Poker识别

1. 实验内容

识别扑克牌的花色与字符。

1. 解决方案
2. 工程流程

提取扑克牌（即删掉除扑克牌以外的其它区域）

校正扑克牌的位置（即让扑克牌竖直摆放）

打开要识别的扑克牌图片

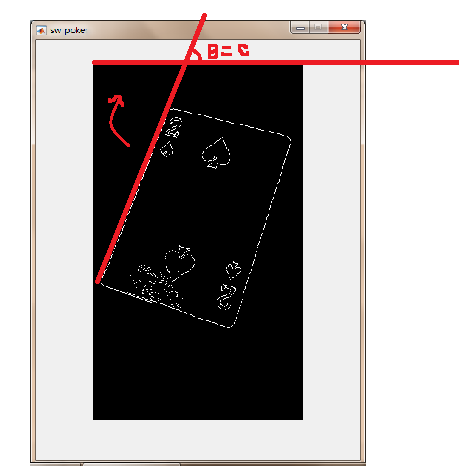
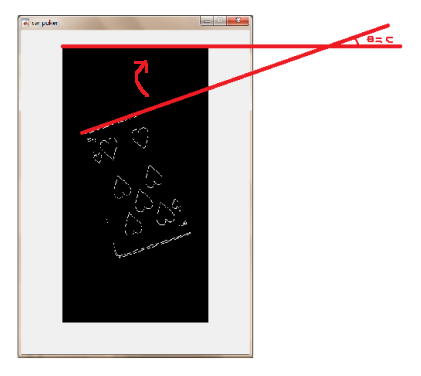
扑克牌灰度化、二值化

提取扑克牌的特征（字符与花色）

将特征字符与扑克牌图片模板比对识别纸牌字符与花色

1. 具体实现
2. 校正扑克牌的位置

i原理：通过edge得到只含扑克牌边缘的图片，然后在该图片上沿着不同方向进行像素叠加，通过最大像素叠加和（即最长直线）可得扑克牌与图片边缘的夹角，然后将原图沿一定的方向旋转相应角度即可得到扑克牌竖直放置的图片。



图片1 图片2

ii核心代码：

function poker=rectify(rgbimage)

grey=rgb2gray(rgbimage);

bw=edge(grey,'sobel','horizontal');%检测图像边缘直线

[m,n]=size(bw);%计算图像大小

S=round(sqrt(m^2+n^2));%S可以取到的最大值

ma=180;%θ角最大值

md=S;

r=zeros(md,ma);%产生初值为零的计数矩阵

for i=1:m

for j=1:n

if bw(i,j)==1

for k=1:ma

ru=round(abs(i\*cos(k\*3.14/180)+j\*sin(k\*3.14/180))); %确定记录累加像素值的矩阵的行号

r(ru+1,k)=r(ru+1,k)+1;%对矩阵记数（对经过（i，j）位置的像素点并与水平方向夹角为k的方向上的像素进行累加）

end

end

end

end

[m,n]=size(r);

for i=1:m

for j=1:n

if r(i,j)>r(1,1)

r(1,1)=r(i,j);

c=j;%把矩阵元素最大值所对应的列坐标送给c，列坐标即为最长直线与水平方向的夹角。

end

end

end

if c<=90

rot=-c; %确定旋转角度，c<90时顺时针旋转c度

else

rot=180-c; %c>90时逆时针旋转180-c度

end

%------------------------------------------

%bw=imrotate(bw,rot,'');

%[m,n]=find(bw);

%bw1=bw(min(m):max(m),min(n):max(n));

%[m,n]=size(bw1);

%if m<n%如果扑克牌旋转之后是水平放置的，则将扑克牌逆时针旋转90度让其竖直放置

% rot=rot+90;

%end

%------------------------------------------------

poker=imrotate(rgbimage,rot,'crop'); %对图片进行旋转，矫正图像

1. 提取扑克牌

i原理：在校正好的图片上用鼠标框出扑克牌，并将框出的扑克牌保存为新的图片。

ii核心代码：

h=imrect;

%图中就会出现可以拖动以及改变大小的矩形框，选好位置后：

pos=getPosition(h);

%pos有四个值，分别是矩形框的左下角点的坐标 x y 和 框的 宽度和高度

%拷贝选取图片

POKER= imcrop(POKER, pos );

