信息处理课程设计

————基于Matlab GUI的车牌识别

学院：计算机学院

指导教师：杨小东

班级：信科12（3）班

姓名：郭柱材

学号：08123451

摘要：此次课程的主题为信息处理，本设计为基于Matlab GUI车牌识别系统，以此例学习研究有关于图像处理方面的原理知识及应用。此次车牌识别系统的设计过程可分为五大模块：原图预处理，车牌提取，车牌预处理，字符分割，字符模板匹配，基于Matlab GUI的程序界面化。最终成功实现通过人机交互界面提取车牌并识别车牌字符的功能。

关键词：车牌识别，Matlab，GUI

**目录**

[一. 绪论 3](#_Toc418508119)

[1.1车牌号识别技现实意义 3](#_Toc418508120)

[1.2车牌识别技术的研究现状 3](#_Toc418508121)

[1.3车牌识别技术的未来发展趋势 4](#_Toc418508122)

[二.车牌识别技术简介 5](#_Toc418508123)

[三.车牌识别系统设计流程 6](#_Toc418508124)

[3.1 源图像预处理 6](#_Toc418508125)

[3.1.1.车牌读取 6](#_Toc418508126)

[3.1.2.车牌图像灰度化 7](#_Toc418508131)

[3.1.3.边缘检测 8](#_Toc418508135)

[3.1.4.灰度图腐蚀 9](#_Toc418508137)

[3.1.5.图像平滑处理 10](#_Toc418508140)

[3.1.6.移除小对象 11](#_Toc418508143)

[3.2车牌提取 12](#_Toc418508144)

[3.3车牌图像预处理 14](#_Toc418508185)

[3.4字符分割 15](#_Toc418508186)

[3.5字符模板匹配识别 19](#_Toc418508253)

[四.系统界面化 23](#_Toc418508308)

[五.补充代码 25](#_Toc418508309)

[六．总结 29](#_Toc418508417)

# 一. 绪论

## **1.1车牌号识别技现实意义**

随着我国公路交通事业的发展，车辆的数量正在迅速增长，在给出行提供方便的同时，车辆管理上存在的问题日益突出，人工管理的方式已经不能满足实际的需要。微电子、通信和计算机技术在交通领域的应用极大地提高了交通管理效率。作为信息来源的自动检测、图像识别技术越来越受到人们的重视。近年来计算机的飞速发展和数字图像技术的日趋成熟，为传统的交通管理带来巨大转变，先进的计算机处理技术，不但可以将人力从繁琐的人工观察、监测中解放出来，而且能够大大提高其精确度，汽车牌照自动识别系统就是在这样的背景与目的下进行开发的。汽车牌照等相关信息的自动采集和管理对于交通车辆管理、园区车辆管理、停车场管理、交警稽查等方面有着十分重要的意义，成为信息处理技术的一项重要研究课题。

## 1.2车牌识别技术的研究现状

自1988年车辆牌照识别(LicensePlateReeognition，LPR)技术推出以来，人们就对其进行了广泛的研究。国外己有不少相关的文章发表，有的己经非常成熟，投入实际使用。在各种应用中有对静态图像处理的系统，如Huang一MuHwang的《APC一BASEDCARLICENSEPLATEREADER》在收费站硬件设施较好的条件下，平均0.7秒内识别率达到97%，如此一些系统的成功实现大大受益于国外车牌的规范统一。从实用产品来看，比较有名的是以色列的Hi一TeehSolutions公司、Zamir公司、匈牙利的AdaptiveReeognition公司和新加坡的Optasia公司等。

我国车牌自动识别的研究起步较晚，大约发生在八十年代末。1988年戴营等人利用常见的图像处理技术方法提出汉字识别的分类是在提取汉字特征的基础上进行的。根据汉字的投影直方图(ProjectionHIStogram)，选取浮动阂值，进行量化处理后，形成一个变长链码，再用动态规划法，求出与标准模式链码的最小距离，实现细分类，完成汉字省名的自动识别。上海交通大学计算机戚飞虎等人研究的基于彩色分割的牌照自动识别系统;华中科技大黄心汉等人研究的基于板匹配和神经网络的牌照识别系统;以及浙江大学图形图像研究所潘云鹤等人研究的“车牌通”产品等。此外，西安交通大学的图像处理和识别研究室、清华大学人工智能国家重点实验室等各大高校也正在做类似的研究。

