



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

(۱) خواندن عکس از ورودی

(۲) 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

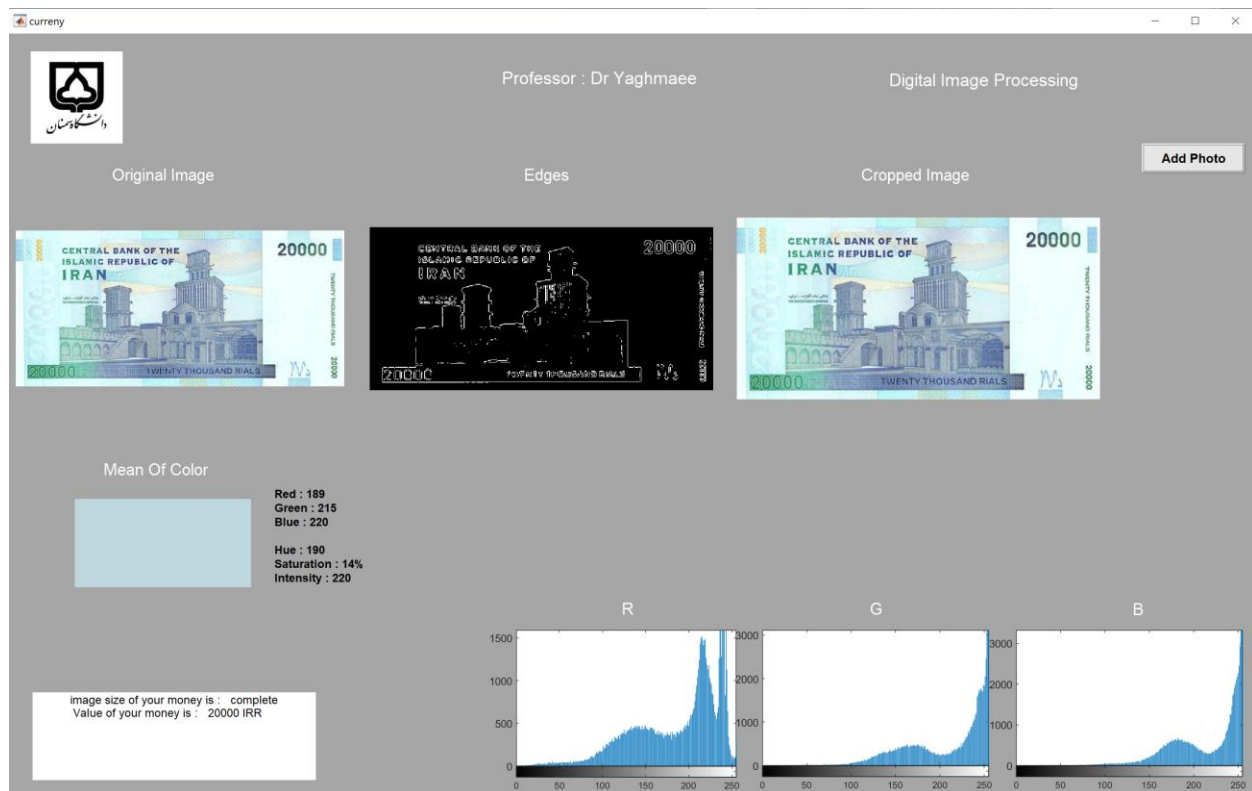
ت) متعادل سازی روشنایی اسکناس ها

(۳) تفکیک کانال های رنگی

(۴) استفاده از فضای رنگی RGB و HSI

(۵) 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,۹۰);  
end
```

نمایش هیستوگرام کانال های رنگی تصویر :

