بسمه‌تعالی

دانشکده مهندسی برق و کامپیوتر

پروژه تشخیص اسکناس

درس تصویرپردازی رقمی

استاد: جناب آقای دکتر یغمایی

تدریس‌یار: آقای مهندس شکری

ارائه توسط:

محمدسعید قاسمی

۹۹۱۱۹۲۰۰۱۵

بهار ۱۴۰۱

پروژه تشخیص اسکناس

فایل اصلی:

https://github.com/semnan-university-ai/image-processing-class-002/blob/main/project/msg67/README.md

در این پروژه تصویر اسکناس به برنامه داده می‌شود، و برنامه ما باید تشخیص درستی از مقدار پول به وسیله دیتابیس از قبل آماده خود بدهد، در حقیقت باید ورودی را در دسته درست دسته بندی کند. مثل همه برنامه‌ها در این کلاس، بازهم برنامه با ۳ عبارت زیر جهت پاک کردن حافظه، workspaceو متغییرها شروع می‌شود:

```
clc;

clear;

close all;
```

یک رابط کاربری جهت ورود اطلاعات باز می‌شود uigetfile در قسمت بعد، بوسیله:

```
 [filename,path] = uigetfile({'*.jpg';'*.jpeg';},...

                    'Select an Image File');


  img=imread([path filename]);
```

بعد از import شدن یک عکس JPG یا JPEG به برنامه، بوسیله یک تابع به قسمت اصلی کد وارد می‌شویم:

در ابتدای تابع و بعد از مقداردهی‌های اولیه، **فولدر** فایل‌های اسکناس‌های اصلی (دیتابیس) را مشخص می‌کنیم، تا همه فایل‌های داخل آن را که با .jpg تمام شده‌است، خوانده شود:

```
folder='train';

dirImg=dir(fullfile(folder,'*.jpg'));
```

سپس با این دستور بررسی می‌کنیم که تصویر ورودی(تابع) در راستای درستی می‌باشد یا خیر. در نهایت اگر نبود آن را rotate می‌کند:

```
if(size(input_img,1)>size(input_img,2))

   check_img=imresize(input_img,[NaN 295]);

else

   check_img=imresize(input_img,[295 NaN]);

end
```

با حلقه for زیر، ابتدا یکی یکی فایل‌های فولدر train که ابتدای فانکشن مقدار دهی شده بود، خوانده می‌شوند، سپس کانال‌های رنگی‌شان جدا شده و میانگین intensity کلی هر رنگ حساب می‌شود. سپس همین کار برای ورودی انجام می‌شود. این مقادیر را ازهم کم می‌کنیم، اگر قدرمطلق اخلاف رنگ ها ار ۱۵۰ کمتر بود وارد شرط می شود و اگر نبود این حلقه به ازای تصویر اصلی جدید تکرار می شود. در صورتی که شرط درست باشد ابتدا تصویر را خاکستری میکنیم سپس با استفاده از الگوریتم KAZE که پایداری بالایی در برابر تغییرات روشنایی دارد، نقاط متناظر بین دو تصویر را استخراج می کنیم و نتیجه را در متغییری ذخیره

می‌کنیم. مجموع نقاط مشترک در allMatchPoint ذخیره می‌شود تا بیشترین اشتراک به عنوان جواب انتخاب شود) :ابتدای کار من می‌خواستم با میانگین گیری از چند قسمت از تصاویر، و بررسی با فایل‌های دیتابیس کل مسئله را حل کنم، اما با چند تست متوجه شدم اعداد بعضی اسکناس‌ها حتی در ۳کانال رنگی، خیلی بهم نزدیک هستند، و با توجه به عکس‌برداری در شرایط گوناگون، حتی با وجود Histogram Equalization ، شدت روشنایی‌ها از قاعده خیلی خاصی پیروی نمی‌کنند(.

```
for k=1:numel(dirImg)
    filename=fullfile(folder,dirImg(k).name);
    trainImg=imread(filename);



    a=imresize(trainImg,imSize); %Input image is resized here
    a1=a(:,:,1); %Red component
    a2=mean2(a1); %Mean of Red component
    a3=a(:,:,2); %Green component
    a4=mean2(a3); %Mean of Green component
    a5=a(:,:,3); %Blue component
    a6=mean2(a5); %Mean of Blue component



    % ////////////////////////////////


    I1=check_img(:,:,1);
    I2=mean2(I1);
    I3=check_img(:,:,2);
    I4=mean2(I3);
    I5=check_img(:,:,3);
    I6=mean2(I5);



    if(abs(a2-I2) < 150 && abs(a4-I4) < 150 && abs(a6-I6) < 150)
```

```
        a=rgb2gray(a);
        b=rgb2gray(check_img);

        apoints = detectKAZEFeatures(a,'Diffusion','region');
        bpoints = detectKAZEFeatures(b,'Diffusion','region');


        [features1,valid_points1] = extractFeatures(a,apoints);
        [features2,valid_points2] = extractFeatures(b,bpoints);


        [indexPairs matchMetric] = matchFeatures(features1, features2) ;

        matchedPoints1 = valid_points1(indexPairs(:,1),:);
        matchedPoints2 = valid_points2(indexPairs(:,2),:);


        allMatchPoint=[allMatchPoint size(matchedPoints1,1)];
        allFileName=[allFileName filename];


    end


end
```