## **1.3车牌识别技术的未来发展趋势**

　　车牌识别产品的竞争将更多的强调成像、图像预处理、算法、业务逻辑等系统各环节的相互配合。特别是物理成像以及图像预处理的水平可能会成为下阶段发展的关键，其原因在于当识别算法的发展达到一定的高度，短期内很难会出现非常突破性的变革，这时如何能够通过成像、光源、降噪等环节来为核心算法扫清障碍就成为了影响产品最终性能的制衡点。此外，与整体安防行业的发展相应，以下几个趋势对车牌识别产品的影响将会非常深远：

**1.3更高的分辨率及画面质量**

作为一种典型的模式识别算法，车牌识别一直期望有更高的图像分辨率。如果相机的分辨率可以提升千万量级，车牌识别的识别指标将会更加逼近“几乎不出错”的极限，而且车牌识别能够稳定工作的区域也会明显增大。除了分辨率提升外，画面质量也是同等重要的问题，近年来高清相机的分辨率虽然提升的速度很快，但在动态范围的表现上却差强人意，导致在一些场景下出现亮处细节和暗处细节无法兼顾的问题，这将是下阶段相机发展的关键点之一。

**2．更高的计算能力和集成度**

高清相机的发展必然导致车牌识别算法需要更大的计算资源，嵌入式平台核心芯片的更新速度将会直接影响车牌识别产品的形态以及发展方向，可以预见，今后更多的相关业务将会被放置在车牌识别智能相机上。

**3．更高的易用性、更强的适应能力**

目前车牌识别算法本身的难题主要集中在车牌受到各种干扰时如何保证尽可能准确地识别，例如大型车常见的车牌污损、私家车辆规避限号的车牌遮挡、光照环境形成的阴影干扰等等。此外，车牌识别产品的参数配置一直相对繁琐，这很大程度上是为了能够通过精确的调整相机成像、补光、算法的各个环节来确保达到最好的识别效果。如何利用好高分辨率和高计算能力来降低系统的操作难度，进一步解决这些经典的技术问题，提高系统配置的自动化水平是车牌识别产品下阶段需要着重投入的方向。

# 二.车牌识别技术简介

一个完整的车牌号识别系统要完成从图像采集到字符识别输出，过程相当复杂，基本可以分成硬件部分跟软件部分，硬件部分包括系统触发、图像采集，软件部分包括图像预处理、车牌位置提取、字符分割、字符识别四大部分，一个车牌识别系统的基本结构如图2.1所示：

车牌预处理

原始

图像

图像预处

理

车牌

位置

提取

字符

分割

字符

识别

输出

结果

图 2.1 车牌识别系统基本结构框图

（一）：原始图像：由停车场固定彩色摄像机、数码相机或其他扫描装置拍摄到的图像。

（二）：图像预处理:对动态采集到的图像进行滤波，边界增强等处理以克服图像处理。

（三）：车牌位置提取：通过运算得到图像的边缘，再计算边缘图像的投影面积，寻找谷峰点以大概确定车牌的位置，再计算连通域的宽高比，剔除不在阈值范围内的连通域，最后便得到了车牌区域。

（四）：车牌预处理：对提取到的车牌进行二值化，滤波，删除小面积等预处理

（五）：字符分割：利用投影检测的字符定位分割方法得到单个的字符。

（六）：字符识别：利用模板匹配的方法与数据库中的字符进行匹配从而确认出字符。

（七）：输出结果：得到最后的汽车牌照，包括汉字、字母和数字。

# 三.车牌识别系统设计流程

## **3.1 源图像预处理**

### 3.1.1.车牌读取

目前常用的图像格式主要有\*.PCX、\*.BMP、\*.JPG、\*.TIFF、\*.GIF 等，本设计采集到的图片格式为\*.JPG格式，这种格式的图像占有的存储空间小，而且是使用最广的图片保存和传输格式，大多数的摄像设备也都是以\*.JPG格式保存图像的。利用图像工具的图像读取函数imread() 来读取一副图像，其使用格式为：

