**车牌识别系统的脚本代码**

1.输入车牌原始图像

[filename,pathname] = uigetfile('\*.jpg','读取图片');

if isequal(filename,0)

msgbox('没有图片')

else

pathfile=fullfile(pathname,filename);

I=imread(pathfile);

End

（若未识别到图像，之后会有未识别到图像的提示）

2.转化为灰度图像

I1=rgb2gray(I);

figure(2),subplot(1,2,1),imshow(I1);title('灰度图');

figure(2),subplot(1,2,2),imhist(I1);title('灰度图直方图');

3.算子边缘检测

I2=edge(I1,'sobel',0.15,'both');

figure(3),imshow(I2);title('算子边缘检测');

se=[1;1;1];

4.图像腐蚀

I3=imerode(I2,se);

figure(4),imshow(I3);title('腐蚀后图像');

se=strel('rectangle',[25,25]);

5.图像闭运算

I4=imclose(I3,se);

figure(5),imshow(I4);title('平滑图像的轮廓');

6.移除小面积对象

I5=bwareaopen(I4,2000);

figure(6),imshow(I5);title('从对象中移除小对象');

[y,x,z]=size(I5);

myI=double(I5);

tic

Blue\_y=zeros(y,1);

for i=1:y

for j=1:x

if(myI(i,j,1)==1)

Blue\_y(i,1)= Blue\_y(i,1)+1;%蓝色像素点统计

end

end

End

7.车牌定位

图像预处理结束获得车牌号大致位置后将图像转化为双精度图像。之后对像素值进行扫描以确定车牌区域边界。

[temp MaxY]=max(Blue\_y);%Y方向车牌区域确定

PY1=MaxY;

while ((Blue\_y(PY1,1)>=5)&&(PY1>1))

PY1=PY1-1;

end

PY2=MaxY;

while ((Blue\_y(PY2,1)>=5)&&(PY2<y))

PY2=PY2+1;

end

IY=I(PY1:PY2,:,:);

%%%%%% X方向 %%%%%%%%%

Blue\_x=zeros(1,x);%进一步确定x方向的车牌区域

for j=1:x

for i=PY1:PY2

if(myI(i,j,1)==1)

Blue\_x(1,j)= Blue\_x(1,j)+1;

end

end

End

8.字符分割

在对字符进行切割之前需要对车牌区域图像再次进行一次处理。

分别是灰度化处理、二值化处理、均值滤波处理、腐蚀（膨胀处理）。

[temp MaxY]=max(Blue\_y);%Y方向车牌区域确定

PY1=MaxY;

while ((Blue\_y(PY1,1)>=5)&&(PY1>1))

PY1=PY1-1;

end

PY2=MaxY;

while ((Blue\_y(PY2,1)>=5)&&(PY2<y))

PY2=PY2+1;

end

IY=I(PY1:PY2,:,:);

%%%%%% X方向 %%%%%%%%%

Blue\_x=zeros(1,x);%进一步确定x方向的车牌区域

for j=1:x

for i=PY1:PY2

if(myI(i,j,1)==1)

Blue\_x(1,j)= Blue\_x(1,j)+1;

end

end

end

PX1=1;

while ((Blue\_x(1,PX1)<3)&&(PX1<x))

PX1=PX1+1;

end

PX2=x;

while ((Blue\_x(1,PX2)<3)&&(PX2>PX1))

PX2=PX2-1;

end

PX1=PX1-1;%对车牌区域的校正

PX2=PX2+1;

dw=I(PY1:PY2-8,PX1:PX2,:);

G=toc;

figure(7),subplot(1,2,1),imshow(IY),title('行方向合理区域');

figure(7),subplot(1,2,2),imshow(dw),title('定位剪切后的彩色车牌图像');

imwrite(dw,'dw.jpg');

a=imread('dw.jpg');

b=rgb2gray(a);%功能是将真彩色图像转换为灰度图像，即灰度化处理

imwrite(b,'1.车牌灰度图像.jpg');

figure(8);subplot(3,2,1),imshow(b),title('1.车牌灰度图像')

g\_max=double(max(max(b)));

g\_min=double(min(min(b)));

T=round(g\_max-(g\_max-g\_min)/3); % T 为二值化的阈值 向最近的方向取整

[m,n]=size(b);

d=(double(b)>=T); % d:二值图像

imwrite(d,'2.车牌二值图像.jpg');

figure(8);subplot(3,2,2),imshow(d),title('2.车牌二值图像')

figure(8),subplot(3,2,3),imshow(d),title('3.均值滤波前')

% 均值滤波处理

h=fspecial('average',3);

d=im2bw(round(filter2(h,d)));%filter2(B,X),B为滤波器.X为要滤波的数据,这里将B放在X上,一个一个移动进行模板滤波.

imwrite(d,'4.均值滤波后.jpg');

figure(8),subplot(3,2,4),imshow(d),title('4.均值滤波后')

se=eye(2);%产生m×n的单位矩阵

[m,n]=size(d);

if bwarea(d)/m/n>=0.365 %bwarea是计算二值图像中对象的总面积的函数

d=imerode(d,se);%腐蚀

elseif bwarea(d)/m/n<=0.235

d=imdilate(d,se);%膨胀

end

imwrite(d,'5.膨胀或腐蚀处理后.jpg');

figure(8),subplot(3,2,5),imshow(d),title('5.膨胀或腐蚀处理后')

