دانشگاه تهران

پردیس دانشکدگان فنی

دانشکده برق و کامپیوتر

سیگنال و سیستم

تمرین کامپیوتری شماره ۲

استاد: دکتر سعید اخوان

نام و نام خانوادگی

سید علی رضاعی

شماره دانشجویی

۸۱۰۱۰۳۴۳۲

بهار ۱۴۰۵

بخش ۱

کد اصلی

```
[file,path] = uigetfile({'*.jpg;*.bmp;*.png;*.tif'},'Choose an image');
s = [path,file];

picture = imread(s);
picture = imresize(picture,[300 500]);

gray_image = mygrayfun(picture);

wb = mybinaryfun(gray_image, 0.3); % tresh hold must be some value between 0 and 1

clean_image = myremoveobject(wb, 500);


[L , Ne] = mysegmentation(clean_image);

propied=regionprops(L,'BoundingBox');
hold on
for n=1:Ne
    rectangle('Position',propied(n).BoundingBox,'EdgeColor','g','LineWidth',2)
end
hold off

load TRAININGSET;
totalLetters=size(TRAIN,2);



final_output=[];
t=[];

for n=1:Ne

    if n==7
       saeed=1;
    end


    [r,c]=find(L==n);
    Y=clean_image(min(r):max(r),min(c):max(c));
    Y=imresize(Y,[42,24]);



    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



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

خط ۱: با استفاده از دستور uigetfile عکس را گرفتیم در ورودی اولی تعیین میکنیم که از چه تایپ عکسی میتونه استفاده بشه و آپلود بشه و در ورودی بعدی اینکه در پنجره باز شده چه چیزی نوشته شود و این دوتا خروجی داره یکی خود فایل یکی هم آدرسش

خط۲: این آدرس و فایل رو توی s ذخیره کردیم

خط ۴: با استفاده از دستور imread میره و اون عکس رو میخونه

خط ۵: با استفاده از دستور imresize تصویر را ابعادشو تغییر میدهیم این دستور ورودی اول عکس را میگیره و در ورودی دوم ابعاد جدید را میگیرد که در این جا ابعاد مد نظر ۳۰۰ در ۵۰۰ بوده است.

در خط بعد به توضیح تابع mygrayfun میپردازیم

```
function [gray_image] = mygrayfun(RGB_image)

gray = zeros(300,500);

for i = 1:300
    for j = 1:500
        gray(i,j) = 0.299*RGB_image(i,j,1) + 0.587*RGB_image(i,j,2) + 0.114*RGB_image
    end
end
gray_image = gray/255;
figure(1);
imshow(gray_image);
```

در این تابع یک عکس ورودی میگیره یه عکس تبدیل شده به خاکستری تحویل میده برای کار ابتدا یه ماتریس خالی تعریف میکنیم در ادامه میاییم و برای هر I , j ای قسمت های سوم آن که اولی مربوط به رنگ قرمز دومی مربوط به رنگ سبز و سومی مربوط به زنگ آبی است با توجه به دستور العمل با ضرایبی که تعیین شده ضرب میکنیم و این را در gray(i,j) ذخیر میکنیم در ادامه برای اینکه تصویر را با استفاده از imshow بخواهیم نشان بدهیم باید اعداد بین ۰ و ۱ داده شده باشند برای همین همه رو تقسیم بر ۲۵۵ کردیم و سپس در imshow استفاده کردیم.

در ادامه ی کد اصلی به تابع mybinaryfun میرسیم که قرار است یه عکس خاکستری بهش بدیم یعنی تمام آرایه هاش بین ۰ و ۱ باشه و بعد بیاد با توجه به ترشهولدی که در نظر میگیریم برامون کاملا لاجیکال اش بکنه

```
function [black_and_white] = mybinaryfun(gray_image, treshHold)

