UNIVERSITY OF TEHRAN

signals and systems

CA2
Spring 2024

Amir shahang

810101448

بخش اول :

در سلول selecting the test data با استفاده از دستور uigetfile یک پنجره جدید باز میکنیم تا از کاربر آدرس فایل تصویر پلاک اتومبیل را بپرسد
با استفاده از تابع اماده imresize ابعا تصویر را 500*300 می کنیم

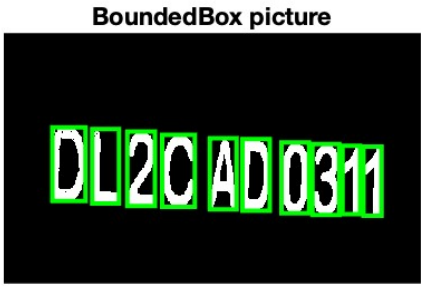
در سلول Rgb2grey برای کاهش پیچیدگی محاسبات تصویر را خاکستری میکنیم
تابع mygreyfun این کار را با حرکت روی پیکسل های عکس و ضرب کردن آن در ضریب های داده شده انجام میدهد .
در تابع mygreyfun ابتدا عکس را double میکنیم تا داده های آن در بازه [0,1] قرار بگیرند و بعد خاکستری کردن آن با دستور unit8 عکس را از نوع unit8 تبدیل میکنیم به طوری که داده های آن در بازه ی [0,255] قرار میگیرد.

در سلول conversion to a binary image تصویر را باینری میکنیم . این کار را در تابع mybinaryfun به سادگی با تعیین یک threshold انجام میدهیم:
در تابع عکس و threshold به عنوان ورودی گرفته و پیکسل های عکس بررسی میکنیم ,اگر از threshold بزرگ تر بود آن را 1 و در غیر اینصورت آن را 0 میکنیم. در نهایت در خروجی عکس باینری شده را تحویل میگیریم.

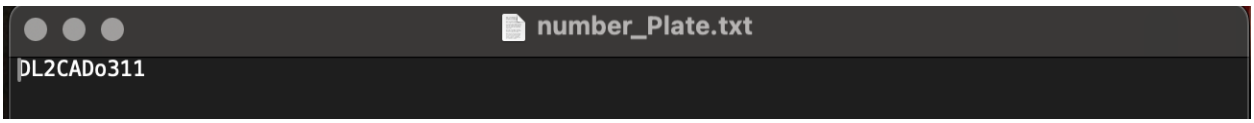
نگاشت باینری بدست آمده در بخش قبل کمی نویزی است و نیاز است آبجکت های بدون مفهوم با سایز کوچک حذف شوند.

در سلول Removing the small objects با استفاده از تابع myremovecom این کار را انجام میدهیم . این تابع عکس و یک عدد n ورودی میگیرد و کاموپننت های متصل بهم کمتر از n pixel را حذف میکند
ابتدا با استفاده از دستور bwconncomp کامپوننت های متصل بهم را پیدا کرده و با numobjects تعداد پیکسل های آن را پیدا میکنیم و در ادامه بررسی میکنیم اگر تعداد آن از n کمتر بود آن را حذف میکنیم.
این کار را برای یک n بزرگ مثلا 2500 نیز انجام داده تا backgroundای از عکس بدست بیاید سپس عکس را از background کم میکنیم تا همه ی نویز ها حذف شوند

در سلول labeling connected component دور کامپوننت های مشخص شده با خط سبزخط میکشیم
در این سلول از تابع mysegmentation استفاده میکنیم .
در این تابع با استفاده از دستور bwconncomp کامپوننت های متصل بهم را پیدا کرده و با استفاده ازدستور lablematrix هر کامپوننت را شماره گذاری میکنیم
با استفاده از regionprops برای کامپوننت های شماره گذاری شده در L یک BoundingBox در نظر گرفته میشود و در ادامه با استفاده از یک حلقه و دستورBoundingBox, rectangle هارا با رنگ سبز خط کشی میکنیم.