H=toc;

d=qiege(d);

figure,subplot(2,1,1),imshow(d),title(n)

[word1,d]=getword(d);

% 分割出第二个字符

[word2,d]=getword(d);

% 分割出第三个字符

[word3,d]=getword(d);

% 分割出第四个字符

[word4,d]=getword(d);

% 分割出第五个字符

[word5,d]=getword(d);

% 分割出第六个字符

[word6,d]=getword(d);

% 分割出第七个字符

[word7,d]=getword(d);

subplot(5,7,1),imshow(word1),title('1');

subplot(5,7,2),imshow(word2),title('2');

subplot(5,7,3),imshow(word3),title('3');

subplot(5,7,4),imshow(word4),title('4');

subplot(5,7,5),imshow(word5),title('5');

subplot(5,7,6),imshow(word6),title('6');

subplot(5,7,7),imshow(word7),title('7');

end

function [word,result]=getword(d)

word=[];flag=0;y1=8;y2=0.5;

while flag==0

[m,n]=size(d);

wide=0;

while sum(d(:,wide+1))~=0 && wide<=n-2

wide=wide+1;

end

temp=qiege(imcrop(d,[1 1 wide m]));%返回图像的一个切割区域

[m1,n1]=size(temp);

if wide<y1 && n1/m1>y2

d(:,[1:wide])=0;

if sum(sum(d))~=0

d=qiege(d); % 切割出最小范围

else word=[];flag=1;

end

else

word=qiege(imcrop(d,[1 1 wide m]));

d(:,[1:wide])=0;

if sum(sum(d))~=0;

d=qiege(d);flag=1;

else d=[];

end

end

end

end

result=d;

end

function e=qiege(d)

[m,n]=size(d);

top=1;bottom=m;left=1;right=n; % init

while sum(d(top,:))==0 && top<=m

top=top+1;

end

while sum(d(bottom,:))==0 && bottom>=1

bottom=bottom-1;

end

while sum(d(:,left))==0 && left<=n

left=left+1;

end

while sum(d(:,right))==0 && right>=1

right=right-1;

End

9.字符识别

首先建立单个字符二值化图像的样本库，然后将切割后的每个字符二值化图像与所有样本做减法，所得到的1值像素最少的样本即为识别结果。

dd=right-left;

hh=bottom-top;

e=imcrop(d,[left top dd hh]);%该函数用于返回图像的一个裁剪区域

[m,n]=size(word1);

% 归一化大小为 40\*20

word1=imresize(word1,[40 20]);

word2=imresize(word2,[40 20]);

word3=imresize(word3,[40 20]);

word4=imresize(word4,[40 20]);

word5=imresize(word5,[40 20]);

word6=imresize(word6,[40 20]);

word7=imresize(word7,[40 20]);

subplot(5,7,15),imshow(word1),title('1');

subplot(5,7,16),imshow(word2),title('2');

subplot(5,7,17),imshow(word3),title('3')

subplot(5,7,18),imshow(word4),title('4');

subplot(5,7,19),imshow(word5),title('5');

subplot(5,7,20),imshow(word6),title('6');

subplot(5,7,21),imshow(word7),title('7');

imwrite(word1,'1.jpg');

imwrite(word2,'2.jpg');

imwrite(word3,'3.jpg');

imwrite(word4,'4.jpg');

imwrite(word5,'5.jpg');

imwrite(word6,'6.jpg');

imwrite(word7,'7.jpg');

liccode=char(['0':'9' 'A':'Z' '京津沪晋辽吉鲁苏浙皖鲁豫粤川陕新黑宁']);

JG=zeros(40,20);%产生一个40\*20大小的零矩阵

l=1;

L=toc;

for I=1:7 %I为待识别的字符位

ii=int2str(I);%整形数据转化为字符串类型

t=imread([ii,'.jpg']);

MB=imresize(t,[40 20],'nearest');%缩放处理

if l==1 %车牌号第一位为汉字识别，使用37-53号样本库

kmin=37;

kmax=53;

elseif l==2 %车票号第二位为 A~Z 大写字母识别，使用11-36号样本库

kmin=11;

kmax=36;

else l>=3; %第三位以后是字母或数字识别，使用1-36号样本库

kmin=1;

kmax=36;

end

for k2=kmin:kmax

fname=strcat('样本库\',liccode(k2),'.bmp');

YB = imread(fname); %调用样本库图像文件

for i=1:40

for j=1:20

JG(i,j)=MB(i,j)-YB(i,j); % 这里是将待识别图像与模板图像两幅图相减得到第三幅图

end

end

Dmax=0;

for k1=1:40

for l1=1:20

if ( JG(k1,l1) > 0 || JG(k1,l1) <0 )

Dmax=Dmax+1;

end

end

end

Error(k2)=Dmax;

end

Error1=Error(kmin:kmax);

MinError=min(Error1);

findc=find(Error1==MinError);

Code(l\*2-1)=liccode(findc(1)+kmin-1);

Code(l\*2)=' ';

l=l+1;

end

t=toc;

figure(10),imshow(dw),title (['车牌号:', Code],'Color','k' ,'font','20');