WB = zeros(300,500);

for i = 1:300
    for j = 1:500
        if gray_image(i,j) < treshHold
            WB(i,j) = 0;
        else
            WB(i,j) = 1;
        end
    end
end
black_and_white = ~WB;
figure(2);
imshow(black_and_white);

end
```

در این جا خیلی ساده دوتا حله فور داریم که میگیم که اگر اون gray_image(i,j) مون کمتر از ترشهولد بود ۰ اش کن و اگر بیشتر بود ۱ اش کن بعد برای اینکه اعداد که مقدار دارند رو ۱ بکنیم اومدم یه نقیض گذاشتم پشت میتونیم توی همون حلقه فور هم جای ۰ و ۱ رو عوض کنم مشکلی نبود ولی خب اینطوری نوشتم و در ادامه میاد و اون تصویر کامل سیاه و سفید رو بهمون نشون میده.

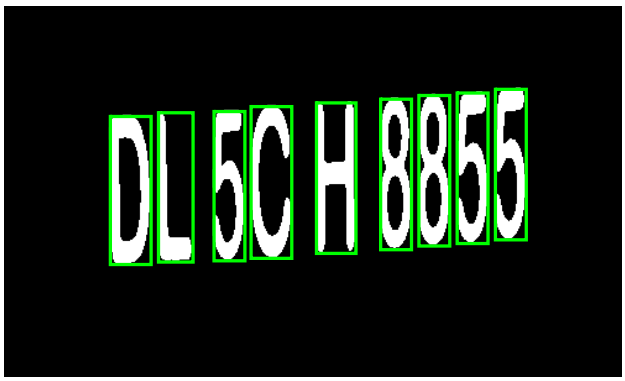
در ادامه توی کل اصلی میرسیم به تابع myremoveobjet که یه عکس و مقداری پیکسلی که چسبیده به هم هستن تا ریمو بشوند را مشخص میکینم

```
function [clean_image] = myremoveobject(picture, minSize)
cc = bwconncomp(picture,4);
clean_image = zeros(300,500);

for i = 1:cc.NumObjects
    if numel(cc.PixelIdxList{i}) >= minSize
        clean_image(cc.PixelIdxList{i}) = 1;
    end
end
figure(3);
imshow(clean_image);

end
```

توی این تابع از دستور bwconncomp استفاده کردم این خیلی دستور جالبیه یعنی connected component ها رو شناسایی میکنه دوتا ورودی میگیره اولی اون عکسیه که میخواییم کامپوننت های به هم چسبیده اش رو پیدا کنیم دومی میگه که ۸ تایی بگردم یا ۴ تایی یعنی فرض کنید که یه ماتریس داریم ۳ در ۳ اگر ۸ تایی رو بزنیم که دیفالتشم همون ۸ تاییه میاد چی کار میکنه مگرده تمام ۸ تا خونه ی دورش که اگر مثلا وسطی یک باشه کدوم یکی از اون ۸ تا کناریش ۱ هست تا با اون کانکت بشه ولی توی ۴ تایی میاد و فقط بالا و پایین و چپ و راستش رو نیگا میکنه. خروجی این هم یه استراکچر هست که ۴ تا کامپوننت داره توش یکی تعدادشه که در ادامه ازش استفاده کردیم یکی دیگه میاد یه سلول میده از اون پیکسل هایی که به هم وصل هستن اومدم دوباره یه clean_image خالی تعریف کردم بعد توی حلقه فور میره میگرده اگر number of element اون پیکسل هایی که به هم وصل هستن بیشتر از اون مینیموم سایزمون بود اونارو میکنیم ۱ و اونایی هم که کمترن که هیچی دیگه صفر میموند چون اولش زیروس تعریف کردم در نهایت هم میام و imshow میکنم.

در ادامه به تابع mysegmentation میرسیم این تابع قراره بیاد و تصویر مارو سگمنت بندی بکنه دوتا خروجی بده بهمون اولی اینکه ماتریس جدیدی که سگمنت بندی شده رو بده و Ne هم تعدادشو دوباره میاییم و از bwconncomp استفاده میکنیم و میگیم توی حلقه فور new_segmantation و توش میگیم کدوم هارو اون segment.PixelIlxList آی ام رو اونارو مقدار هاشو بذار آی در واقع گفتیم چون خروجی bwconncomp یه استراکچره پس از دات در صدا زدنش استفاده میکنیم و چون حالا این پیکسل آی لیست یه سلوله از کروشه استفاده میکنیم برای دسترسی به اعضای آن که اعضای آن شماره یونیک اعضای ماتریس که شده ۱ رو بهمون میده و اونارو به جای ۱ میکنیم آی در نیهات هم خروجی های تابع رو اوکی میکنیم

برای اینکه صحت کار رو مشخص کنیم اومدیم طبق کد های آماده ای که استاد در سایت قرار داند یه بردر سبز هم برای سگمنت های انتخاب دشده کشیدیم.



در ادامه به نحوه لود دیکشنری و ساخت آن میپردازیم.