و درآخر در سلول Loading the mapset مپ ست train شده در فایل training_loading.m را فراخوانی میکنیم و در ادامه بررسی میکنیم که آیا کامپوننت های پیدا شده از پلاک با مپ ست ما همخوانی دارد یا خیر . برای این کار بین کامپوننت های پیدا شده از پلاک و مپ ست correlation میگیریم و اگر pick آن از 0.45 بیشتر حرف آن بدرستی تشخیص داده شده و سپس آن را در ارایه out میریزیم و سپس آن را در یک note چاپ میکنیم



Mygreyfunction :

```
function picture = mygrayfun(picture)

    picture = double(picture);

    redChannel = picture(:,:,1);
    greenChannel = picture(:,:,2);
    blueChannel = picture(:,:,3);

    picture = 0.299 * redChannel + 0.578 *  greenChannel + 0.114 * blueChannel;

    picture = uint8(picture);

end
```

Mybinaryfunction :

```
function image = mybinaryfun(picture,thr)

    image = ~(picture >= thr);
end
```

Myremovecomfunction :

```
function result = Myremovecom(picture, n)

    components = bwconncomp(picture);
    num = components.NumObjects;

    result = zeros(size(picture));

    for i = 1:num

        component = false(size(picture));
        component(components.PixelIdxList{i}) = true;

        if sum(component(:)) >= n
            result = result | component;
        end
    end
end
```

Mysegmentationfunction :

```
function [L,Ne] = mysegmentation(picture)

    figure
    imshow(picture)

    component = bwconncomp(picture);
    L = labelmatrix(component);
    Ne = component.NumObjects;

    propied=regionprops(L,'BoundingBox');

    hold on

    for n=1:size(propied,1)
    rectangle('Position',propied(n).BoundingBox,'EdgeColor','g','LineWidth',2)
    end

    hold off

end
```

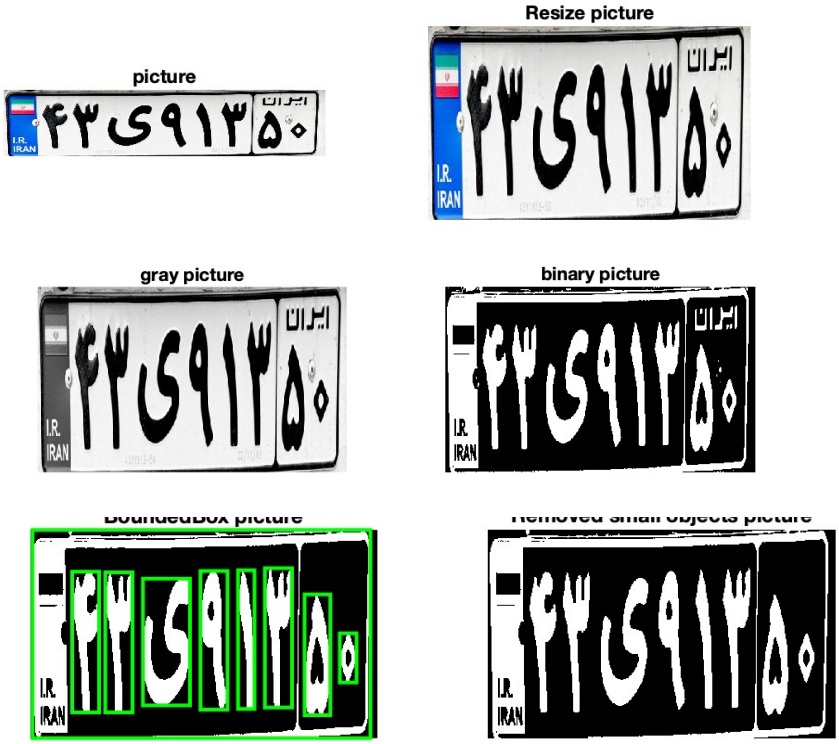
P1.m :

```
clc
close all;
clear;

%% SELECTING THE TEST DATA

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


figure
subplot(1,2,1)
imshow(picture)
title('picture')

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

subplot(1,2,2)
imshow(picture)
title('Resize picture')

%% RGB2GRAY

picture = mygrayfun(picture);
figure
subplot(1,2,1)
imshow(picture)
title('gray picture')


%% CONVERSION TO A BINARY IMAGE

picture = mybinaryfun(picture,0.3*255) ;

subplot(1,2,2)
imshow(picture)
title('binary picture')


%% Removing the small objects

 picture = Myremovecom(picture,500);


background = Myremovecom(picture,2500);

picture2 = picture-background;
figure
subplot(1,2,1)
```

