

پروژه دوم درس سیگنال‌ها و سیستم‌ها

فاطمه‌زهرآ برونندیا-۸۱۰۱۰۰۹۴

پارت اول:

تابع **mygrayfun**:

عکس در ابتدا سه بعدی است؛ در واقع از سه **table** برای قرمز و سبز و آبی تشکیل شده که می‌خواهیم آن را به دو بعد و یک جدول که هر خانه‌اش از ۰ تا ۲۵۵ مقدار بگیرد برسانیم. طبق ضرایب داده شده از صورت پروژه، در این سه جدول ضرب شده‌اند و نهایتاً نتیجه در **table** عکس **grayscale** آمده.

```
1 function outPic = mygrayfun(picture)
2     coef=[0.299, 0.578, 0.114];
3     Rpart = coef(1).*picture(:,:,1);
4     Bpart = coef(2).*picture(:,:,2);
5     Gpart = coef(3).*picture(:,:,3);
6     outPic= Rpart+Bpart+Gpart;
7 end
```

تابع **mybinaryfun**:

عملکرد این تابع ساده است، طبق **threshhold** که ورودی می‌گیرد، برای تک‌تک خانه‌های عکس تصمیم می‌گیرد. اگر بالاتر از **threshhold** باشند مقدار ۱ و اگر پایین‌تر باشند مقدار ۰ می‌گیرند. به این ترتیب عکس دودویی می‌شود.

```
1 function outPic = mybinaryfun(picture,threshhold)
2     th=threshhold*ones(size(picture));
3     outPic= (picture > th);
4
5 end
```

تابع **myremovecom**:

در این تابع، ابتدا باید آبجکت‌ها را به صورت ماتریسی از موقعیت **point**‌هایشان تشکیل دهیم و سپس اگر تعداد **point**‌هایشان از ورودی تابع کمتر بود، خانه نقاط آن آبجکت در تصویر را صفر کنیم (که حذف شود)

برای تشکیل‌دادن آبجکت‌ها، ابتدا تمام خانه‌هایی که مقدار ۱ دارند را پیدا می‌کنیم؛ حالا از نقطه اول شروع می‌کنیم، آبجکت اول را تشکیل می‌دهیم و ۸ همسایگی آن را به کمک تابع کمکی **myfindneighbors** بررسی می‌کنیم. اگر آن‌ها هم ۱ بودند آن‌ها را از مجموعه نقاط حذف می‌کنیم و به آبجکت نقطه اول اضافه می‌کنیم. حالا باید مجدد همسایگی‌های این مجموعه نقاط چک شوند و... اگر کار به جایی برسد که هیچ

همسایگی دیگری از هیچ نقطه‌ای نمانده باشد که به این آبجکت اضافه شده باشد، دوباره اولین نقطه از مجموعه نقاط را انتخاب می‌کنیم و آبجکت بعدی را تشکیل می‌دهیم و... کار تا جایی که مجموعه نقاط مقدار ۱ تمام شود، ادامه پیدا می‌کند.

```
1 function outPic = myremovecom(picture,n)
2     [row, col]=find(picture==1);
3     points=[row';col'];
4
5     flagObjNum=1;
6     currObjNum=1;
7     OBJECT={};
8     while size(points,2)>0
9         firstPoint=points(:,1);
10        points(:,1)=[];
11        [points, neighbors] = myFindNeighbors(firstPoint, points);
12        currObj=[firstPoint neighbors];
13        newPointsLen=size(neighbors,2);
14        while newPointsLen>0
15            newPointSpare2=[];
16            for i=1:newPointsLen
17                m=neighbors(:,i);
18                [points, newPointSpare] = myFindNeighbors(neighbors(:,i), points);
19                newPointSpare2=[newPointSpare2 newPointSpare];
20            end
21            currObj=[currObj newPointSpare2];
22            neighbors=newPointSpare2;
23            newPointsLen=size(neighbors,2);
24        end
25        %OBJECT(currObjNum)=currObj;
26        OBJECT=[OBJECT,currObj];
27        currObjNum=currObjNum+1;
28        %c
29    end
30
31
32    currObjNum=currObjNum-1;
33    r1=[];
34    c1=[];
35    for i=1:currObjNum
36        if size(OBJECT{1,i},2)<n
37            r1=[r1 OBJECT{1,i}(1,:)];
38            c1=[c1 OBJECT{1,i}(2,:)];
39        end
40    end
41
42    index=sub2ind(size(picture),r1,c1);
43    outPic=picture;
44    outPic(index)=0;
45 end
```

تابع mysegmentation:

برای پیدا کردن آبجکت‌ها، مشابه تابع myremovecom عمل می‌کند. پس از اینکه آبجکت‌ها را پیدا کرد، ماتریسی جدید با مقادیر ۰ در اندازه تصویر می‌سازد. سپس از آبجکت اول تا ان‌ام شروع کرده و در ایندکس نقاط آن آبجکت، شماره متناظر آن آبجکت را در جدول قرار می‌دهد.

```
1 function [labeledPic num] = mysegmentation(picture)
2     [row, col]=find(picture==1);
3     points=[row';col'];
4
5     flagObjNum=1;
6     currObjNum=1;
7     OBJECT={};
8     while size(points,2)>0
9         firstPoint=points(:,1);
10        points(:,1)=[];
11        [points, neighbors] = myFindNeighbors(firstPoint, points);
12        currObj=[firstPoint neighbors];
13        newPointsLen=size(neighbors,2);
14        while newPointsLen>0
15            newPointSpare2=[];
16            for i=1:newPointsLen
17                m=neighbors(:,i);
18                [points, newPointSpare] = myFindNeighbors(neighbors(:,i), points);
19                newPointSpare2=[newPointSpare2 newPointSpare];
20            end
21            currObj=[currObj newPointSpare2];
22            neighbors=newPointSpare2;
23            newPointsLen=size(neighbors,2);
24        end
25        %OBJECT(currObjNum)=currObj;
26        OBJECT=[OBJECT, currObj];
27        currObjNum=currObjNum+1;
28        %c
29    end
30
31    currObjNum=currObjNum-1;
32    labeledPic=zeros(size(picture));
33    for i=1:currObjNum
34        r1=OBJECT{1,i}(1,:);
35        c1=OBJECT{1,i}(2,:);
36        index=sub2ind(size(picture),r1,c1);
37        labeledPic(index)=i;
38    end
39    num=currObjNum;
```

```

2 - [file,path]=uigetfile({'*.jpg;*.bmp;*.png;*.tif'},'Choose an image');
3 - s=[path,file];
4 - picture=imread(s);
5 - figure
6 - subplot(1,2,1)
7 - imshow(picture)
8 - picture=imresize(picture,[300 500]);
9 - subplot(1,2,2)
10 - imshow(picture)
11 - %RGB2GRAY
12 - %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 - picture=mygrayfun(picture);
14 - figure
15 - subplot(1,2,1)
16 - imshow(picture)
17 - % THRESHOLDING and CONVERSION TO A BINARY IMAGE
18 - %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
19 - threshold = graythresh(picture);
20 - picture =~mybinaryfun(picture,threshold*256);
21 - subplot(1,2,2)
22 - imshow(picture)
23 - % Removing the small objects and background
24 - %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
25 - picture = myremovecom(picture,500); % removes all connected components (objects) that have fewer than 500 pixels from the binary

28 - background=myremovecom(picture,2500);
29 - subplot(1,3,2)
30 - imshow(background)
31 - picture2=picture-background;
32 - subplot(1,3,3)
33 - imshow(picture2)
34 - %%
35 - % Labeling connected components
36 - %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
37 - figure
38 - imshow(picture2)
39 - [L,Ne]=mysegmentation(picture2);
40 - propied=regionprops(L,'BoundingBox');
41 - hold on
42 - for n=1:size(propied,1)
43 -     rectangle('Position',propied(n).BoundingBox,'EdgeColor','g','LineWidth',2)
44 - end
45 - hold off
46 - % Decision Making
47 - %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
48 - % Loading the mapset
49 - load TRAININGSET;
50 - totalLetters=size(TRAIN,2);

```

```

52 - figure
53 - final_output=[];
54 - t=[];
55 - for n=1:Ne
56 -     [r,c]=find(L==n);
57 -     Y=picture2(min(r):max(r),min(c):max(c));
58 -     imshow(Y)
59 -     Y=imresize(Y,[42,24]);
60 -     imshow(Y)
61 -     pause(0.2)
62 -
63 -     %%
64 -     ro=zeros(1,totalLetters);
65 -     for k=1:totalLetters
66 -         ro(k)=corr2(TRAIN{1,k},Y);
67 -     end
68 -     %%
69 -     [MAXRO,pos]=max(ro);
70 -     if MAXRO>.45
71 -         out=cell2mat(TRAIN(2,pos));
72 -         final_output=[final_output out];
73 -     end
74 - end

76 % Printing the plate
77 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
78 file = fopen('number_Plate.txt', 'wt');
79 fprintf(file, '%s\n', final_output);
80 fclose(file);
81 winopen('number_Plate.txt')
82 toc

```

پارت دوم:

در این بخش چون پلاک‌ها فارسی شده‌اند، لازم است مپست را عوض کنیم. مشکل زبان فارسی این است که حروف نقطه دارند و هنگام استفاده از `bwareopen` برای حذف نویزها، نقطه‌ها هم حذف می‌شوند. برای حل این مشکل نقطه‌ها را از مپست فارسی حذف کردم که نتیجه `correlation` گیری صحیح‌تر شود (چون حروف داده شده، بدون نقطه با یکدیگر اشتباه گرفته نمی‌شوند)

پارت سوم:

چون این بار یک عکس کامل از جلوبندی داریم و نه خود پلاک، باید محدوده پلاک را در تصویر تشخیص دهیم. میتوان گفت ناحیه‌ای که پلاک در آن قرار دارد تغییرات زیادی از سیاه به سفید یا از سفید به سیاه دارد پس ایده تشخیص ناحیه پلاک این است که ببینیم در چه سطر و ستونی از عکس، ۱. تغییرات از سیاه به سفید یا از سفید به سیاه بیشتر است و ۲. این سطر و ستون در مکان مناسبی از تصویر باشد؛ (تقریباً وسط) چون ممکن است به طور تصادفی شرط بالای در ناحیه‌های بسیار پایین یا بسیار بالای عکس محقق شود.