```
axes(handles.axes۲);imshow(img);  
axes(handles.axes۴);imhist(img(:,:,۱));  
axes(handles.axes۵);imhist(img(:,:,۲));  
axes(handles.axes۱۱);imhist(img(:,:,۳));
```

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 < \,۵*height  
        if \,۱*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.axes۹);  
imshow(edges);  
  
img=pre_processing(img);  
axes(handles.axes۱۰);  
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,[۲۰۰,۴۰۰]); %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',۰,۴);
    binaryImage = bwareafilt(edges, ۱);
    [rows, columns] = find(binaryImage);
    row۱ = min(rows);
    row۲ = max(rows);
    col۱ = min(columns);
    col۲ = max(columns);
    [height_img,width_img,k]=size(img);
    w=col۲-col۱;
    h =row۲-row۱;
    if w/h >۴
        h= floor(((w/h)*h)/۱,۹);
        row۲ =h+row۱;
    end
    if w < ۱,۶*h
        if width_img*۰,۳۰ >col۱
            col۲=col۲+w;
        else
            col۱=col۱-w;
```

```

        end
    end

    if width_img/w > 0
        if width_img*0.5 > col1
            col2=col1+2*w;
        else
            col1=col1-2*w;
        end
    end
    if height_img/h > 0
        if height_img*0.5 > row1
            row2=row1+2*h;
        else
            row1=row1-2*h;
        end
    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 :

- (۱) ابتدا عکس ورودی به عکس خاکستری تبدیل می شود.
- (۲) با استفاده از ابزار لبه یابی متلب و با استفاده از متد `canny` و همچنین `threshold = 0.4` لبه ها را پیدا می کنیم.
- (۳) با استفاده از تابع `bwareafilt` شی را از عکس جدا می کنیم.
- (۴) با استفاده از ابزار `find` طول ها و عرض ها را در تصویر بدست می آوریم.
- (۵) بیشترین مقدار و کمترین مقدار را به عنوان طول و عرض قرار می دهیم
- (۶) 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 ۴۰۰*۲۰۰

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 ابعاد ۴۰۰*۲۰۰ تبدیل میکنیم

```
img=imresize(img,[۲۰۰,۴۰۰]);
```

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(:,:,۱);
```

```
G =img(:,:,۲);
```

```
B =img(:,:,۳);
```

```
if mean(R(:)) <۱۵۰ && mean(G(:)) <۱۵۰ && mean(B(:)) <۱۵۰
```

```
    out =img+۳۰;
```

```
elseif mean(R(:)) >۲۰۰ && mean(G(:)) >۲۰۰ && mean(B(:)) >۲۰۰
```

```
    out = img-۲۰;
```

```
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 گوشه های هر نوع اسکناس می باشد)
فراخوانی میشود

```
%training data lists
[train000,train\000,train۲000,train0000,train\0000,train00000]=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
[train000,train\000,train۲000,train0000,train\0000,train00000] =
load_train()
train000
=[dir('train/000/*.jpg');dir('train/000/*.jpeg');dir('train/000/
*.png')];
train\000
=[dir('train/\000/*.jpg');dir('train/\000/*.jpeg');dir('train/\0
00/*.png')];
train۲000
=[dir('train/۲000/*.jpg');dir('train/۲000/*.jpeg');dir('train/۲0
00/*.png')];
train0000
=[dir('train/0000/*.jpg');dir('train/0000/*.jpeg');dir('train/00
00/*.png')];
train\0000
=[dir('train/\0000/*.jpg');dir('train/\0000/*.jpeg');dir('train/
\0000/*.png')];
train00000
=[dir('train/00000/*.jpg');dir('train/00000/*.jpeg');dir('train/
00000/*.png')];
end
```

سپس میانگین رنگ های RGB را 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)*۳۶۰);
```

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(:, :, ۱);  
    G = img(:, :, ۲);  
    B = img(:, :, ۳);  
  
    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.

(۱) 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.

(۲) در صوتی بر آورده نشدن شرط ها ،فقط با استفاده از رنگ اسکناس ها را ارزش گذاری می کنیم.

(۳) 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.

(۴) 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 > ۱۱۴ & h < ۱۹۳ & meanR > ۱۳۰ &
max(temp_matching(train۱۰۰۰, '۱۰۰۰', img)) > ۰, ۷
    Value = '۱۰۰۰';

elseif h > ۱۸۹ & h < ۲۳۱ & meanB > ۱۶۰ &
max(temp_matching(train۲۰۰۰, '۲۰۰۰', img)) > ۰, ۶
    Value = '۲۰۰۰';

elseif h > ۴۴ & h < ۸۶ & meanR < ۲۱۰ &
max(temp_matching(train۱۰۰۰۰, '۱۰۰۰۰', img)) > ۰, ۶
    Value = '۱۰۰۰۰';
elseif h > ۳۱ & h < ۴۴ &
max(temp_matching(train۰۰۰۰, '۰۰۰۰', img)) > ۰, ۷
    Value = '۰۰۰۰';
elseif h > ۰ & h < ۳۰
    if max(temp_matching(train۰۰۰, '۰۰۰', img)) > ۰, ۷ &
meanR > ۱۶۰
        Value = '۰۰۰';
    elseif
max(temp_matching(train۰۰۰۰۰, '۰۰۰۰۰', img)) > ۰, ۶
        Value = '۰۰۰۰۰';
    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 <119 & meanR>13.
            Value= '1...';

        elseif h > 119 & h < 121 & meanB>16.
            Value= '2...';

        elseif h > 121 & h < 122
            Value= '0...';

        elseif h > 129 & h <130 & meanR <11.
            Value= '1.....';