```
clc;
clear;
close all;
files=dir('P1_Map Set');
files([1,2])=[];
len=length(files);
TRAIN=cell(2,len);

for i=1:len
    TRAIN{1,i}=imread(['P1_Map Set','\',files(i).name]);
    TRAIN{2,i}=files(i).name(1);
end

save TRAININGSET TRAIN ;
```

در این قسمت اومدیم و ابتدا اون مپ ستیم رو لود کردیم و دوتا المان اولی که به درد نمیخوره رو انداختیم دور در ادامه یه سلول ۲ در تعداد مپ سل تعریف کردیم و اسمشو گذاشتیم TRAIN توی حلقه فور میاییم و اولی میگه که برو عکس رو بخون و بریز توی سلول شماره ۱ و آی و بعدش میگیم حالا اینو name 1 اش رو بهمون بده چون دیگه پسوندش که دیگه به دردمون نمیخوره و در نیهات هم سیو اش کردیم و توی کد اصلی لودش کردیم

در ادامه کد اصلی گفتیم که توتال لتر میشه size(TRAIN,2) تابع سایز این طوری که تعداد سطر و ستون رو برمیگردونه اگر بهش میگفتیم

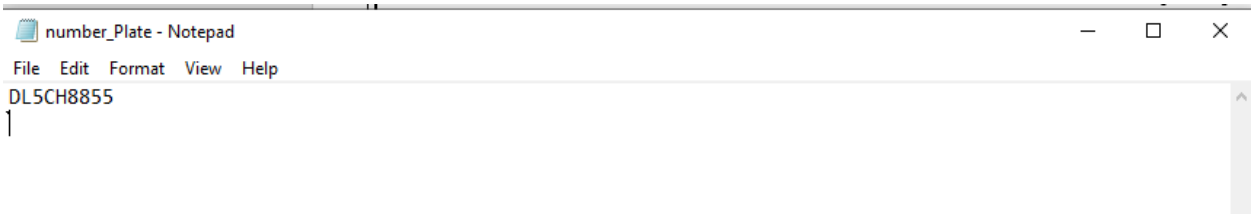
size(TRAIN, 1) ⟶ تعداد سطرها = ۲

size(TRAIN, 2) ⟶ تعداد ستون‌ها = len

در داخل حلقه فور خط مهم خط ۳۸ کد اصلیه

میگیم که برو find کن اون L هایی که مساوی N هستن رو و سطر و ستون هاشونو برگردون به r,c بعد میگیم که Y میشه اون قسمت هایی از clean_image که از کمترین r تا بیشترین r و از کمترین c تا بیشترین c هستن و بعد اینو ریسایز میکنیم به ۴۲ در ۲۴ چون این خوب در میومده کرولیشن اش با اون مپ ستی که داشتیم

در بعدشم که میاییم و دونه دونه کرولیشن میگیریم از هر کدوم از حروفمون با اون سگمنتی که ری سایز کردیم و بعدش میگیم که برو ازش ماکسیموم بگیر دوتا خروجی بهمون میده یکیش مقدار ماکسیمومه یکی دیگش میگه که این ماکسیمومه توی کدوم پوزیشن بوده بعد میگیم اگر مثلا بیشتر از یه ترشهولدی بود بیا این trran(2,position) رو بکن عدد و بهمون خروجی بده و در نهایت هم خروجی بهمون میده و توی یه فایل تکست ذخیره سازی انجام میشه.



بخش دوم:

در بخش دوم مانند بخش اول به جای اینکه یه مپ ست انگلیسی لود کنیم یه مپ ست فارسی لود کردیم تنها گیری که داشت این بود که یه ارور میداد اونم بخاطر این بود که مپ ست کاملا لاجیکال نبود لذا توی کد ابتدا همرو لاجیکال کردم و سپس مثل بخش قبل کار هاش رو انجام دادم. کد بخش دوم به شکل زیر است.

farsi_mapset_genaratior.m | mybinaryfunF.m | p2.m | p3.m

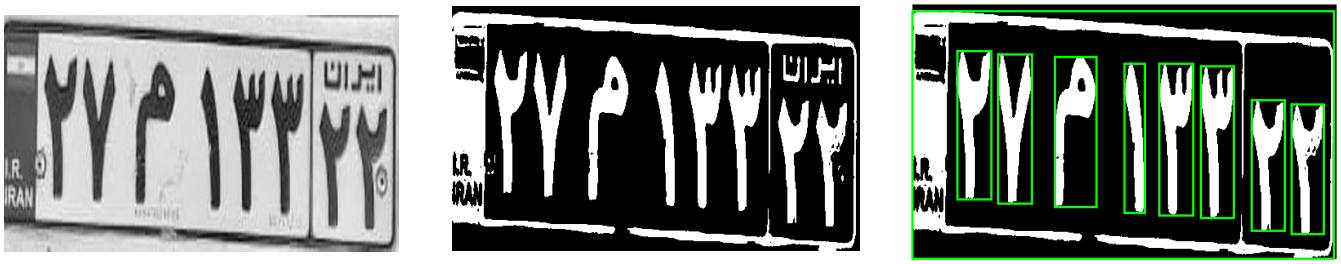
```
files = dir('Map set Farsi');
files([1,2]) = [];
len = length(files);
TRAINF = cell(2, len);

