

**به نام خدا**

دانشکده مهندسی برق و کامپیوتر

استاد: دکتر فرزین یغمایی

دانشجو: امید غلامی

مقطع: کارشناسی ارشد

رشته: هوش مصنوعی

درس: پردازش تصویر رقمی

گزارش کار تشخیص اسکناس با استفاده از رنگ

**هدف پروژه**

شناسایی نوع اسکناس با استفاده از رنگ

**شرح مراحل کار**

1. خواندن عکس از ورودی
2. انجام پیش پردازش بر روی آن

الف)تبدیل عکس های عمودی به افقی

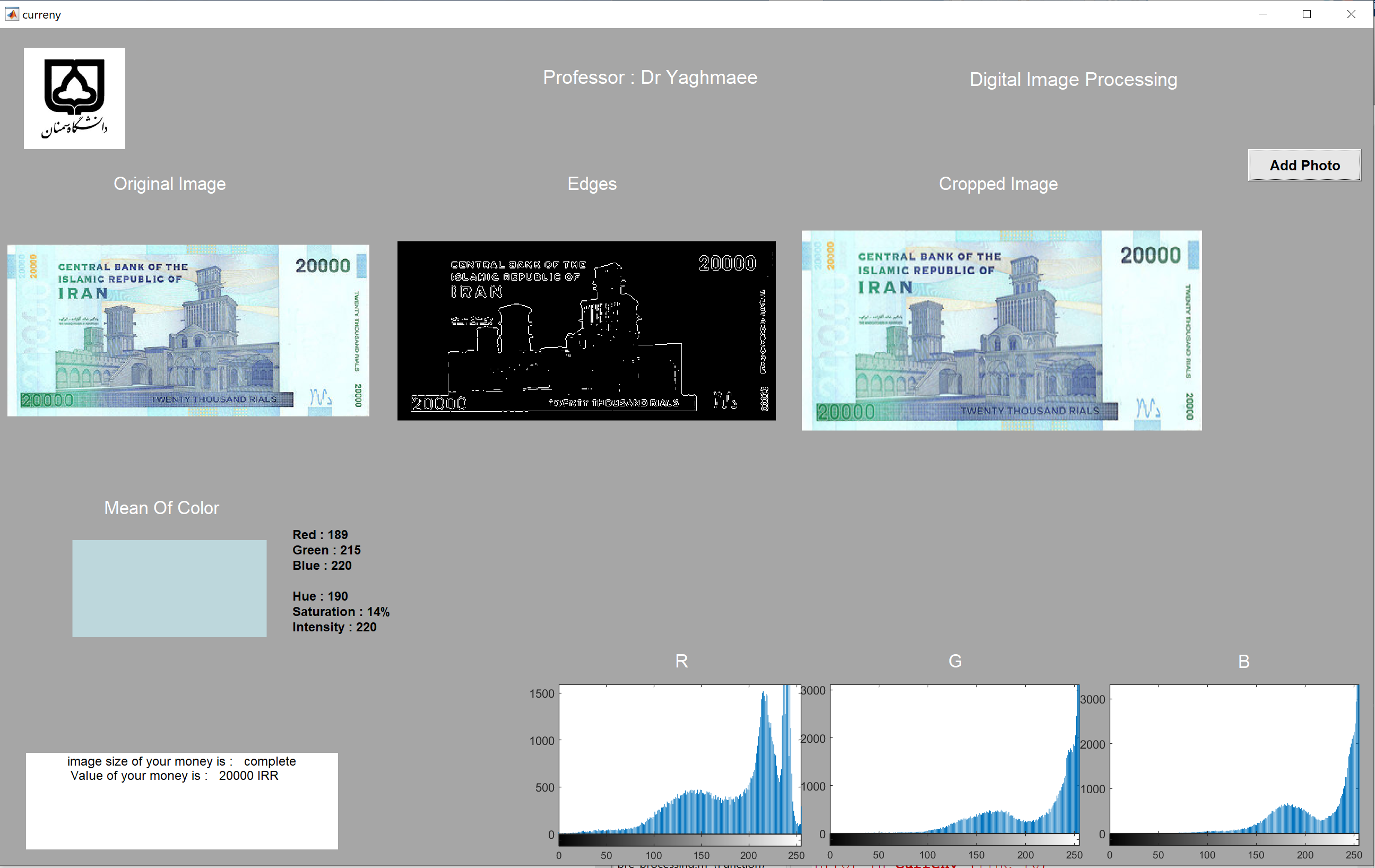
ب)بریدن پس زمینه ی اسکناس ها

پ)هم سایز کردن اسکناس ها

ت)متعادل سازی روشنایی اسکناس ها

1. تفکیک کانال های رنگی
2. استفاده از فضای رنگی HSI و RGB
3. استفاده تابع همبستگی (correlation) برای شناسایی الگو ها

**نمایه نمایی از نرم افزار تشخیص اسکناس**



**شروع کار**

ابتدا تابع CURRENCY.fig را اجرا می کنیم.

با کلیک و انتخاب دکمه Add Photo کار پردازش تصاویر آغاز میشود .

در ابتدا با استفاده از تابع uigetfile فایل از ورودی خوانده می شود

[imname,impath]=uigetfile({'\*.jpg;\*.jpeg;\*.png'});

imgPath =strcat(impath,imname);

img = imread(imgPath);

سپس اگر عکس بصورت عمودی باشد آن را می چرخانیم.

[height ,width ,k] = size(img);

if height > width % if is it portraint ,rotate to landscape

img =imrotate(img,90);

end

نمایش هیستوگرام کانال های رنگی تصویر :

axes(handles.axes2);imshow(img);

axes(handles.axes4);imhist(img(:,:,1));

axes(handles.axes5);imhist(img(:,:,2));

axes(handles.axes11);imhist(img(:,:,3));

**تشخیص و بررسی اندازه اسکناس :**

در این قسمت با استفاده از تابع check\_size و با استفاده از اندازه تصویر اصلی تشخیص می دهیم که آیا این پول کامل است یا نصفه و یا اینکه پس زمینه دارد یا نه.

