THRESHOLD
52         fprintf("%s", "DID NOT FOUND THE BLUE PART");
53     end
54     if corr_max >= CORRELATION_THRESHOLD
55         bbox_full = [round((bbox(1) - ERR_MARGIN) * ratio), ...
56                     round((bbox(2) - ERR_MARGIN) * ratio), ...
57                     round((bbox(3) + 2 * ERR_MARGIN) * ratio), ...
58                     round((bbox(4) + 2 * ERR_MARGIN) * ratio)];
59
60         bounding_box = bbox_full;
61         bounding_box(3) = BLUE2PLATE_RATIO * bbox(3) * ratio;
62
63         bounding_box
64
65         figure;
66         cols = 1;
67         rows = 3;
68         cnt = 1;
69         subplot(rows, cols, cnt);
70         cnt = cnt + 1;
71         imshow(picture);
72         title('Picture')
73
74         subplot(rows, cols, cnt);
75         cnt = cnt + 1;
76         imshow(picture_full);
77         rectangle('Position', bbox_full, 'edgecolor', 'r', 'linewidth', 2);
78         rectangle('Position', bounding_box, 'edgecolor', 'g', 'linewidth', 1);
79         title("Bounding boxes around the detected parts");
80
81
82         subplot(rows, cols, cnt);
83         imshow(iscrop(picture_full, bounding_box));
84         title("The cropped part");
85     end
86 end
87

```

در کد بالا از تابع کورلیشن رنگی استفاده شده که کد آن به صورت زیر است (کد زیر را از سایتی برداشتم و خودم ننوشتم)

```

1 function [corr_mix, corr_max, bbox] = rgb_corr2(template, pic)
2     corrRed = normxcorr2(template(:, :, 1), pic(:, :, 1));
3     corrGrn = normxcorr2(template(:, :, 2), pic(:, :, 2));
4     corrBlu = normxcorr2(template(:, :, 3), pic(:, :, 3));
5     corr_mix = (corrRed + corrGrn + corrBlu) / 3;
6
7     [corr_max, corrIdx] = max(abs(corr_mix(:)));
8     [peakY, peakX] = ind2sub(size(corr_mix), corrIdx(1));
9     corr_offset = [peakX - size(template, 2), peakY - size(template, 1)];
10    bbox = [corr_offset(1), corr_offset(2), size(template, 2), size(template, 1)];
11 end

```

در انتها نیز اسکریپت زیر کد بخش سوم را اجرا میکند.

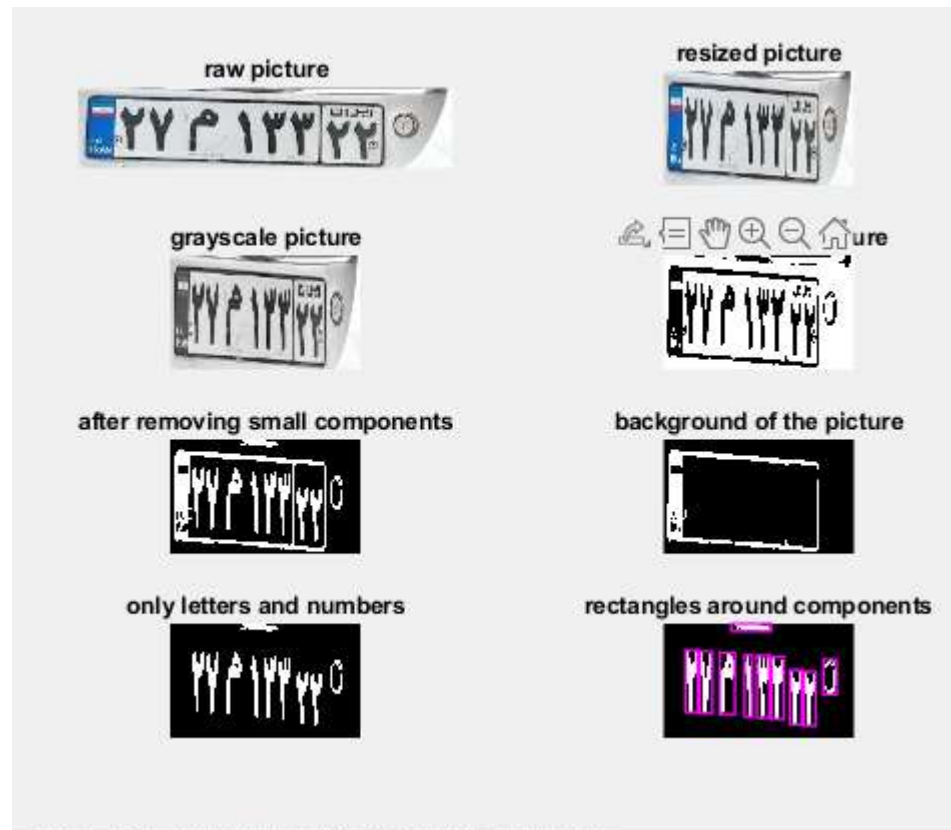
```