        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
[train000,train1000,train2000,train0000,train10000,train00000]=load_train();

    %dw100 = difference with 100 toman
    %Calculate template matching
    dw000=max(temp_matching(train000,'000',img));
    dw1000=max(temp_matching(train1000,'1000',img));
    dw2000=max(temp_matching(train2000,'2000',img));
    dw0000=max(temp_matching(train0000,'0000',img));
    dw10000=max(temp_matching(train10000,'10000',img));
    dw00000=max(temp_matching(train00000,'00000',img));
    %We get the max from each and
    % then we drop all of them within one array
    values=[dw000,dw1000,dw2000,dw0000,dw10000,dw00000];
    values=sort(values);
    maxValue=max(values);

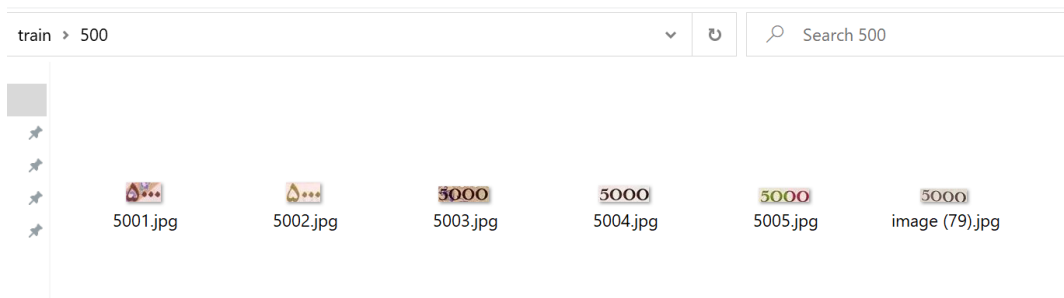
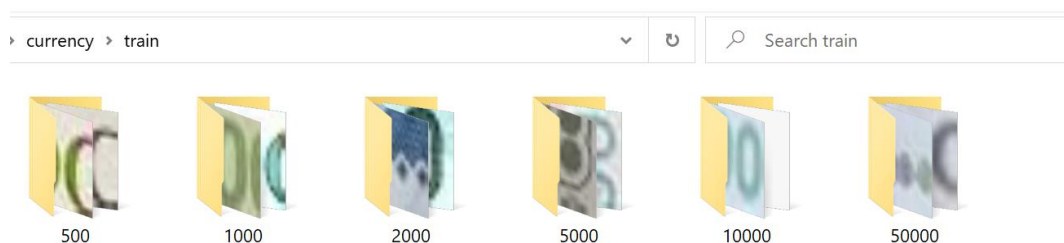
    if maxValue==dw000
        Value='000';
    elseif maxValue==dw1000
        Value='1000';
    elseif maxValue==dw2000
        Value='2000';
    elseif maxValue==dw0000
        Value='0000';
    elseif maxValue==dw10000
        Value='10000';
    elseif maxValue==dw00000
        Value='00000';
    else
        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 ۰ و یک می باشند).

نمونه ای از پوشه های داده های آموزشی



train > 2000



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) < 0.0 || boxRect(2) > 100) && ...
            (boxRect(1) < 100 || boxRect(1) > 200) ...
            && (boxRect(3) > boxRect(4)) && maxCorrValue
>= 0.1 && boxRect(4) <= 0.0 ...
            && 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
end
```

```
% %      imshow(img);  
  
corrs=values;  
  
end
```

نتایج :

با توجه به ارزیابی های انجام شده و تست بر بیش از ۲۲۳ عکس از همه نوع اسکناس ، ۳۱ عکس را به عنوان ناشناخته معرفی کرده و از ۱۹۲ عکس باقی مانده تعداد درست ها برای هر دسته حداکثر ۹۳ درصد و حداقل ۸۴ درصد بوده است.