check= check\_size(img);

check="image size of your money is : " + check+newline;

**تابع اصلی check\_size**  :

ورودی : تصویر اصلی

خروجی : متنی که نشان دهنده ی نوع تصویر می باشد.

function check = check\_size(img)

[height ,width,k] =size(img);

if width < 1.5\*height

if 1.1\*height> width

check= 'half';

else

check='hasBackground';

end

else

check ='complete';

end

end

**نمایش لبه ها و عکس پیش پردازش شده در رابط کاربری**:

edges=edge(rgb2gray(img),'sobel');

axes(handles.axes9);

imshow(edges);

img=pre\_processing(img);

axes(handles.axes10);

imshow(img);

تابع pre\_processing :

ورودی : تصویر اصلی

خروجی : تصویر پیش پردازش شده

در ابتدا با استفاده از تابع check\_size و با توجه به ابعاد تصویر بررسی میکنیم در صورتی که عکس دارای پس زمینه باشد ،تابع بریدن پس زمینه را اجرا می کنیم.

function image\_out = pre\_processing(img)

check=check\_size(img);

if strcmpi(check,'hasBackground')

img=crop\_background(img);

end

img=imresize(img,[200,400]); %resize image

img=balance\_color(img);

image\_out =img;

end

**تابع حذف پس زمینه** :

function image = crop\_background(img,name)

imgGray=rgb2gray(img);

% find edges

edges=edge(imgGray,'canny',0.4);

binaryImage = bwareafilt(edges, 1);

[rows, columns] = find(binaryImage);

row1 = min(rows);

row2 = max(rows);

col1 = min(columns);

col2 = max(columns);