for i = 1:len

    img = imread(['Map set Farsi','\',files(i).name]);


    if size(img,3) == 3
        img = rgb2gray(img);
    end


    img = ~mybinaryfunF(img, 0.4);


    img = imresize(img, [42, 24]);


    TRAINF{1,i} = img;
    TRAINF{2,i} = files(i).name(1);
end

save TRAININGSETF TRAINF;
```

در این قسمت مشابه قسمت اول مپ ست فارسی رو لود کردیم و اومدیم اون وسط چک کردیم که اگر عکس رنگیه بیاد و بکنش خاکستری چون اگر آی جی بی نباشه و این دستور رو بهش بزنیم ارور میده و بعدش هم مثل قبلیه.

```
[file,path] = uigetfile({'*.jpg;*.bmp;*.png;*.tif'},'Choose an image');
s = [path,file];

picture = imread(s);
picture = imresize(picture,[300 500]);

gray_image = mygrayfun(picture);

wb = mybinaryfun(gray_image, 0.4); % tresh hold must be some value between 0 and 1

clean_image = myremoveobject(wb, 500);


[L , Ne] = mysegmentation(clean_image);

propied=regionprops(L,'BoundingBox');
hold on
for n=1:Ne
    rectangle('Position',propied(n).BoundingBox,'EdgeColor','g','LineWidth',2)
end
hold off

load TRAININGSETF;
totalLetters=size(TRAINF,2);

```

```
final_output=[];
t=[];

for n=1:Ne

    if n==7
       saeed=1;
    end


    [r,c]=find(L==n);
    Y=clean_image(min(r):max(r),min(c):max(c));
    Y=imresize(Y,[42,24]);



    ro=zeros(1,totalLetters);
    for k=1:totalLetters
        ro(k)=corr2(TRAINF{1,k},Y);
    end

    [MAXRO,pos]=max(ro);
    if MAXRO>.45
        out=cell2mat(TRAINF(2,pos));
        final_output=[final_output out];
    end


end



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

خروجی بخش دوم:



number_Plate - Notepad

File Edit Format View Help

27M13322

بخش سوم:

در این بخش ابتدا باید بریم و پلاک رو پیدا کنیم وقتی که پلاک رو پیدا کردیم که بقیه اش میشه مثل بخش دوم.