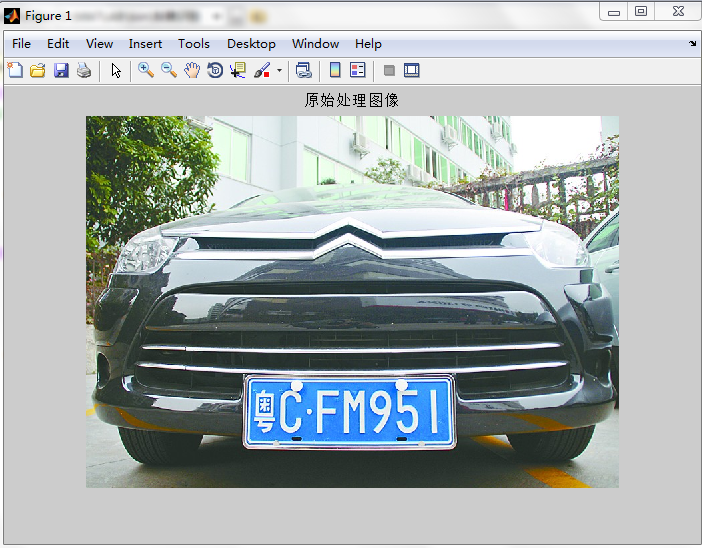
%uigetfile（读取文件名和路径）'图片名.JPG'（文件名和类型）[filename ,pathname]=uigetfile('\*.JPG','选择图片');

str=[pathname, filename];

I=imread(str);

figure,imshow(I); %显示要处理的图像

title('原始处理图像');



### 3.1.2.车牌图像灰度化

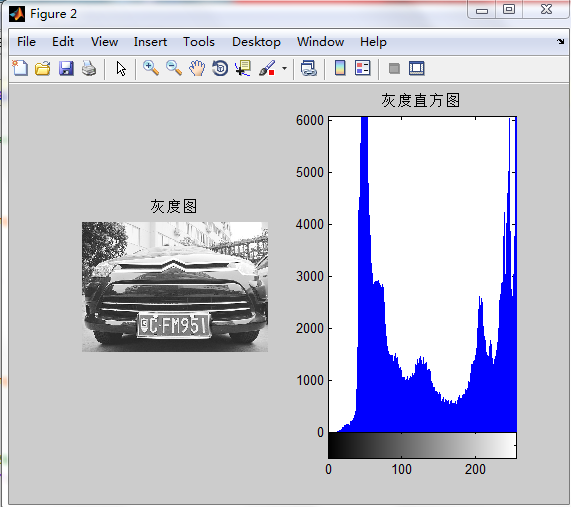
由于车牌底色跟上面的字符的颜色对比度很大，所以将RGB图像转化为灰度图时，车牌底色跟字符的灰度值也会相差很大，这样就可以很明显的显现出车牌区域，便于后续处理。本系统使用的是MATLAB内的灰度图转换函数rgb2gray()，其图像灰度值计算公式为：G=0.299R+0.578G+0.114B;

程序源代码为：

I1=rgb2gray(I);

figure(2),subplot(1,2,1),imshow(I1);title('灰度图');%画出灰度图的灰度直方图

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

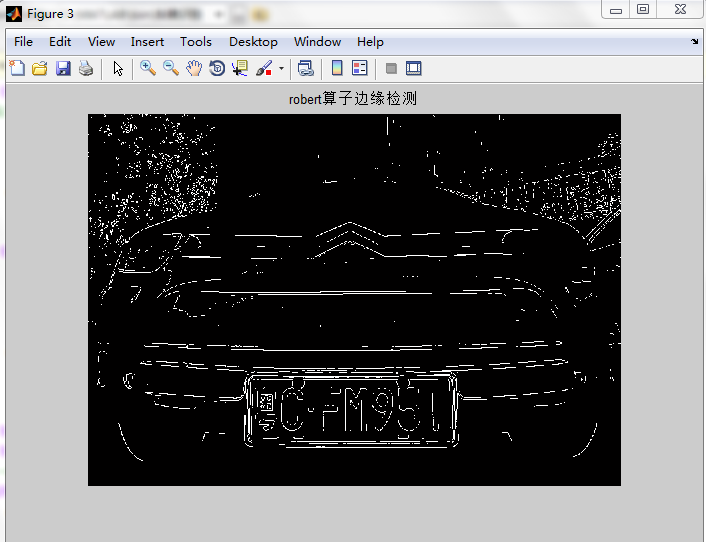


### 3.1.3.边缘检测

在Matlab中利用函数edge()实现边缘检测，在edge()函数中有Sobel算子，Prewitt算子，Roberts算子，Log算子，Canny算子及Robert算子，几种算法相比之下，Roberts算子是一种最简单的算子，是一种利用局部差分算子寻找边缘的算子，他采用对角线方向相邻两象素之差近似梯度幅值检测边缘。检测垂直边缘的效果好于斜向边缘，定位精度高，对噪声敏感,无法抑制噪声的影响。 因此本课题使用了Robert算子。