1   clc
2   clearvars
3   close all
4
5   [file,path]=uigetfile({'*.jpg;*.bmp;*.png;*.tif'}, 'Choose an image');
6   picture=imread([path,file]);
7   bb = p3(picture);
8   cropped_plate = imcrop(picture, bb);
9   train_p2();
10  plate = p2(cropped_plate);
11  plate

```

نتیجه کد بالا بر روی یکی از ۳ عکس داده شده (ماشین ۲۰۶)





بخش چهارم)

برای این بخش گفته شده که از بخش سوم کمک بگیریم. من و هم تیمی ام به علت کمبود وقت به صورت موازی این دو بخش رو زدیم و کد بخش چهارم کاملاً مستقل از کد بخش سوم است.

```

1  vidObj = VideoReader('car_vid.mp4');
2  timestamp = 3.9; % Accessing the frame at 1 seconds
3  frame_number = round(vidObj.FrameRate * timestamp);
4  vidObj.CurrentTime = (frame_number - 1) / vidObj.FrameRate;
5  frame1 = readFrame(vidObj);
6  imshow(frame1);
7  imwrite(frame1, 'plate_frame.png');
8
9  timestamp = 4; % Accessing the frame at 1.5 seconds
10 frame_number = round(vidObj.FrameRate * timestamp);
11 vidObj.CurrentTime = (frame_number - 1) / vidObj.FrameRate;
12 frame2 = readFrame(vidObj);
13 imshow(frame2);
14
15 gray1 = rgb2gray(frame1);
16 gray2 = rgb2gray(frame2);
17 points1 = detectSURFFeatures(gray1);
18 points2 = detectSURFFeatures(gray2);
19 [features1, validPoints1] = extractFeatures(gray1, points1);
20 [features2, validPoints2] = extractFeatures(gray2, points2);
21 indexPairs = matchFeatures(features1, features2);
22 matchedPoints1 = validPoints1(indexPairs(:, 1), :);
23 matchedPoints2 = validPoints2(indexPairs(:, 2), :);
24 displacements = matchedPoints2.Location - matchedPoints1.Location;
25 averageDisplacement = mean(sqrt(sum(displacements.^2, 2)));
26 timeInterval = 0.1;
27 velocity = averageDisplacement / timeInterval;
28 disp(['Velocity: ', num2str(velocity), ' pixels per second']);
29
30 [file,path]=uigetfile({'*.jpg;*.bmp;*.png;*.tif'}, 'Choose an image');
31 s=[path,file];
32 picture=imread(s);
33 figure
34 subplot(3,3,1)
35 imshow(picture)
36 title('Original Image')
37

```

```

37
38 %2
39 cropfunc(picture);
40 picture=imread("cropped_pic.jpg");
41 subplot(3,3,2)
42 imshow(picture)
43 title('Cropped Image')
44
45 %3
46 picture=imresize(picture,[300 500]);
47 subplot(3,3,3)
48 imshow(picture)
49 title('Resized Image')
50
51
52 %4
53 picture=rgb2gray(picture);
54 subplot(3,3,4)
55 imshow(picture)
56 title('Grayscale Image')
57
58 %5
59 %threshold = graythresh(picture);
60 threshold = 0.35;
61 picture = ~imbinarize(picture,threshold);
62 subplot(3,3,5)
63 imshow(picture)
64 title('Binary Image')
65
66 %6
67 number_of_pixels=700;
68 picture = bwareaopen(picture,number_of_pixels);
69 background=bwareaopen(picture,5500);
70 picture=picture-background;
71 subplot(3,3,6)
72 imshow(picture)
73 title('Clean Image')

```

```

72 imshow(picture)
73 title('Clean Image')
74
75 %7
76 [picture,Ne]=bwlabel(picture);
77 subplot(3,3,8)
78 imshow(picture)
79 title('Segmentated Image')
80
81 %8
82 load TRAININGSETPER;
83 totalLetters=size(TRAINPER,2);
84
85 figure
86 final_output=[];
87 for n=1:Ne
88     [r,c]=find(picture==n);
89     Y=picture(min(r):max(r),min(c):max(c));
90     imshow(Y)
91     Y=imresize(Y,[42,24]);
92     imshow(Y)
93     pause(0.2)
94
95     ro=zeros(1,totalLetters);
96     for k=1:totalLetters
97         ro(k)=corr2(TRAINPER{1,k},Y);
98     end
99     [MAXRO,pos]=max(ro);
100     if MAXRO>.5
101         out=cell2mat(TRAINPER(2,pos));
102         final_output=[final_output out];
103     end
104 end
105

```

```

%5
%6 %9
%7 disp(final_output);
%8 file = fopen('Number_Plate_4.txt', 'wt');
%9 fprintf(file,'%s\n',final_output);
%10 fclose(file);
%11 winopen('Number_Plate_4.txt')

```

نتیجه اجرای این کد روی ویدیو گرفته شده:

```

fx >> Velocity: 78.5831 pixels per second

```