ایده ی پیدا کردن پلاک:

ابتدا اومدم قسمت های آبی تصویر رو جدا کردم بعد از اینکه جداشون کردم یه تمپلیت پیدا کردم که بتونم باهاش کرولیشن بگیرم اومدم و قسمت های آبی رو ریسایز کردم اندازه ی اون تمپلیم و بعدش باهاش کرولیشن گرفتم و اونی که بیشتری بود رو پیدا کردم. این کلیت ایده ی پیدا کردن قسمت آبی پلاک بعد که این پیدا شد تقریبا دیدم که ۱۰ برابر عرض این رو ادامه بدیم میشه کل پلاک و بعد هم یه ذره ماریجین گذاشتم که بالاخره ممکنه تصویر صاف صاف بهمون نرسیده باشه یه ۲۰ تا پیکسل و در نهایت عکس خام پلاک رو بهمون داد.

```
1 -    [file,path] = uigetfile({'*.jpg;*.bmp;*.png;*.tif'},'Choose an image');
2 -    s = [path,file];
3
4 -    img = imread(s);
5
6 -    imgHSV = rgb2hsv(img);
7 -    H = imgHSV(:,:,1);
8 -    S = imgHSV(:,:,2);
9 -    V = imgHSV(:,:,3);
10
11 -   blueMask = (H > 0.55 & H < 0.75) & (S> 0.4) & (V > 0.2);
12 -   blueMask = bwareaopen(blueMask , 50);
```

در این قسمت ابتدا فایل رو میخونیم و بعدش با میاییم و رنگ های هر کدوم رو میگیریم توی بلوماسک اون بازه ای که رنگ های آبی رو جدا میکنیم و بعدش با bwareaopen میاییم و یه ذره تر و تمیزش میکنیم.

در ادامه که در تصویر زیر عکس کد آمده

خط ۱۳ اومدیم و قسمت هایی که به هم چسبیده است رو جدا کردیم با دستور bwconncomp که به توضیح آن پرداختیم

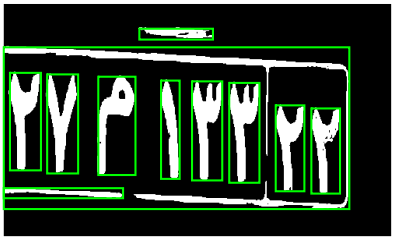
و یه تمپلیت رو در خط ۱۴ لود کردیم و بعدش توی حلقه فور میاییم از اون قسمت ها رو ریسایز میکنیم و به خاکستری تبدیلش میکنیم که بشه کرولیشن گرفت و بعد دابل میکنیم برای کرولیسشت و دونه دونه کرولیشن میگیریم و در نهایت هم خارج از حلقه ی فور میاییم و ماکسیموم اش رو به دست میاریم توی خط ۴۷ اومدیم از ind2sub استفاده کردیم که به جای شناسه یکتا هر کدوم از دیتا ها بهمون سطر و ستونشو بده و بعد دیگه اون قسمت آبی پلاک پیدا شد بعد که پیدا شده توی خط ۶۶ اومدیم کراپ کردیم پلاک رو از صفحه که یه ۱۰ تای عرض پلاک اون طرف تر و یه مارجینی هم برای ارتفاعش و بعدشم که دیگه میشه مثل بخش ۲

```
CC = bwconncomp(blueMask);
template = imread('bluestrip.png');

scores = zeros(1, CC.NumObjects);

for k = 1:CC.NumObjects

    pixels = CC.PixelIdxList{k};
    [r, c] = ind2sub(size(blueMask), pixels);

    ymin = min(r); ymax = max(r);
    xmin = min(c); xmax = max(c);


    cropImg = img(ymin:ymax, xmin:xmax, :);


    cropResized = imresize(cropImg, [size(template,1), size(template,2)]);

    tGray = rgb2gray(template);
    cGray = rgb2gray(cropResized);


    score = corr2(double(tGray), double(cGray));

    scores(k) = score;

end


[bestScore, bestIdx] = max(scores);


pixels = CC.PixelIdxList{bestIdx};
[r, c] = ind2sub(size(blueMask), pixels);

ymin = min(r); ymax = max(r);
xmin = min(c); xmax = max(c);

figure(1);
subplot(1,2,1);
imshow(img);
```

```
figure(1);
subplot(1,2,1);
imshow(img);
hold on;

blueStripBox = [xmin ymin xmax-xmin ymax-ymin];

rectangle('Position', blueStripBox, ...
          'EdgeColor', 'red', ...
          'LineWidth', 2);
hold off;



onlyPelak = img(ymin-20:ymax+20 , xmax:xmax+10*(xmax-xmin) , :);
subplot(1,2,2);

imshow(onlyPelak);

picture = onlyPelak;
```