程序源代码为：

I2=edge(I1,'roberts',0.23,'both'); %0.23为敏感度阀值



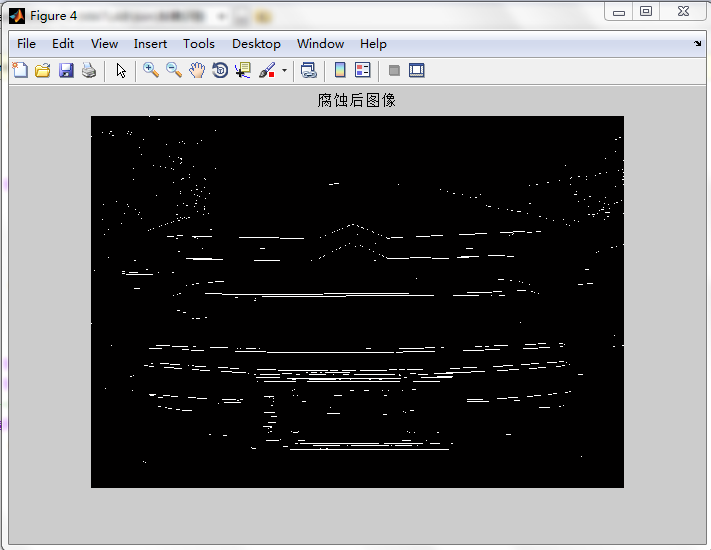
### 3.1.4.灰度图腐蚀

所谓腐蚀即一种消除边界点，使边界向内部收缩的过程。利用它可以消除小而且无意义的物体。腐蚀的规则是输出图像的最小值是输入图像领域中的最小值，在一个二值图像中，只要有一个像素值为0，则相应的输出像素值为0。假设B对X腐蚀所产生的二值图像E是满足以下条件的点（x,y)的集合：如果B的原点平移到点（x,y），那么B将完全包含于X中。本系统使用imerode()函数：

程序源代码为：

se=[1,1,1];

I3=imerode(I2,se);



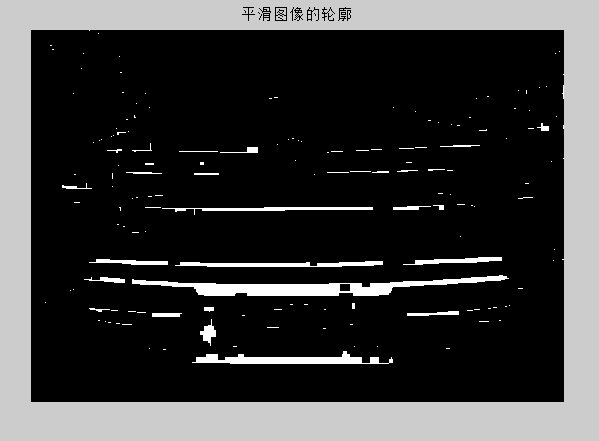
### 3.1.5.图像平滑处理

得到车牌区域的图像轮廓线后，由于图像的数字化误差和噪声直接影响了脚点的提取，因此在脚点提取之前必须对图像进行平滑处理，Matlab有一个图像平滑处理函数imclose(),它与开运算相反，融合窄的缺口和细长的弯口，去掉小洞，填补轮廓上的缝隙。结构单元中Se一个小于对象闭合图形，只要两个封闭域的距离小于Se，就将这两个连接成一个连通域。其中Se是一个一个10\*10的矩形，使用矩形是因为车牌是一个矩形，这样，可以是提取的图像最接近预期效果。由于车牌图像经过腐蚀以后只剩下车牌区域以及车的标志。

程序源代码为：

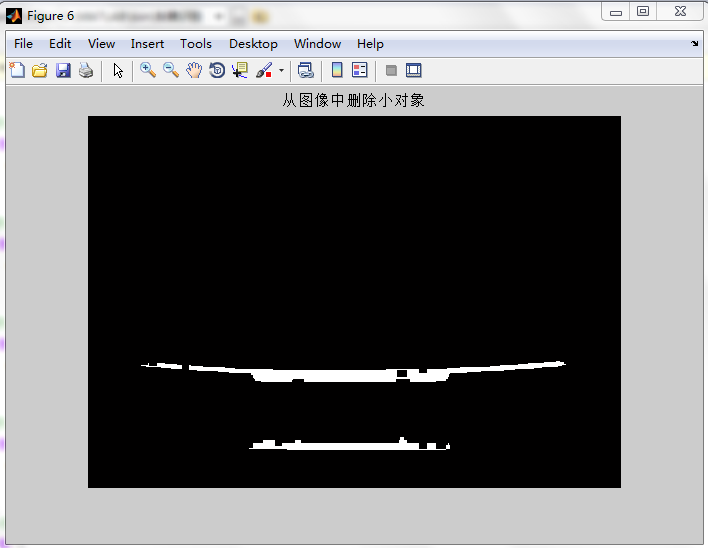
se=strel('rectangle',[10,10]);