برای دقت بیشتر پس از هر با بررسی با کل تصاویر اصلی ، تصویر ورودی را ۹۰ درجه میچرخانیم و عملیات قبل را تکرار میکنیم این کار ٤ بار انجام می شود.

در قسمت‌های بعد، بیشتر کار برای رسم نمودار نتیجه انجام شده است:

با استفاده از بیشترین نقاط اشتراک، تصویر انتخاب شده از دیتابیس ابتدا خاکستری می‌شود، سپس دوباره نقاط اشتراک استخراج می‌شوند، و در آخر نیز به وسیله str2double و با آرگومان '\d+\' ابتدا نام رو دابل و سپس مقادیر غیر عددی رو از نام حذف کردیم تا با استفاده از نامی که برای فایل‌های دیتابیس بکار برده شده است، مقدار اسکناس ورودی نوشته می‌شود:

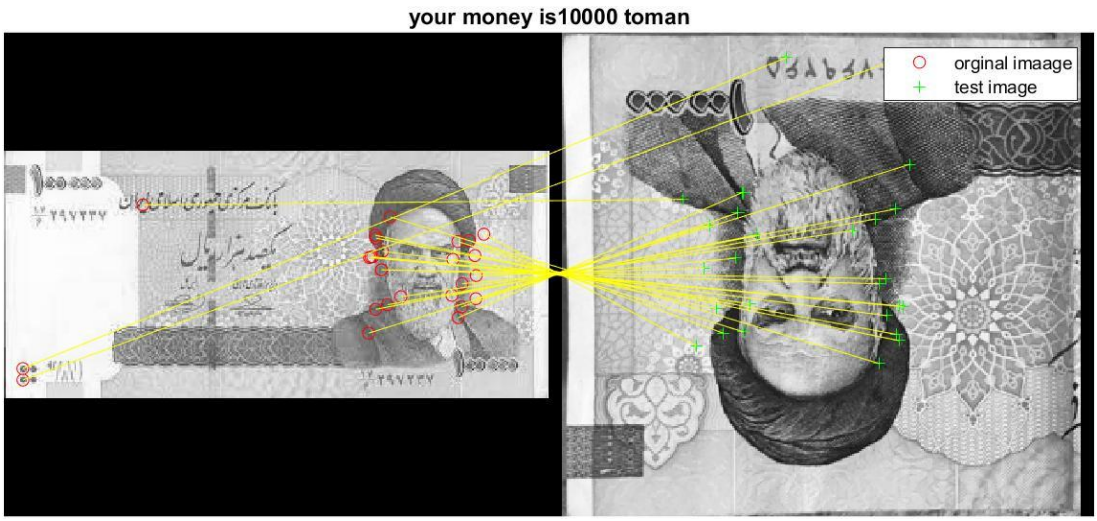
(تابع strcat جهت نوشتن چند مورد در plot استفاده می‌شود .legend نیز برای نوشتن شرح عکس در کنار تصویر استفاده می‌شود)

```
[~,index]=max(allMatchPoint);


cashAmount =str2double(regexp(allFileName{index}, '\d+', 'match'));



a=imread(allFileName{index});
a=im2gray(a);



apoints = detectKAZEFeatures(a,'Diffusion','region');
bpoints = detectKAZEFeatures(b,'Diffusion','region');



[features1,valid_points1] = extractFeatures(a,apoints);
[features2,valid_points2] = extractFeatures(b,bpoints);


[indexPairs matchMetric] = matchFeatures(features1, features2) ;


matchedPoints1 = valid_points1(indexPairs(:,1),:);
matchedPoints2 = valid_points2(indexPairs(:,2),:);


figure; ax = axes;
showMatchedFeatures(a,b,matchedPoints1,matchedPoints2,'montage','Parent',ax);
title(ax,strcat('your money is  ',string(cashAmount),' toman'));
legend(ax, 'orginal imaage','test image');
```

