# 基于matlab的人民币面额识别

来源：网络 作者：清香如梦 更新时间：2024-01-15

*基于matlab的人民币的自动识别姓名学号院系信息科学与工程学院专业年级指导老师摘要本文通过分析第五版人民币的特征，利用纸币中央数字的特征提取和识别的方法，通过matlab软件实现对第五版人民币的100元、50元和20元的识别。关键词：第五...*

基于matlab的人民币的自动识别

姓

名

学

号

院

系

信息科学与工程学院

专

业

年

级

指导老师

摘要

本文通过分析第五版人民币的特征，利用纸币中央数字的特征提取和识别的方法，通过matlab软件实现对第五版人民币的100元、50元和20元的识别。

关键词：第五套人民币

边缘检测和提取

中央数字特征

前言：科技在不断进步，曾经的很多手工劳动如今都被先进的机械和电子设备代替。曾经人们交话费、存款、买东西都必须到营业厅或者商场去办理手续，通过客户和服务人员之间的业务交流实现服务的实施和体验。如今，自动缴费机、存取款一体机和自动售货机出现在了我们的生活中。这些先进的设备中最重要的一门技术就是：纸币识别技术。当我们将钞票投入缴费机或者存款机时，机器必定会首先识别用户投入的是面额为多少的纸币，之后再进行真伪辨别、数据写入等功能。目前已经有很多的识别技术诸如图像匹配法识别，神经网络识别、尺寸识别、纸币内部荧光物质识别等等。本小组提出使用matlab软件，利用软件对图像处理的超强能力，在保证识别准确率的前提下对100元、50元和20元的人民币进行快速有效的识别。

正文

一.背景介绍

二.Matlab函数介绍

1.Imread

函数imread用于读取图片文件中的数据。

调用格式：

A

=

imread(filename,fmt)

[X,map]

=

imread(filename,fmt)

[...]

=

imread(filename)

2.Imshow

imshow是matlab中显示图像的函数。

调用格式：

imshow(BW)：显示一张二值图像BW

imshow(RGB)：显示一张真彩色图像RGB

imshow(X,map)：用指定调色板来显示图像

3.im2bw

matlab中DIP工具箱函数im2bw使用阈值（threshold）变换法把灰度图像（grayscale

image）转换成二值图像。一般意义上是指只有纯黑（0）、纯白（255）两种颜色的图像。

当然，也可以是其他任意两种颜色的组合。所谓二值图像，一般意义上是指只有纯黑（0）、纯白（255）两种颜色的图像。

当然，也可以是其他任意两种颜色的组合。

调用格式：

BW

=

im2bw(I,level)

BW

=

im2bw(X,map,level)

BW

=

im2bw(RGB,level)

其中level就是设置阈值的。level取值范围[0,1]。

4.Imfill

该函数用于填充图像区域和“空洞”。

调用格式：

BW2

=

imfill(BW)

这种格式将一张二值图像显示在屏幕上，允许用户使用鼠标在图像上点几个点，这几个点围成的区域即要填充的区域。要以这种交互方式操作，BW必须是一个二维的图像。用户可以通过按Backspace键或者Delete键来取消之前选择的区域；通过shift+鼠标左键单击或者鼠标右键单击或双击可以确定选择区域。

[BW2,locations]

=

imfill(BW)

这种方式，将返回用户的取样点索引值。注意这里索引值不是选取样点的坐标。

BW2

=

imfill(BW,locations)

这种格式允许用户编程时指定选取样点的索引。locations是个多维数组时，数组每一行指定一个区域。

BW2

=

imfill(BW,＇holes＇)

填充二值图像中的空洞区域。

如，黑色的背景上有个白色的圆圈。

则这个圆圈内区域将被填充。

I2

=

imfill(I)

这种调用格式将填充灰度图像中所有的空洞区域。

BW2

=

imfill(BW,locations,conn)

5.Bwperim

用于查找二值图像的边缘。

调用格式：

BW2

=

bwperim(BW1)

BW2

=

bwperim(BW1,conn)

BW2

=

bwperim(BW1,conn)

表示从输入图像BW1中返回只包括对象边缘像素点的图像。

6.Mode

众数函数，用于计算一组数据中的众数。

众数，简单的说，就是一组数据中占比例最多的那个数。

调用格式：

mode(x)

7.medfilt2

中值滤波函数.调用格式：

medfilt2(A,[m

n])