[height\_img,width\_img,k]=size(img);

w=col2-col1;

h =row2-row1;

if w/h >4

h= floor(((w/h)\*h)/1.9);

row2 =h+row1;

end

if w < 1.6\*h

if width\_img\*0.35 >col1

col2=col2+w;

else

col1=col1-w;

end

end

if width\_img/w >5

if width\_img\*0.4 >col1

col2=col2+2\*w;

else

col1=col1-2\*w;

end

end

if height\_img/h >5

if height\_img\*0.4 >row1

row2=row2+2\*h;

else

row1=row1-2\*h;

end

end

%Crop

cropped = img(row1:row2, col1:col2, :);

[height,width,k]=size(cropped);

if height >97

image=cropped;

else

image =img;

end

%imshow(image);

end

در این تابع مراحل کار به صورت زیر انجام می شود :

1. ابتدا عکس ورودی به عکس خاکستری تبدیل می شود.
2. با استفاده از ابزار لبه یابی متلب و با استفاده از متد canny و همچنین threshold =0.4 لبه ها را پیدا می کنیم.
3. با استفاده از تابع bwareafilt شی را از عکس جدا می کنیم.
4. با استفاده از ابزار find طول ها و عرض ها را در تصویر بدست می آوریم.
5. بیشترین مقدار و کمترین مقدار را به عنوان طول و عرض قرار می دهیم
6. با کمک آرایه ها عکس را کراپ می کنیم.

\* در بعضی مواقع عکس ها بصورت مربعی و یا خیلی نازک کراپ می شدند که برای حل این مشکل چند شرط در این تابع قرار دادیم

**تغییر سایز اسکناس ها** :

ورودی : تصویر با ابعاد مختلف

خروجی : تصویر با ابعاد 200\*400

ابتدا باید ابعاد تمام عکس های ورودی یک اندازه داشته باشند تا بتوان به یک نتیجه ی مطلوب در پردازش تصویر رسید . برای همین عکس ها ورودی را به ابعاد200 \*400 تبدیل میکنیم

img=imresize(img,[200,400]);

**متعادل سازی رنگ و روشنایی :**

ورودی : تصویر

خروجی : تصویر

بعضی از عکس های ورودی بسیار تیره بودن و مناسب پردازش نبود. برای متعادل سازی عکس های ورودی ، عکس هایی که روشنایی خیلی کم و یا خیلی زیاد دارند را با استفاده از تابع balance\_color افزایش می هیم.

function out = balance\_color(img,name)

R =img(:,:,1);

G =img(:,:,2);

B =img(:,:,3);

if mean(R(:)) <150 && mean(G(:)) <150 && mean(B(:)) <150

out =img+30;

elseif mean(R(:)) >200 && mean(G(:)) >200 && mean(B(:)) >200

out = img-20;

else

out =img;

end

**تابع processing ( تابع اصلی برنامه برای پردازش تصویر اسکناس):**

ورودی : تصویر

خروجی : ارزش پول به تومان بصورت متن

مراحل کار پردازش تصویر به صورت زیر می باشد

داده های آموزشی مربوط به هر اسکناس (شامل الگوهایی از کوشه های هر نوع اسکناس می باشد) فراخوانی میشود

%training data lists

[train500,train1000,train2000,train5000,train10000,train50000]=load\_train();

**تابع بارگذاری داده های آموزشی**

ورودی : ندارد(فقط برای تمیز نوشتن کد ها و جلوگیری از تکرار استفاده می شود)