I4=imclose(I3,se);



### 3.1.6.移除小对象

图像平滑处理了，可能会有多个闭合区域，对于不是车牌区域的必须予以删除，Matlab提供了一个函数bwareaopen()，用于删除二值图像中面积小于一个定值的对象， 本设计中面积小于2500的对象都需要被删除。

**I5=bwareaopen(I4,2500); %删除I4中面积小于2500**

## **3.2车牌提取**

将原图像预处理后，采用水平垂直双向投影法，将图像分别投影到X、Y坐标轴，然后分别沿X、Y轴扫描图像。当沿Y轴扫描时，一边扫描一边统计图像中白色像素点的个数，第一次扫描到白色像素点最多的行时停止扫描，并记录下白色像素点最多的行，然后以这行为基点，分别向上、向下扫描直到统计的像素点小于像素点阀值时，停止扫描，记录上下行的，即为车牌在图像中的上边沿和下边沿所在的行数

源程序代码为

[y,x,z]=size(I5); %返回图像I5的大小（行，列，面）

myI=double(I5);

tic

%以图片行数y建立一个零向量，只有一列，数值全为0，用于记录白色像素值

Blue\_y=zeros(y,1);

for i=1:y %从图片第一行开始，从上往下检索

for j=1:x %从图片第一列开始，从左往右检索

if(myI(i,j,1)==1) %如果该点为白色，

Blue\_y(i,1)=Blue\_y(i,1)+1;

%Blue\_y(i,1)为第i行，第一列（只有一列）的值，初值都为0，存储白色像素的多少

end

end

end

[temp MaxY]=max(Blue\_y); %temp记录蓝色像素最大为多少，MaxY记录蓝色像素最多的一行

PY1=MaxY;

while (Blue\_y(PY1,1)>=5)&&(PY1>1) %只要MaxY不是最后一行和第一行，那么就同时往上往下搜索，当蓝色像素值小于一个阀值的时候停止搜索

PY1=PY1-1; %图片往上扩展

end

PY2=MaxY;

while (Blue\_y(PY2,1)>=3)&&(PY2<y)

PY2=PY2+1; %图片往下扩展

end

IY=I(PY1:PY2,:,:); %显示图像I的PY1到PY2行，即白色像素最多的几行（车牌）

同理可得X方向

Blue\_x=zeros(1,x); %创建一个零向量（X轴）向量的每个点都为零

for j=1:x %从图片第一列开始，从左往右检索

for i=PY1:PY2 %从图片第PY1行开始，从上往下检索，到PY2行

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

Blue\_x(1,j)=Blue\_x(1,j)+1; %从初值零开始计数蓝色像素的值

end

end

end

PX1=1;

while((Blue\_x(1,PX1)<9)&&(PX1<x)) %纵向的蓝色像素从图像的两头开始往中间检索

PX1=PX1+1;

end

PX2=x;

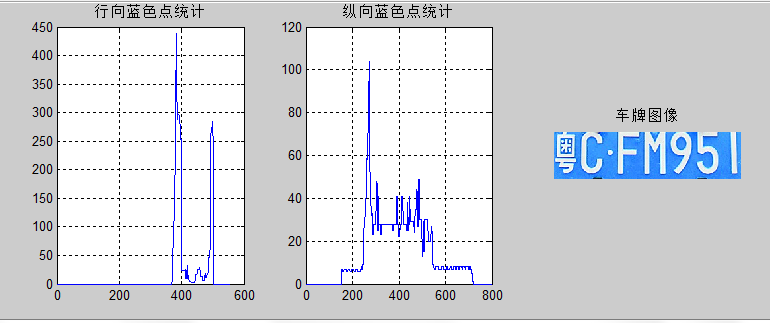
while((Blue\_x(1,PX2)<9)&&(PX2>PX1))

PX2=PX2-1;

end

CP=I(PY1+45:PY2-18,PX1+5: PX2-15,:);