نتایج آن یه مقداری خطا داره که همون که استاد فرموند با توجه به اینکه کاری که ما داریم انجام میدیم خیلی ساده شده اونچیزیه که در واقع با هوش مصنوعی و ... انجام میشه نباید توقع بیش از اندازه هم داشت.

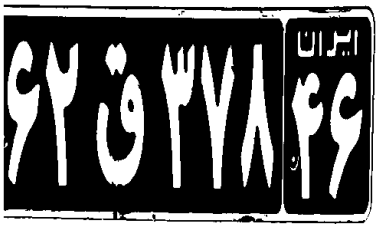
نتایج آن به شکل زیر است.



number_Plate - Notepad

File Edit Format View Help

27M913322



number_Plate - Notepad

File Edit Format View Help

62Y037846

بخش ۴

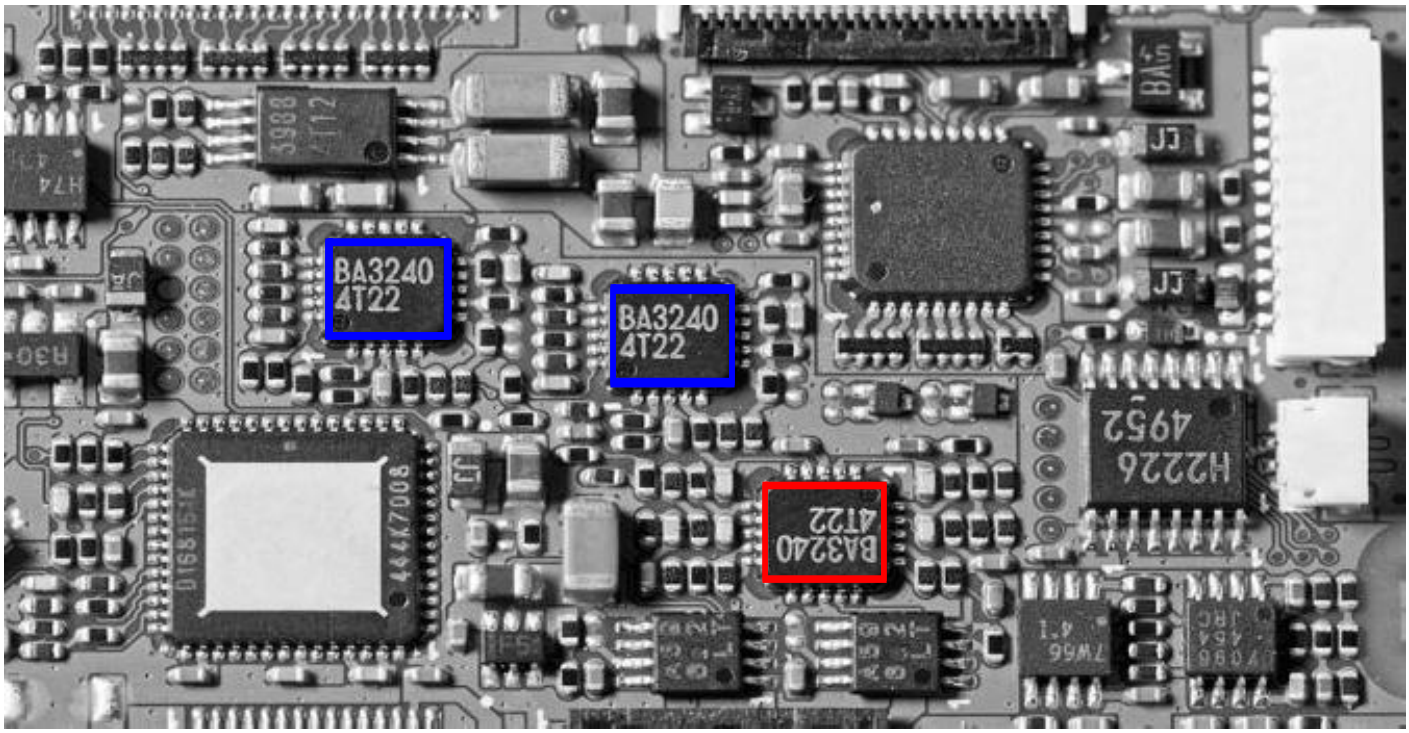
```
function ICrecognition(icImage, pcbImage)



if ndims(icImage) == 3
    icImage = rgb2gray(icImage);
end
if ndims(pcbImage) == 3
    pcbImage = rgb2gray(pcbImage);
end


template = icImage;
templateRot = imrotate(template, 180);


[ht, wt] = size(template);


corr0  = normxcorr2(template, pcbImage);
corr180 = normxcorr2(templateRot, pcbImage);

threshold = 0.7;


[y0, x0] = find(corr0 >= threshold);
[y180, x180] = find(corr180 >= threshold);
figure;
imshow(pcbImage);
hold on;


for i = 1:length(x0)
    xTopLeft = x0(i) - wt;
    yTopLeft = y0(i) - ht;

    rectangle('Position', [xTopLeft, yTopLeft, wt, ht], ...
        'EdgeColor', 'b', 'LineWidth', 2);
end


for i = 1:length(x180)
    xTopLeft = x180(i) - wt;
    yTopLeft = y180(i) - ht;

    rectangle('Position', [xTopLeft, yTopLeft, wt, ht], ...
        'EdgeColor', 'r', 'LineWidth', 2);
end

end
```