خروجی :یک آرایه شامل لیست داده های آموزشی

function [train500,train1000,train2000,train5000,train10000,train50000] = load\_train()

train500 =[dir('train/500/\*.jpg');dir('train/500/\*.jpeg');dir('train/500/\*.png')];

train1000 =[dir('train/1000/\*.jpg');dir('train/1000/\*.jpeg');dir('train/1000/\*.png')];

train2000 =[dir('train/2000/\*.jpg');dir('train/2000/\*.jpeg');dir('train/2000/\*.png')];

train5000 =[dir('train/5000/\*.jpg');dir('train/5000/\*.jpeg');dir('train/5000/\*.png')];

train10000 =[dir('train/10000/\*.jpg');dir('train/10000/\*.jpeg');dir('train/10000/\*.png')];

train50000 =[dir('train/50000/\*.jpg');dir('train/50000/\*.jpeg');dir('train/50000/\*.png')];

end

سپس میانگین رنگ های RGBرا حساب کرده و آن را به مدل رنگی HIS تبدیل می کنیم.

[meanR,meanG,meanB]=mean\_color(img);

HSV = rgb2hsv(meanR,meanG,meanB);

h=round(HSV(:,:,1)\*360);

**تابع mean\_color**

ورودی : تصویر

خروجی : میانگین رنگ ها هر کانال قرار داده شده در یک آرایه

function [R\_out,G\_out,B\_out] = mean\_color(img)

R = img(:,:,1);

G = img(:,:,2);

B = img(:,:,3);

meanR=floor(mean(R(:)));

meanG=floor(mean(G(:)));

meanB=floor(mean(B(:)));

R\_out =meanR;

G\_out =meanG;

B\_out =meanB;

End

**مراحل پردازش تصویر به صورت زیر انجام می شود.**

1. ابتدا با استفاده از هم شرط رنگ) برحسب ترکیب مدل رنگی HSI و (RGB و همچنین با استفاده از تابع همبستگی (تشخیص گوشه های اسکناس ) پول ها را ارزش گذاری میکنیم.
2. در صوتی بر آورده نشدن شرط ها ،فقط با استفاده از رنگ اسکناس ها را ارزش گذاری می کنیم.
3. در صورت دسته بندی نشدن در گام دوم، با استفاده از تابع همبستگی و انطباق نمونه ها بر روی عکس ، اسکناس ها را تشخیص می دهیم.
4. در صورتی که هیچ کدام از این سه مرحله نتوانند پول را به خوبی تشخیص دهند. ارزش پول به عنوان شناخته نشده معرفی خواهد شد.

if h > 114 & h <193 & meanR>130 & max(temp\_matching(train1000,'1000',img))>0.7

Value= '1000';

elseif h > 189 & h < 231 & meanB>160 & max(temp\_matching(train2000,'2000',img))>0.6

Value= '2000';

elseif h > 44 & h <86 & meanR <210 & max(temp\_matching(train10000,'10000',img))>0.6

Value= '10000';

elseif h > 31 & h < 44 & max(temp\_matching(train5000,'5000',img))>0.7

Value= '5000';

elseif h > 5 & h <35

if max(temp\_matching(train500,'500',img))>0.7 & meanR >160

Value= '500';

elseif max(temp\_matching(train50000,'50000',img))>0.6

Value= '50000';

else

Value ='not\_recognized';

end

else

Value =check\_by\_color();

if strcmpi(Value, 'not\_recognized')

Value =catByNumber(img);

end

end

function Value= check\_by\_color()

if h > 114 & h <193 & meanR>130

Value= '1000';

elseif h > 189 & h < 231 & meanB>160

Value= '2000';

elseif h > 31 & h < 44

Value= '5000';

elseif h > 49 & h <60 & meanR <210

Value= '10000';

else

Value ='not\_recognized';

end

end

end

**تابع catByNumber**

ورودی : تصویر

خروجی :ارزش پول به تومان بصورت متن

تمام الگوهای موجود در پوشه داده های آموزشی را بر روی پول اعمال می کند و با استفاده از بیشترین مقداری که از تابع temp\_matching بر میگرداند ارزش پول ها را انتخاب میکند.

function Value = catByNumber(img)

% training data lists

[train500,train1000,train2000,train5000,train10000,train50000]=load\_train();