8.Logical

用于判断参数是否为逻辑值，如果检验内容为逻辑值，将返回TRUE（1），否则返回FALSE（0）。

调用格式：

Logical（value）

9.Find

用来对原始数据中的某个字符串进行定位，以确定其位置。

10.edge

调用格式：

BW=edge(I)

采用灰度或一个二值化图像I作为它的输入，并返回一个与I相同大小的二值化图像BW，在函数检测到边缘的地方为1，其他地方为0.BW=edge(I,’sobel’)

自动选择阈值用sobel算子进行边缘检测。

算子还有roberts算子，prewitt算子，log算子等等。

三.纸币识别流程图

读入待检测图片

得出结果

进行图像边缘检测提取和边缘剪切

中央数字部分识别

提取中央数字部分

四.流程各部分详细说明

1.读入待检测图片

Im=imread（‘100.jpg’）;

figure(1)

imshow(Im);

读入一张有黑色背景的100元人民币图片。然后使用imshow函数进行图片查看。

2.进行图像边缘检测提取

边缘检测共有两种方法，一种是使用edge函数进行边缘检测；另一种是二值化+图像填充+提取边缘的方法。

使用edge函数进行边缘检测，选择Sobel算子。

F=edge(I2,＇sobel＇);%sobel算子边缘检测

figure(2)

imshow(F)

如图，sobel算子边缘检测后的图像。达到需要的效果。但是由于内部白色纹路较多，为了不影响边缘截取，故选择第二种方法：二值化+图像填充+提取边缘。

首先对图像进行二值化处理，然后使用imfill函数，调用“hole”格式，对图像中的空洞进行填充。然后调用bwperim函数，对二值图像进行边缘检测。下面是检测的过程和结果。

3.边缘剪切

得到了二值化的边缘图像后，需要对边缘进行剪切，以切掉背景图案，只留下钞票图案，便于按比例分割图片。

我们采用mode函数，对边缘位置确定。首先，将二值图读入矩阵，生成m行n列的矩阵。然后一行一行进行检测，遇到白色的区域（值为1）时，就将白色区域所处的列位置记入数组。当检测完毕后，使用mode函数对数组进行检测，数组中数字出现的最多的即是边缘的位置。关键循环如下：

for

i=1:m

for

j=1:n

if

F(i,j)~=0%0代表黑色,如果F(i,j)不是黑色（则是图像区域）

k=k+1;

left(k)=j;

for

j1=n:-1:j

if

F(i,j1)~=0

l=l+1;

right(l)=j1

break

end

end

break

end

end

end

如图为剪切后的效果灰度图。

4.提取中央数字部分

百元钞票左边到中央数字1的距离大概占整个长度的0.25，右边是0.5，上边是0.275，下边是0.5。20元和50元的类似。所以使用：

FFv2=jianqie(0.275\*m:0.50\*m,0.254\*n:0.5\*n)

就可以得到中央数字的图片。

5.中央部分识别

中央部分识别分两步：1.区分百元和50、20元；2.区分50元和20元。

区分百元的方法很简单。因为无论是百元还是50元还是20元，截取的部分大小都是固定的，而100是3位数，故而所占图片的比例肯定最大。经手算得100元的中央数字大约占超过截取道德图像的80%。所以由此可以判断出待检图像是不是100元。以下是关键程序：

Ty\_100=logical(sum(FFv2\_bw)=0.8%

val\_mid=100;%val\_mid为检测结果

Else

……

如果待检图片不是100元图片，则进行下一步，50元和20元的区分。

因为50和20均为2位数，故无法通过所占比例进行区分。只能通过数字的某些特征进行区分。我们只需要区分开5和2即可。

如图所示5和2的细节图。取5和2的靠左的一小部分进行比较（红框圈的），数字5后两根蓝线之间的距离小于上面两根蓝线的距离，而2恰恰相反。于是可以得出结果。

关键程序如下：

[hang\_m,hang\_n]=size(hang);

hang\_new=round(sum(hang,2)/hang\_n);

hang\_black=hang\_new(2)-hang\_new(1);%第一个黑点和第二个白点之间的距离

hang\_white=hang\_new(3)-hang\_new(2);%第二个白点和第二个黑点之间的距离

if

hang\_black>=hang\_white

val\_mid

=

50;

else

val\_mid

=

20;

end

end

五.程序运行结果

结果显示正确。

六.遇到的问题

1.关于背景复杂度的问题。

由于边缘检测和提取的算法不完善导致识别程序具有局限性。体现在如下两方面：

（1）

过于复杂的背景无法正常识别；

因为背景如果过于复杂，二值化后会导致很多复杂的背景变成连续的线条。当这些线条比较长的时候，边缘检测程序就会将这段线条识别为边缘，因为众数计算的时候这个位置同样满足。所以系统会认为此处就是边缘，但事实上只是背景中的一些二值化的线条。故而导致剪切边缘无法正确实现，进而影响到后面的中央分割和识别。

