

بسمه تعالی

موضوع

تشخیص اسکناس با استفاده از رنگ آنها

درس

تصویر پردازی رقمی (پردازش تصویر)

استاد

دکتر فرزین یغمایی

TA

آقای امیر شکری

گردآورنده

حوا عسکری

ش.د: 40011920006

مقدمه

تشخیص اسکناس یکی از موضوعات جهانی برای تقلبی یا واقعی بودن آن است. در اینجا ما با استفاده از رنگ های اسکناس و فاصله رنگ ها و تست و مقایسه آنها با میانگین فاصله رنگ های داده های آموزشی سعی در تشخیص اسکناس داریم. ما برای هر نوع اسکناس بصورت جدا گانه آموزش را انجام میدهم و به صورت جداگانه آنها را تست میکنیم.

برنامه

برای این برنامه ما یک تابع `ColorDistance()` ، یک تابع `MeanRGB()` و چهار تابع `Train` برای هر یک از اسکناس ها (1000-2000-5000-10000) داریم.

تابع `ColorDistance()` برای بدست آوردن اختلاف رنگ های هر اسکناس تست و میانگین رنگ های اسکناس های آموزشی می باشد. در این تابع از رابطه فاصله اقلیدسی استفاده کرده ایم (دلیل استفاده : باعث مثبت شدن جواب ها می شود) :

```
function [CD] = ColorDistance(r1,g1,b1, r2,g2,b2)
CD = sqrt(double((r1 - r2).^2 + (g1 - g2).^2 + (b1 - b2).^2));
end
```

در تابع `MeanRGB()` همانطور که از اسم تابع مشخص است برای بدست آوردن میانگین مقادیر کانال های رنگ های اسکناس ها می باشد :

```
function [Rm ,Gm ,Bm] = MeanRGB(img)

imgs = size(img);
Rs = 0;
Gs = 0;
Bs = 0;

n = 1;

for i = 1: imgs(1)
    for j = 1: imgs(2)
        Rs =Rs + double(img(i,j,1));
        Gs =Gs + double(img(i,j,2));
        Bs =Bs + double(img(i,j,3));
        n = n+1;
    end
end

Rm = round(Rs/n);
Gm = round(Gs/n);
```

```
Bm = round(Bs/n);  
end
```

در توابع آموزش ما (Train1000 تا Train1000) فایل های تصاویر اسکناس موجود را فراخوانی میکنیم ،
در این تابع تابع میانگین رنگ ها را فراخوانی میکنیم ، آنها را بدست می آوریم سپس میانگین میانگین ها را در
متغیر RGB می گذاریم و مینیمم و ماکسیمم هر کانال رنگ را استخراج می کنیم:

```
function [test_matrix, RGB, maxRGB, minRGB] = Train10000()  
  
root = '../train/10.000/';% '../train/5000/'% '../train/2000/'% '../train/1000/'  
path = '*.jpg';  
money = dir([root path]);  
lenm = max(size(money));  
  
test_matrix = zeros(lenm , 3);  
  
for i = 1: lenm  
  
    img = imread([root money(i).name]);  
  
    [Rm ,Gm ,Bm] = MeanRGB(img);  
  
    test_matrix(i,1) = Rm;  
    test_matrix(i,2) = Gm;  
    test_matrix(i,3) = Bm;  
  
    RGB = round(mean(test_matrix, 1));  
  
    maxRGB = max(test_matrix);  
    minRGB = min(test_matrix);  
  
end  
  
end
```

و در نهایت تابع Main() که تابع اصلی برای تست برنامه است قرار دارد که تصاویر مربوطه را از پوشه تست
می خواند و آنها را مقایسه میکند و نتیجه را بصورت تصویر اسکناس با برچسب مبلغ آن (اگر تشخیص درست
باشد) یا برچسب عدم انطباق نمایش می دهد :

```
clc;  
clear;  
close all;  
%%  
root = '../test/10/';% '../test/5/'% '../test/2/'% '../test/1/'  
path = '*.jpg';  
money = dir([root path]);  
len = max(size(money));  
%%  
[test_matrix, RGB, maxRGB, minRGB] = Train10000();  
%%
```

```

for i = 1 : len
    img = imread([root money(i).name]);
    imges = size(img);
    [Rm ,Gm ,Bm] = MeanRGB(img);
    for l = 1: imges(1)
        for k = 1 : imges(2)
            R1 = double(img(l,k,1));
            G1 = double(img(l,k,2));
            B1 = double(img(l,k,3));
        end
    end

    temp = double(round(ColorDistance(R1, G1, B1, RGB(1, 1), RGB(1, 2),
    RGB(1, 3))));
    if (minRGB <= temp & temp <= maxRGB)
        figure, imshow(uint8(img)), title('10000 Toman');
    else
        figure, imshow(uint8(img)), title('No Match');
    end
end
end

```

بعضی از نتایج به صورت زیر می باشد:



همانطور که میبینید بعضی ها را درست تشخیص داده است و بعضی را نادرست.

با توجه به نتایج به دست آمده متوجه میشویم که فقط با داشتن میانگین رنگ ها و تفاوتشان نمی توان نتیجه بهینه داشت . برای نتیج بهتر میتوان از فیلتر ها ، حوزه فرکانس ، و حتی کانال های رنگی دیگر مانند HSI و استفاده نمود.

با سپاس از توجه شما.