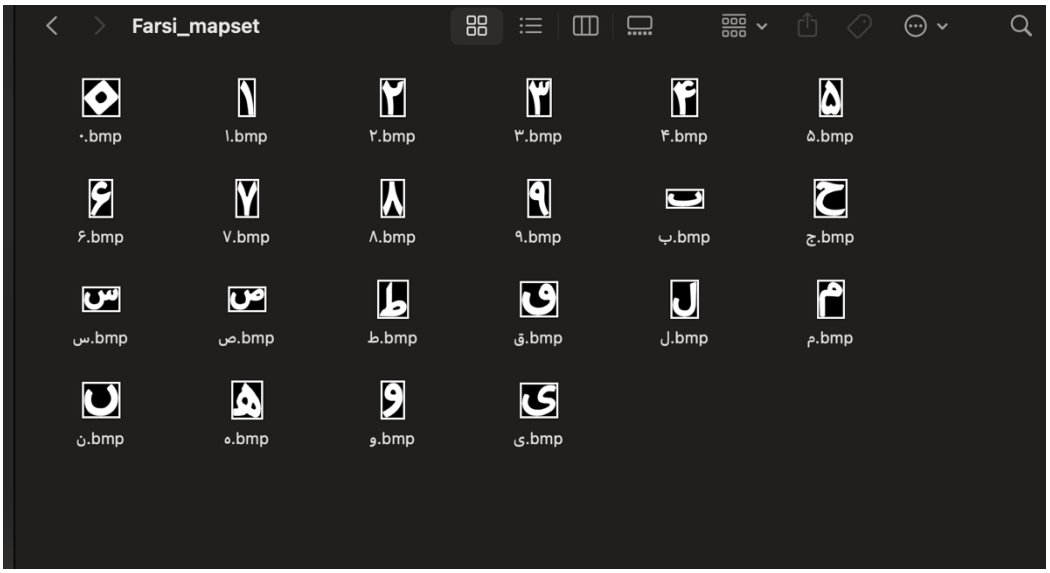
```
imshow(picture2)
title('BoundedBox picture')


%% Labeling connected components

[L,Ne] = mysegmentation(picture2);
subplot(1,2,2)
imshow(picture2)
title('Removed small objects picture')

%% Loading the mapset

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


figure
final_output=[];
t=[];
for n=1:Ne

    [r,c]=find(L==n);
    Y=picture2(min(r):max(r),min(c):max(c));
    imshow(Y)
    Y=imresize(Y,[42,24]);
    imshow(Y)
    pause(0.2)
    if n==7
        hg=1;
    end




    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

%% Printing the plate
file = fopen('number_Plate.txt', 'wt');
fprintf(file,'%s\n',final_output);
fclose(file);
system('open number_Plate.txt');
```

بخش دوم :

تمام توضیحات این بخش مانند بخش قبلی است فقط با این تفاوت که در این قسمت مپ ست فارسی train می کنیم.