1. 扑克牌灰度化、二值化

i原理：将提取出的扑克牌通过rgb2grey灰度化，将灰度化的图片通过im2bw二值化，其中二值化的阈值由最大类间方差法得到。

ii核心代码：

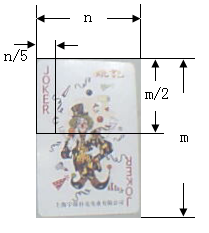
GRAY=rgb2gray(POKER);

bw=im2bw(GRAY,ostu(GRAY)); %最大类间方差法ostu()

BW=logical(abs(double(bw)-1));%取反色

1. 提取扑克牌特征

i原理：先进行字符粗略定位（即取纸牌1/2的上部，1/5.5的左部）,然后利用求每列像素和删除孤立线条，最后利用find找到字符与花色的位置，利用位置信息的得到只含纸牌特征（字符与花色）的图片。



ii核心代码：

[m,n]=size(GRAY);

pokerstr=GRAY(2:m/2,n/20:n/5.5); %字符粗略定位

bw=im2bw(pokerstr,ostu(pokerstr)); %自定义函数ostu()

bw1=bwmorph(bw,'clean'); %清除孤立点

bw2=logical(abs(double(bw1)-1)); %二值图像反色

bw3=reduce(bw2); %自定义函数reduce()

[m,n]=size(bw3);

temp=sum(bw3); %以列为对象，对每一列求和

shadow(2:n+1)=temp; %把temp的每一列赋给shadow从2到n+1相应列

shadow(1)=0;

shadow(n+2)=0;

for i=2:n+1

if shadow(i)~=0&shadow(i-1)==0&shadow(i+1)==0%出现孤立线条

for j=1:m

bw3(j,i-1)=0; %删除孤立线条列

end

end

end

[m,n]=find(bw3);

BWSTR=bw3(min(m):max(m),min(n):max(n));

subplot(1,1,1),imshow(BWSTR),title('纸牌特征字符');

1. 扑克牌识别

i原理：将只含纸牌特征的图片大小利用imresize转换为58\*22的大小，然后将每行的像素相加得到一维矩阵（用sum(pic,2)实现）。将这个一维矩阵与扑克牌的模板进行求差值运算（扑克牌的模板获得方法与得到这个一维矩阵一致），差值最小的那个模板所对应的扑克牌的字符与花色即为本次扑克牌识别结果。

ii核心代码：

sym=bwmorph(BWSTR,'clean'); %清除孤立点

SYM1=imresize(sym,[58 22]); %模板归一化

shadow1=(sum(SYM1,2))';%侧面投影-------对每一行求和

for i=1:40 %搜索最佳匹配的模板----------共有13个纸牌字符

temp=STR(i,:);

error=abs(shadow1-temp); %求差值找到最符合的模板

error=mean(error,2); %求绝对均差

chazhi(i)=error;

end

chazhi=chazhi';

chazhi\_sort=sort(chazhi);

sn1=strfind(chazhi,chazhi\_sort(1));%得到差值最小的纸牌模板序号

%特征映射

switch sn1

case 0

result1='－不能识别出字符';

case 1

result1='方片-A';

case 2

result1='方片-2'; %对应数字转成字符型

...

...

case 39

result1='红桃-9'; %对应数字转成字符型

case 40

result1='红桃-10'; %对应数字转成字符型

end

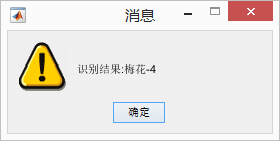
result=strcat('识别结果:',result1);

msgbox(result,'消息','warn');

end

end

1. 识别结果



1. 总结
2. 扑克识别的过程为图片预处理，图片切割，图片匹配。其中预处理包括对图片校正，校正时利用hough变换原理得到扑克与图片边缘的夹角，在获取扑克边缘时因为图片质量可能会导致所得到的最长直线为扑克牌较小的那条边，所以在进行原图图片旋转校正前会先对只含扑克边缘的图片进行旋转，如果旋转后的扑克边缘图中较长的那条边竖直放置，则对原图进行相同旋转。如果旋转后扑克边缘图中较长的那条边水平放置，则将旋转角度变换90度后，再对原图进行旋转（由于本工程处理识别的图像大多数为图片1那样的，所以对旋转角度的处理可以忽略）。图片切割利用手工截取是为了避免图片背景里可能存在其它对象的影响。
3. 扑克识别的精度与模板有关，可以通过修改模板数据来识别更多扑克，也可以增加模板来识别更多扑克。
4. 参考资料

http://www.pudn.com/downloads169/sourcecode/graph/detail778667.html