如图，是复杂背景的边缘检测结果。

第一幅图的原图为黑色背景中有随机绘制的几条长曲线，第二幅图的原图是背景为实际生活中的建筑物照片。两幅图的轮廓均无法正常提取。第一幅从曲线部分开始进行提取，位置错误。第二幅图根本无法正常提取。

所以，本程序存在一定的局限性，只能处理一些背景不太复杂、二值化之后不会出现过多长曲线的纸币图。

（2）

浅色背景无法正常识别。

本程序的边缘检测方法是图像二值化后利用背景和纸币之间的黑白差异来寻找纸币的边缘。当人民币的背景为浅色时，二值化的时候会将浅色自动二值化为白色（1），而纸币内部只有少部分会二值化为黑色，但边缘周围仍为白色，这样一来根本无法区分纸币和背景，所以用imfill填充后无法清除地看到二值图的边缘。

以下是实验图片：

从图中可以看到，二值化后整幅图片除了中央一少部分其他地方均被二值化为白色，所以轮廓提取失败了。

2.关于边缘检测的两种算法的问题。

小组成员首先使用了edge函数中使用sobel算子的检测方法。但是通过实验发现，当处理一些背景比较“干净”的图片时，方法可行；当背景中有一些干扰时，边缘检测和提取会出现各种问题。经过上网查询和思考，我们决定使用另一种边缘检测和提取的方法，即二值化+填充空洞+bwperim函数的方法。下图是edge函数的边缘检测图像。

从这幅图可以看出，用edge函数得到的边缘虽然完整，但是内部图像却仍然存在，有时会干扰边缘位置确定的结果，所以本次课程设计我们选择第二种方法。

下图是背景中存在干扰的图像和经edge函数处理得到的剪切图。

由图可知，sobel算子边缘检测在这种情况下无法正常运行。

七.心得体会

通过这次的课程设计，我们对图像的识别又有了一个新的思路，同时了解了很多关于matlab的专业知识，熟悉了很多函数的用法，并且对编写程序的过程中出现的各种问题有了了解，对于以后的学习有着莫大的好处。通过这次课程设计，我们深知自己还有很多不足，相信在接下来的时间里，我们会不断改正、完善我们的程序。

参考文献：

《matlab初学者教程（翻译版）》

《基于matlab的四种人民币识别的方法》

邵爱刚

《数字信号处理（清华大学出版社）》

附录：

程序代码：

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%-------------【课程设计课题】100元，50元，20元人民币的识别---------------%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%—————————————————————————%

%----------------图像预处理部分------------------%

%—————————————————————————%

%边缘检测

%

im=imread(＇50yuan.jpg＇);%读取原图

%

figure(1)

%

subplot(2,2,1)

%

imshow(im);

%

title(＇原始图像＇);%显示原图

%

im=im2bw(im);%转二值图像

%

subplot(2,2,2)

%

imshow(im)

%

title(＇二值图像＇);%显示二值图像

%

im2=imfill(im,＇holes＇);%填充“空洞”

%

F=bwperim(im2);%轮廓提取(bwperim函数用于二值图像的轮廓提取）

%

subplot(2,2,3)

%

imshow(im2);

%

title(＇填充后的图像＇)%显示填充后的%

subplot(2,2,4)

%

imshow(F);

%

title(＇提取出的轮廓＇)

%以下是利用edge函数和sobel算子进行的边缘检测

I=imread(＇imfill100.jpg＇);

I1=rgb2gray(I);

n=1;

H=1/(n\*n)\*ones(n,n);

I2=imfilter(I1,H);

figure(1)

subplot(1,3,1)

imshow(I)

title(＇原图＇)

subplot(1,3,2)

imshow(I1)

title(＇灰度图＇)

subplot(1,3,3)

imshow(I2)

title(＇滤波后的图＇)

F=edge(I2,＇sobel＇);%sobel算子边缘检测

figure(2)

imshow(F)