%dw100 = difference with 100 toman

%Calculate template matching

dw500=max(temp\_matching(train500,'500',img));

dw1000=max(temp\_matching(train1000,'1000',img));

dw2000=max(temp\_matching(train2000,'2000',img));

dw5000=max(temp\_matching(train5000,'5000',img));

dw10000=max(temp\_matching(train10000,'10000',img));

dw50000=max(temp\_matching(train50000,'50000',img));

%We get the max from each and

% then we drop all of them within one array

values=[dw500,dw1000,dw2000,dw5000,dw10000,dw50000];

values=sort(values);

maxValue=max(values);

if maxValue==dw500

Value='500';

elseif maxValue==dw1000

Value='1000';

elseif maxValue==dw2000

Value='2000';

elseif maxValue==dw5000

Value='5000';

elseif maxValue==dw10000

Value ='10000';

elseif maxValue==dw50000

Value ='50000';

else

Value ='not\_recognized';

end

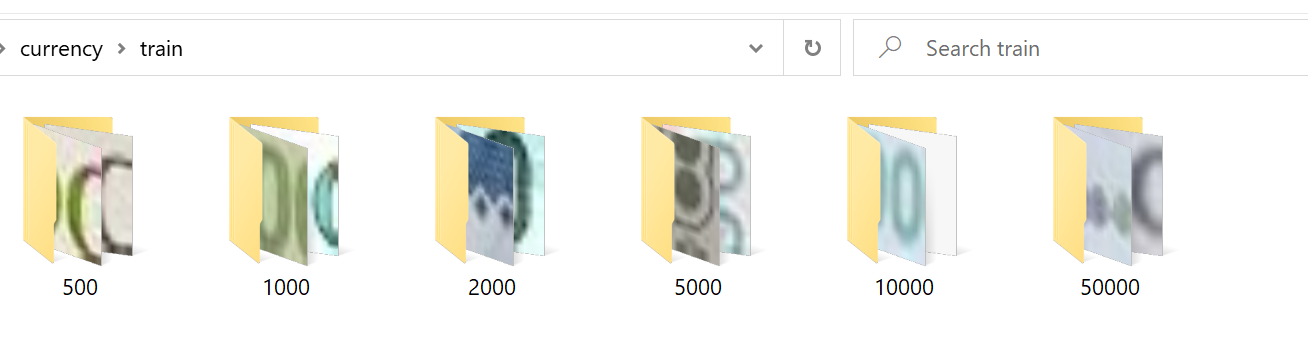
end

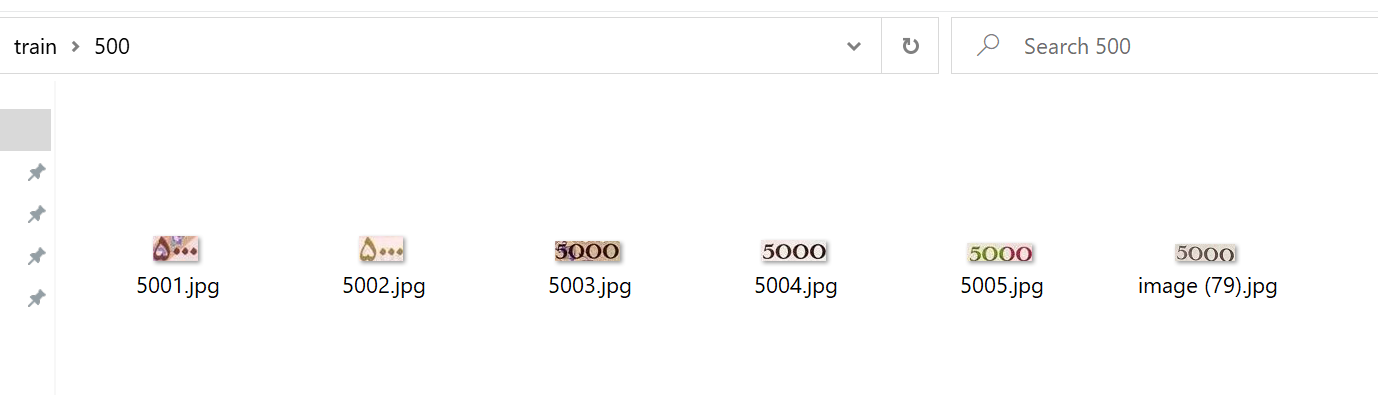
**تابع temp\_matching**

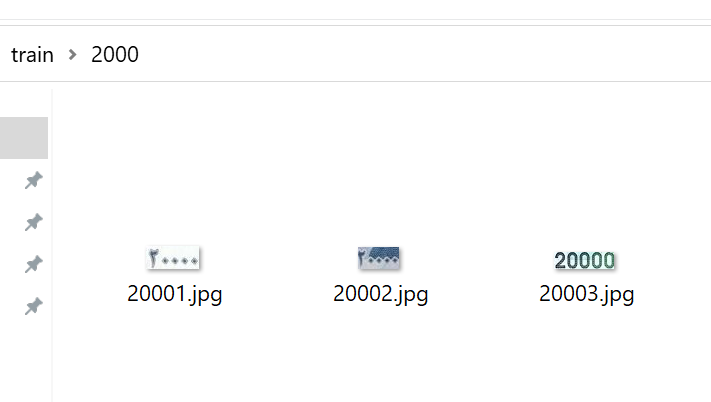
ورودی ها : لیست فایل های داده های آموزشی با توجه به ارزش هر اسکناس ، نام پوشه در داده های آموزشی، تصویر

خروجی : یک آرایه که شامل مقادیر بررسی تمام نمونه های موجود در پوشه ی داده های آموزشی هر کدام از اسکناس ها بر روی تصویر ورودی با استفاده از تابع همبستگی (مقادیر اعدادی بین 0 و یک می باشند).

نمونه ای از پوشه های داده های آموزشی

****

****

****

**تابع :**

function corrs = temp\_matching(templates,folder,img)

img=rgb2gray(img);

values=[];

for i=1:length(templates)

name =templates(i).name;

