**基于matlab的图像形状识别**

王昊飞 201820504010

数字图像处理是一种通过计算机采用一定的算法对图像进行处理的计数。数字图像处理技术已经在各个领域上都有了比较广泛的应用。图像处理的信息量很大，对处理速度的要求也比较高。Matlab强大的运算和图像展示功能，是图像处理便的更加简单和直观。本文介绍了如何利用matlab来进行数字图像形状的边缘提取和识别，来实现对图片中的圆形进行识别。

**图像识别的步骤**

输入图像

图像预处理

图形特征提取

图像分割

图像分类

图像处理

图像识别

**1图像处理**

**1.1图像预处理**

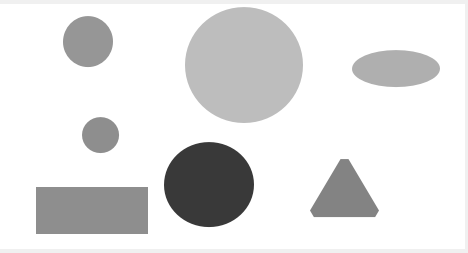
首先将原始图像转化成灰度图像，采用rgb2gray函数。

I = imread('E:\360downloads\28.png');

figure;imshow(I);

bw = rgb2gray(I);

figure;imshow(bw);



灰度图像

再将灰度图像转化为二值图像，用im2bw函数和graythresh函数

bw = im2bw(bw,graythresh(bw));



二值图像

再对图像进行增强，在进行形态学处理。



预处理后的图像

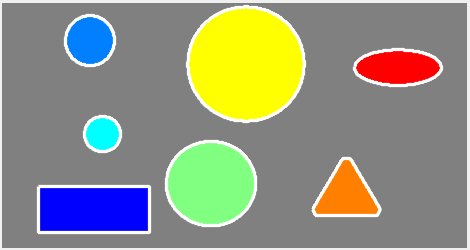
**1.2图像分割**

采用bwboundaries函数来对图像进行边缘检测

[B,L] = bwboundaries(bw,'noholes');

figure;imshow(label2rgb(L, @jet, [.5 .5 .5]))

hold on



边缘提取图像

**2.图像识别**

**2.1图像特征提取**

采用的是几何参数法

形状的表达和匹配采用最为简单的区域特征描述方法

对于面积一定的图像，一般周长越小，圆形表面越光滑，越接近圆；周长越大，圆形便面的褶皱越多，形状越复杂，因此采用圆形度来衡量图形偏向圆形的程度。圆形度为面积与周长平方之比，记为



s表示图形的面积，l表示图形的周长

**2.2图像分类**

通过计算圆的面积和周长，和设定的阈值相比较，圆形度越接近1的图形越和圆形似。

for k = 1:length(B) % 循环处理每个边界，length(B)是闭合图形的个数,

boundary = B{k}; % 获取边界坐标'

delta\_sq = diff(boundary).^2;% 计算周长

perimeter = sum(sqrt(sum(delta\_sq,2))); % 对标记为K的对象获取面积

area = stats(k).Area;% 计算每个连通区域的面积

metric = 4\*pi\*area/perimeter^2; % 圆度计算公式4\*Pi\*s/l^2

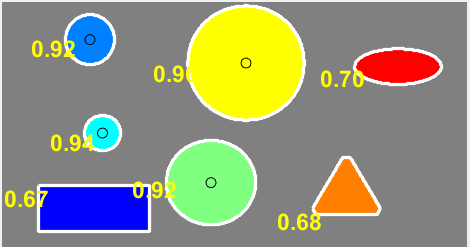
metric\_string = sprintf('%2.2f',metric); %显示圆度结果

if metric > threshold

centroid = stats(k).Centroid;

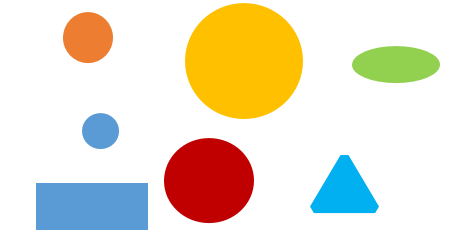
plot(centroid(1),centroid(2),'ko'); % 用一个黑色小圆圈标记圆度大于threshold = 0.90 的对象

end



识别结果

**附录一**



**附录二**

clear;

close all;

clc;

%读取源图像

I = imread('E:\360downloads\28.png');

figure;imshow(I);

%将图片转为灰度图像、取反

bw = rgb2gray(I);

figure;imshow(bw);%灰度图像

bw = imcomplement(bw);%取反

figure;imshow(bw);

%将灰度图像转为二值图像

bw = im2bw(bw,graythresh(bw));%用graythresh找到一个合适的阈值（阈值法将灰度转为二值图像）

imshow(bw);

%消除噪点

se = strel('disk',2);

bw = imclose(bw,se);

figure;imshow(bw);

bw = imfill(bw,'holes');%填补闭合图形

imshow(bw);

%边缘检测

[B,L] = bwboundaries(bw,'noholes');

% 为每个闭合图形设置颜色显示

figure;imshow(label2rgb(L, @jet, [.5 .5 .5]))

hold on

for k = 1:length(B)

boundary = B{k};

plot(boundary(:,2), boundary(:,1), 'w', 'LineWidth', 2)

end

%计算面积

stats = regionprops(L,'Area','Centroid');%找到每个区域的质心

threshold = 0.90;

for k = 1:length(B) % 循环处理每个边界，length(B)是闭合图形的个数,

boundary = B{k}; % 获取边界坐标'

delta\_sq = diff(boundary).^2;% 计算周长

perimeter = sum(sqrt(sum(delta\_sq,2))); % 对标记为K的对象获取面积

area = stats(k).Area; %计算每个连通区域的面积

metric = 4\*pi\*area/perimeter^2; % 圆度计算公式4\*pi\*s/l^2

metric\_string = sprintf('%2.2f',metric); %显示圆度结果

if metric > threshold

centroid = stats(k).Centroid;

plot(centroid(1),centroid(2),'ko'); % 用一个黑色小圆圈标记圆度大于threshold = 0.90 的对象

end

%设置显示字体

text(boundary(1,2)-35,boundary(1,1)+13,metric\_string,'Color','y',...

'FontSize',14,'FontWeight','bold');

end