%边缘提取主要算法

[m

n]=size(F)%j将图像数据赋值到矩阵，m为行数，n为列数

F=double(F)%将图像数据转换为double类型

k=0;l=0;%变量初始化

%左右边缘提取

%\*\*\*\*\*\*\*方法简述：两边同时一行一行检测，\*\*\*\*%

%\*\*\*\*\*\*\*把每行轮廓所在的列数记下，然后\*\*\*\*\*\*%

%\*\*\*\*\*\*\*求众数.\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*%

for

i=1:m

for

j=1:n

if

F(i,j)~=0%0代表黑色,如果F(i,j)不是黑色（则是图像区域）

k=k+1;

left(k)=j;

for

j1=n:-1:j

if

F(i,j1)~=0

l=l+1;

right(l)=j1

break

end

end

break

end

end

end

%计算众数，求出左右边界

right=mode(right)

left=mode(left)

%上下边界提取

k=0;l=0;

for

j=1:n

for

i=1:m

if

F(i,j)~=0%0代表黑色,如果F(i,j)不是黑色（则是图像区域）

k=k+1;

up(k)=i;

for

i1=m:-1:i

if

F(i1,j)~=0

l=l+1;

down(l)=i1

break

end

end

break

end

end

end

%计算众数，求出上下边界

up=mode(up)

down=mode(down)

%根据得到的边缘位置截取图像

pic=imread(＇imfill100.jpg＇)

jianqie=pic(up:down,left:right);

%jianqie=im(up:down,left:right);

figure(2)

subplot(2,1,1)

imshow(im)

title(＇原图像＇)

subplot(2,1,2)

imshow(jianqie)

title(＇边缘剪切后的图像＇)

%—————————————————————————%

%----------------图像识别部分--------------------%

%—————————————————————————%

%中央数字提取

[m

n]=size(jianqie)

%F=I(round(0.24\*m):round(0.50\*m),round(0.25\*n):round(0.52\*n))

FFv2=jianqie(0.275\*m:0.50\*m,0.254\*n:0.5\*n)%截取中央图像

FFv2=medfilt2(FFv2,[4,4])%4\*4矩阵中值滤波

FFv2=FFv2(2:(end-4),2:(end-4));

[mt2

nt2]=size(FFv2)

FFv2\_bw=im2bw(FFv2,0.5);%二值化

figure(3)

subplot(4,1,1)

imshow(jianqie)

title(＇原图像＇)

subplot(4,1,2)

imshow(FFv2)

title(＇中央数字提取+中值滤波后的图像＇)

subplot(4,1,3)

imshow(FFv2\_bw)

title(＇二值化后的中央数字图像＇)

%纸币检测

Ty\_100=logical(sum(FFv2\_bw)=0.8%如果大于0.8则是100.此处的系数与之前截取中央图像有关

val\_mid=100;%val\_mid为检测结果

else%以下用于检测50和20

lie\_f

=

0;

for

j

=

1:(nt2/2)

for

i

=

1:(mt2/2)

if

~(FFv2\_bw(i,j)+lie\_f)

lie=j;

lie\_f=1;

end

end

end

%

向当前列后

2——6

共5列，取黑白行数

FFv2\_bw\_f=FFv2\_bw(:,(lie+2):(lie+6));

hang(3,:)=mt2;

for

j=1:5

for

i=1:(mt2-2)

if

~(FFv2\_bw\_f(i,j)+FFv2\_bw\_f(i+1,j)+FFv2\_bw\_f(i+2,j))

hang(1,j)=i;%第一行记录首次黑点行数

for

k=(i+2):(mt2-2)

if

(FFv2\_bw\_f(k,j)+FFv2\_bw\_f(k+1,j)+FFv2\_bw\_f(k+2,j))>=2

hang(2,j)=k+1;%第二行记录二次白点

for

l=(k+2):mt2

if

~(FFv2\_bw\_f(l,j))

hang(3,j)=l;%第三行记录二次黑点

break

end

end

break

end

end

break

end

end

end

[hang\_m,hang\_n]=size(hang);

hang\_new=round(sum(hang,2)/hang\_n);

hang\_black=hang\_new(2)-hang\_new(1);%第一个黑点和第二个白点之间的距离

hang\_white=hang\_new(3)-hang\_new(2);%第二个白点和第二个黑点之间的距离

if

hang\_black>=hang\_white

val\_mid

=

50;

else

val\_mid

=

20;

end

end

subplot(4,1,4)

imshow(＇jieguo.jpg＇)

xlabel([val\_mid],＇Fontsize＇,36,＇color＇,＇r＇);

本DOCX文档由 www.zciku.com/中词库网 生成，海量范文文档任你选，，为你的工作锦上添花,祝你一臂之力！