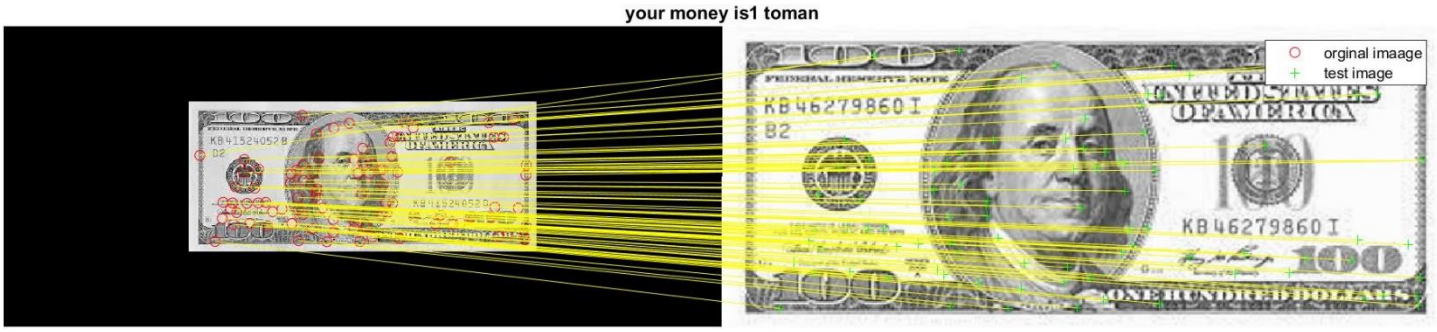
end

از معایب، مزایا و مشکلات این کار می‌توان به موارد زیر اشاره کرد:

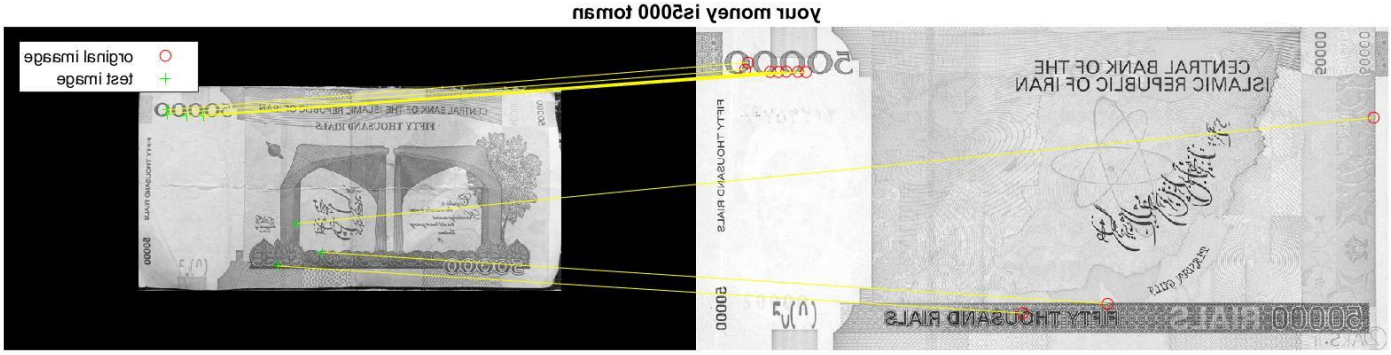
- از کارهای جالب این کد، میتوان به تشخیص اسکناس نصفه و چرخانده شده اشاره کرد :



- مزیت دیگر وابسته نبودن به اعداد روی اسکناس می‌باشد، بنابراین استفاده از پول‌های غیر ایرانی را نیز ممکن می‌سازد :



- از مشکلات این کار می‌توان به کم شدن سرعت، در صورت تعدد فایل‌های دیتابیس اشاره کرد. (با توجه به تعدد اسکناس‌های موجود، این کار اجتناب‌ناپذیر می‌باشد، مثلا در عکس پایین دو نمونه ۵هزار تومانی آورده شده است، البته به طور جالبی سیستم متوجه شباهت ها شده، اما در اکثر تست‌های من این اتفاق نمی‌افتاد)



این تمرین با کمک گرفتن از تمرین خانم صفاریان و با کمی تغییر نوشته شده است...