توضیح کد:

ابتدا برسی میکنیم که آیا تصویر آر جی بیه یا نه اگر بود میاییم و تبدیل اش میکنیم به خاکستری و بعدش دو تا تمبلیت در نظر میگیریم یکی که خود تمپلیتمون یکی دیگه اش ۱۸۰ درجه چرخیده ی تمپلیتمون بعد میاییم با استفاده از دستور normxcorr2 ازش کرولیشن میگیریم توی این دستور در وروردی اون تمپلتی که میخواییم با تصویر کلی کرولیشن بگیریم رو میگیره ازمون تفاوتی که این دستور داره چون تقسیم میشه به نورم اون دیگه اندازه ها مهم نیست که دقیقا یکسان باشه و تفاوتش با کرولشن عادی اینجاست ما توی بخش های قبلی حتما قبلش اینا رو میومدیم و یه سایز میکردیم ولی دیگه اینجا اصلا نباید بکنیم بکنیم کار خراب میشه اتفاقا توی بخش قبلی یکی دیگه از روش هایی که میشد برمیم و اون قسمت آبی پلاک رو پیدا کنیم همین دستور بود که میومدیم و با کل تصویر کرولشن میگرفتیم من این کارو کردم ولی یه

مقداری خطا داشت برای همین از تکنیکی که در بخش قبل گفتم استفاده کردم قسمت های ابی رو جدا کردم و بعدش اومدیم هم سایزشون کردم و بعد اونا رو کرولیشن گرفتم با تمپلیت

در ادامه ی کد هم نکته ی خاصی وجود نداره فقط اونی که ۱۸۰ درجه چرخیده رو با رنگ قرمز نشون دادم و اونایی که عادیه رو با رنگ آبی نشون دادم و توی کد اصلی هم این فاکشن رو صدا زدم عکس ها رو هم قبلش لود کردم.

```
ic = imread('IC.png');
pcb = imread('PCB.jpg');

ICrecognition(ic, pcb);

```

نتایج به شکل زیر است.