دیتابیس فارسی طراحی شده برای این بخش:



P2.m :

```
clc
close all;
clear;
%% SELECTING THE TEST DATA
[file,path]=uigetfile({'*.jpg;*.bmp;*.png;*.tif'},'Choose an image');
s=[path,file];
picture=imread(s);

figure
subplot(1,2,1)
imshow(picture)
title('picture')

%% resize
picture=imresize(picture,[300 500]);

subplot(1,2,2)
imshow(picture)
title('Resize picture')

%% RGB2GRAY
picture = mygrayfun(picture);

figure
subplot(1,2,1)
imshow(picture)
title('gray picture')

%% CONVERSION TO A BINARY IMAGE

threshold = graythresh(picture);
picture =~imbinarize(picture,threshold);

subplot(1,2,2)
imshow(picture)
title('binary picture')

%% Removing the small objects

picture = Myremovecom(picture,500);

picture2 = picture ; %-background;
figure
subplot(1,2,1)
imshow(picture2)
title('BoundedBox picture')


%% Labeling connected components

[L,Ne] = mysegmentation(picture2);
subplot(1,2,2)
```

```
imshow(picture2)
title('Removed small objects picture')

%% Loading the mapset
load TRAININGSET3;
totalLetters=size(TRAIN,2);


figure
final_output=[];
t=[];

for n=1:Ne
    [r,c]=find(L==n);
    Y=picture2(min(r):max(r),min(c):max(c));
    imshow(Y)
    Y=imresize(Y,[100,80]);
    imshow(Y)
    pause(0.2)
    if n==7
        hg=1;
    end
ro=zeros(1,totalLetters);
    for k=1:totalLetters
        ro(k)=corr2(TRAIN{1,k},Y);
    end
[MAXRO,pos]=max(ro);
     if MAXRO>.7

    out=cell2mat(TRAIN(2,pos));
    final_output=[final_output out];
     end
 end



%% Printing the plate
file = fopen('number_PlateF.txt', 'wt', 'n', 'UTF-8');
fprintf(file,'%s\n',final_output);
fclose(file);
system('open number_PlateF.txt');
clc
close all;
```

خروجی چاپ شده در نوت :

بخش سوم :

در این قسمت با استفاده از روش tamplate matching روی نوار آبی سمت چپ پلاک ها پلاک را از بقیه اجزای تصویر جدا میکنیم
ابتدا تابعی برای محاسبه ی همبستگی بین tamplate و تصویر در فضای رنگی RGB مینویسیم

```
function [corr_mix, corrMax, bbox] = templateMatching(template, pic)

    RC = normxcorr2(template(:, :, 1), pic(:, :, 1));
    GC = normxcorr2(template(:, :, 2), pic(:, :, 2));
    BC = normxcorr2(template(:, :, 3), pic(:, :, 3));
    corr_mix = (RC + GC + BC)/3;

    [corrMax, Idx] = max(abs(corr_mix(:)));

    [Y, X] = ind2sub(size(corr_mix), Idx(1));

    offset = [X - size(template, 2), Y - size(template, 1)];

    bbox = [offset(1), offset(2), size(template, 2), size(template, 1)];
end
```

تابع normxcorr2 با استفاده از روش همبستگی نرمال شده دوبعدی میزان شباهت بین قالب و تصویر را در هر سه کانال رنگی قرمز، سبز و آبی به صورت جداگانه محاسبه می‌کند
ازآنجا که cross correlation فقط روی یک چنل عکس انجام میشود سه بار انجام شده سپس، میانگین این سه مقدار همبستگی به عنوان یک نمایش مختلط برای ترکیب اطلاعات رنگی استفاده می‌شود.
بیشترین مقدار همبستگی در کل تصویر به عنوانcorrMax ذخیره میشود
In2sub در متلب برای تبدیل یک شاخص خطی (یک بعدی) به شاخص‌های معادل چندبعدی در یک آرایه یا ماتریس استفاده می‌شود. در زمینه پردازش تصویر یا داده‌های چند بعدی، این تابع به ما اجازه می‌دهد که از یک شماره تکی (اندیس) به مختصات دقیقی که آن شماره در یک ماتریس چندبعدی نشان می‌دهد، برگردیم.
در اخر template مورد نظر در عکس ورودی شناخته شده و با یک offsetای دور قاب پلاک یک Box درست میکنیم.