%load image from train folder

template=imread(strcat('train/',folder,'/',name));

template=rgb2gray(template);

[height, width, k]=size(template);

templateWidth = width;

templateHeight = height;

channelToCorrelate = 1; % Use the red channel.

correlationOutput = normxcorr2(template(:,:,1), img(:,:, channelToCorrelate));

% Find out where the normalized cross correlation image is brightest.

[maxCorrValue, maxIndex] = max(abs(correlationOutput(:)));

[yPeak, xPeak] = ind2sub(size(correlationOutput),maxIndex(1));

% Because cross correlation increases the size of the image,

% we need to shift back to find out where it would be in the original image.

corr\_offset = [(xPeak-size(template,2)) (yPeak-size(template,1))];

boxRect = [corr\_offset(1) corr\_offset(2) templateWidth, templateHeight];

if (boxRect(2) <50 || boxRect(2) > 150) && ...

(boxRect(1) <100 || boxRect(1)>300) ...

&& (boxRect(3)> boxRect(4)) && maxCorrValue >= 0.6 && boxRect(4) <=50 ...

&& boxRect(1) > 0 && boxRect(2) > 0

% axis on; % Show tick marks giving pixels

% hold on; % Don't allow rectangle to blow away image.

% rectangle('position', boxRect, 'edgecolor', 'g', 'linewidth',2);

values = [values,maxCorrValue];

end

end

% % imshow(img);

corrs=values;

end

**نتایج :**

**با توجه به ارزیابی های انجام شده و تست بر بیش از 223 عکس از همه نوع اسکناس ، 31 عکس را به عنوان ناشناخته معرفی کرده و از 192 عکس باقی مانده تعداد درست ها برای هر دسته حداکثر 93 درصد و حداقل84 درصد بوده است.**