پس از تشخیص این ردیف و ستون، باید به قاب پلاک دست پیدا کنیم. برای این کار، برای مثال اگر دنبال خط بالای پلاک باشیم، دنبال اولین سطری می‌گردیم که میزان تغییراتش نزدیک به بیشینه تغییرات در یک سطر باشد و همچنین در جای مناسبی باشد؛ مثلاً خیلی بالاتر از پلاک نباشد چون ممکن است در سطرهای بالاتر از سطر با بیشینه تغییرات هم سطحی با این ویژگی داشته باشیم که یافتن این محدوده با عدد دادن‌های نسبی به دست آمد. برای یافتن چپ و راست پلاک هم کاملاً مشابه با یافتن بالا و پایین عمل می‌کنیم.

پس از آن، با پیدا شدن قاب پلاک، مانند دو بخش قبلی عکس را **segment** بندی می‌کنیم و سپس **label** می‌زنیم و با **correlation** گیری حرفی که بیشترین شباهت را با آن **segment** دارد پیدا می‌کنیم.

```

44 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%joloabandi%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
45
46 - rowDiff=sum(abs([picture2(:,2:size(picture2,2)) repmat(picture2(:,size(picture2,2)),1,1)]-picture2(:,:)),2);
47 - [maxRowDiff, maxRowDiffIdx]=max(rowDiff(300:500))
48 - maxRowDiffIdx=maxRowDiffIdx+300;
49
50 - topArr=find(abs(rowDiff(maxRowDiffIdx-50:maxRowDiffIdx)-repmat(maxRowDiff,51,1))<20);
51 - top=maxRowDiffIdx-50+topArr(1);
52
53 - bottomArr=find(abs(rowDiff(maxRowDiffIdx:maxRowDiffIdx+50)-repmat(maxRowDiff,51,1))<20);
54 - bottom=maxRowDiffIdx+bottomArr(length(bottomArr));
55
56
57 - colDiff=sum(abs([picture2(2:size(picture2,1),:) ; repmat(picture2(size(picture2,1),:),1,1)]-picture2(:,:)),1);
58 - [maxColDiff, maxColDiffIdx]=max(colDiff(300:400))
59 - maxColDiffIdx=maxColDiffIdx+300;
60
61 - leftArr=find(abs(colDiff(1,maxColDiffIdx-230:maxColDiffIdx)-repmat(maxColDiff,1,231))<30);
62 - left=maxColDiffIdx-230+leftArr(1);
63
64 - rightArr=find(abs(colDiff(1,maxColDiffIdx:maxColDiffIdx+300)-repmat(maxColDiff,1,301))<30);
65 - right=maxColDiffIdx+rightArr(length(rightArr));
66
67 - figure
68 - imshow(picture2(top:bottom,left:right));

```

خروجی پارت اول:

عکس اول:

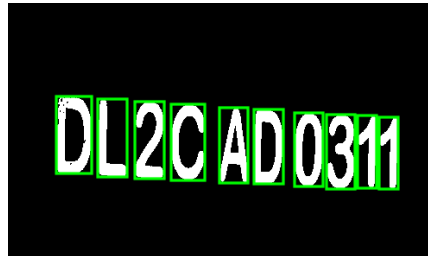


number_Plate.txt - Notepad

File Edit Format View Help

DL5CH8855

عکس دوم:



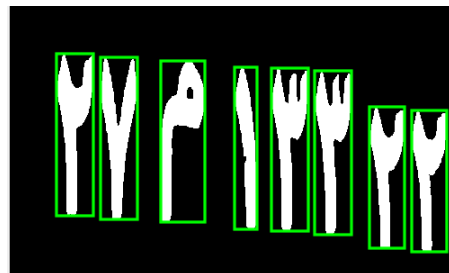
number_Plate.txt - Notepad

File Edit Format View Help

DL2CAdo311

عکس سوم:

خروجی پارت دوم:



farsi_number_Plate.txt - Notepad

File Edit Format View Help

27م13322

خروجی پارت سوم:

عکس اول:



farsi_number_Plate.txt - Notepad

File Edit Format View Help

۵۹۵۸۴۴۱۰

عکس دوم:



farsi_number_Plate.txt - Notepad

File Edit Format View Help

۲۷م ۱۳۳۲۲

عکس سوم:



farsi_number_Plate.txt - Notepad

File Edit Format View Help

62,46378