بدلیل آنکه شاید کاربر در ورودی عکس قاب پلاک را بدهد یک template بزرگ تر نیز نیاز داریم

در سلول setup and image loading عکس جلوبندی ماشین را از ورودی گرفته و در با. دستور imread آن را میخوانیم

در سلول Preprocessing and Template Matching با دستور imread template ها را خوانده و با استفاده از تابع TemplateMatchin برای هر Template correlation میگیریم و در انتها Templateای که correlation آن بیشتر است را استفاده میکنیم .

سلول Adjusting Bounding Box and Display :

این بخش از کد مربوط به تنظیم موقعیت و اندازه‌ی Boxهای محدودکننده برای نشان دادن ناحیه پلاک از تصویر اصلی است که با Template مشخص شده همبستگی بالایی دارد
ابتدا، با استفاده از نسبت تغییر اندازه و حاشیه خطا، ابعاد Box محدودکننده محاسبه و تنظیم می‌شود.
سپس، یک Box محدودکننده دیگر بر اساس نسبت مشخص شده بین قسمت آبی و پلاک تنظیم می‌شود که عرض آن با این نسبت تغییر می‌کند
تصویر اصلی در یک پنجره نمایش داده شده و هر دو Box محدودکننده با رنگ‌ها و ضخامت‌های خط متفاوت رسم می‌شوند تا مکان تطابق الگو (نوار آبی پلاک) مشخص شود در صورتی که میزان همبستگی بالاتر از 0.5 باشد، عنوان 'Match Success' به نمایش درمی‌آید که نشان‌دهنده موفقیت در یافتن تطابق است

سلول Image Cropping and Resizing :

این بخش از کد برای برش دادن بخشی از تصویر اصلی بر اساس جعبه محدودکننده‌ای که قبلاً تعیین شده است، طراحی شده. در صورتی که Box محدودکننده خالی نباشد (یعنی الگویی با همبستگی بالا پیدا شده باشد)، بخش مربوطه از تصویر اصلی با استفاده تابع imcrop بر اساس ابعاد مشخص شده درbounding_box بریده و در متغیر Croppedpicture ذخیره می‌شود در غیر این صورت یک پیام خطا نمایش داده می‌شود که اعلام می‌کند هیچ جعبه محدودکننده‌ای یافت نشده یا همبستگی پایین است و بنابراین امکان برش تصویر وجود ندارد. در نهایت، تصویر بریده شده در یک زیرپنجره نمایش داده می‌شود تا کاربر بتواند نتیجه برش را مشاهده کند

حال که قاب پلاک از عکس اولیه آن جدا شده است بقیه مراحل مانند سوال قبل می باشد!

picture

Resize picture

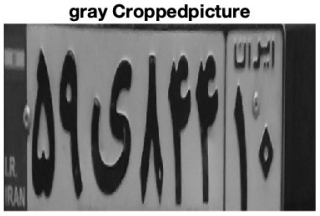
picture

Match Success

Croppedpicture

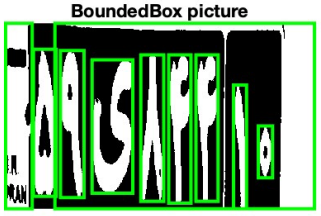
خروجی ادامه ی کد که مانند سوال قبل میباشد:

gray Croppedpicture



binary Croppedpicture

BoundedBox picture



Removed small objects picture

number_PlateF.txt

۹۱۳۵۰ی۴۳

P3.m :

```
clc
close all;
clear;
%% Setup and Image Loading

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


%% Preprocessing and Template Matching

ERR_MARGIN = 10;
BLUE2PLATE_RATIO = 14;

    bluestrip = imread('templateMatching1.png');
    bluestrip_big = imread('templateMatching.png');

    picture = imresize(picture_full, [NaN, 800]);
         figure
        subplot(1,2,1)
        imshow(picture)
        title('picture')

    ratio = size(picture_full, 1) / size(picture, 1);

    [corr_mix, corr_max, bbox] = templateMatching(bluestrip, picture);
    [corr_mixB, corr_maxB, bboxB] = templateMatching(bluestrip_big, picture);


     if corr_maxB > corr_max

         [corr_mix, corr_max, bbox] = deal(corr_mixB, corr_maxB, bboxB);
     end

%% Adjusting Bounding Box and Display

    left = round(bbox(1) * ratio) - ERR_MARGIN;
    top = round(bbox(2) * ratio) - ERR_MARGIN;
    width = round(bbox(3) * ratio) + 2 * ERR_MARGIN;
    height = round(bbox(4) * ratio) + 2 * ERR_MARGIN;

    bbox_full = [left,top,width,height];

    bounding_box = bbox_full;
    bounding_box(3) = BLUE2PLATE_RATIO * bbox(3) * ratio;

         subplot(1,2,2)
        imshow(picture_full)
        title('Resize picture')


        hold on

        rectangle('Position', bbox_full, 'edgecolor', 'k', 'linewidth', 2);
```

```
        rectangle('Position', bounding_box, 'edgecolor', 'g', 'linewidth',
1);

        if corr_max > 0.5
                title('Match Success')
        else
            disp('not Match')

        end
%% Image Cropping and Resizing

if ~isempty(bounding_box)
    Croppedpicture = imcrop(picture_full, bounding_box);
    figure
     subplot(1,2,1)
    imshow(Croppedpicture)
     title('Croppedpicture')

else
    disp('No bounding box found or correlation too low, cannot crop the
image.');
    Croppedpicture = [];
end
    subplot(2,2,2)
       imshow(Croppedpicture)
%% p2.m

picture=imresize(Croppedpicture,[300 500]);


picture = mygrayfun(picture);
figure
subplot(1,2,1)
imshow(picture)
title('gray Croppedpicture')

threshold = graythresh(picture);
picture =~imbinarize(picture,threshold);

subplot(1,2,2)
imshow(picture)
title('binary Croppedpicture')



picture = Myremovecom(picture,500);

picture2 = picture ; %-background;
figure
subplot(1,2,1)
imshow(picture2)
title('BoundedBox picture')


[L,Ne] = mysegmentation(picture2);
```

```
subplot(1,2,2)
imshow(picture2)
title('Removed small objects picture')


load TRAININGSET3;
totalLetters=size(TRAIN,2);


figure
final_output=[];
t=[];

for n=1:Ne
    [r,c]=find(L==n);
    Y=picture2(min(r):max(r),min(c):max(c));
    imshow(Y)
    Y=imresize(Y,[100,80]);
    imshow(Y)
    pause(0.2)
    if n==7
        hg=1;
    end
ro=zeros(1,totalLetters);
    for k=1:totalLetters
        ro(k)=corr2(TRAIN{1,k},Y);
    end
[MAXRO,pos]=max(ro);
     if MAXRO>.7

    out=cell2mat(TRAIN(2,pos));
    final_output=[final_output out];
     end
 end

file = fopen('number_Plate3.txt', 'wt', 'n', 'UTF-8');
fprintf(file,'%s\n',final_output);
fclose(file);
system('open number_Plate3.txt');

%%
% corr_mix  the average correlation map across the RGB channels;
% corr_max  the maximum correlation value
% bbox      he bounding box coordinates of where the template matches in the
larger image
function [corr_mix, corrMax, bbox] = templateMatching(template, pic)

    RC = normxcorr2(template(:, :, 1), pic(:, :, 1));
    GC = normxcorr2(template(:, :, 2), pic(:, :, 2));
    BC = normxcorr2(template(:, :, 3), pic(:, :, 3));
    corr_mix = (RC + GC + BC)/3;

    [corrMax, Idx] = max(abs(corr_mix(:)));

    [Y, X] = ind2sub(size(corr_mix), Idx(1));
```

```
    offset = [X - size(template, 2), Y - size(template, 1)];

    bbox = [offset(1), offset(2), size(template, 2), size(template, 1)];
end
```