显示结果为



3.3车牌图像预处理

由于图像车牌号区域提取后获得的是从原始图像中剪切的，是RGB图像，后续的分割处理同样采取投影法，故同样需要先将RGB图像转换成灰度值，再将灰度图转化成二进制图，转化的方法就是限定一个阀值，如果大于阀值则为1，小于阀值为0。将车牌二值化后还需对车牌灰度图像进行删除小对象处理。经过以上处理过程后可以得到黑底白字的车牌图片。

其中删除小对象处理主要是为了删除第二个字符和第三个字符之间的那一个点符号，为后续进行分割处理做好准备。



## **3.4字符分割**

完成牌照区域的定位后，再将牌照区域分割成单个字符，然后进行识别。字符分割一般采用垂直投影法。由于字符在垂直方向上的投影必然在字符间或字符内的间隙处取得局部最小值的附近，并且这个位置应满足牌照的字符书写格式、字符、尺寸限制和一些其他条件。利用垂直投影法对复杂环境下的汽车图像中的字符分割有较好的效果。

2007年颁布的我国车牌规范(普通中小型汽车)规定车牌总长440mm，牌照中的7个字符的实际总长为409mm 左右，宽140mm，每个字符45mm宽，90mm高，字符间距为10mm，其中第二个字符与第三个字符的间距较为特殊，为15.5mm，最后一个字符与第一个字符距边界25mm。这样，如果平均分配每个字符在牌照中占据的宽度，那么每个字符宽度为：width/7（width 为车牌图像的宽度）。但是，实际上，第二个第三个字符之间存在一个黑点，牌照左右两边与图像边缘也都有一定的宽度，所以每个字符的宽度应该小于width/7。考虑所有的情况，一般情况下最小的宽度为width/9。因此，字符的宽度可以从width/9 到width/7 之间渐进的变化得到,程序流程图为

车牌二值图像

切除周边空白

垂直投影

从左往右查找，垂直投影白色像素点累计小于10的值得wide

以水平区域（0，wide）剪切出字符

N Y

保存这个字符

判断是否为标准字符

将剩余部分保存

字符个数是否为7个

N

Y

源程序代码为：

d=fenge(d);

[m,n]=size(d);

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

k1=1;k2=1;s=sum(d);j=1; %假定图像为n×m矩阵,sum命令对图像的x维进行求和得到一个1×m矢量，每一点的值代表该y坐标上所有点光强之和。

while j~=n

while (s(j)==0)

j=j+1;

end

k1=j;

while(s(j)~=0&&j<=n-1)

j=j+1;

end

k2=j-1;

if k2-k1>=round(n/6.5)

[val,num]=min(sum(d(:,[k1+5:k2-5]))); %[Y,U]=min(A)：返回行向量Y和U，Y向量记录A的每列的最小值，U向量记录每列最小值的行号

d(:,k1+num+5)=0;

end

end

%在切割

d=fenge(d);

%切割出7个字符

y1=10;y2=0.25;flag=0;word1=[];

while(flag==0)

[m,n]=size(d);

left=1;wide=0;

while(sum(d(:,wide+1))~=0)

wide=wide+1;

end

if wide<y1 %左侧干扰

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

d=fenge(d);

else

temp=fenge(imcrop(d,[1 1 wide m]));

[m,n]=size(temp);

all=sum(sum(temp));

two\_thirds=sum(sum(temp([round(m/3):2\*round(m/3)],:)));

iftwo\_thirds/all>y2

flag=1;word1=temp; %第一个字符

end

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

end

end

% 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);

[m,n]=size(word1);

%将待识别字符的图像进行归一化处理，使其长宽与模板字符图像长宽一致，模板使用的是40\*20的图片尺寸，待识别字符图像归一化代码为：

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

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

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

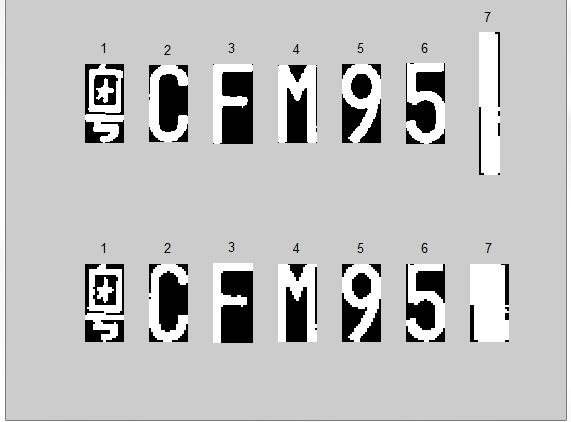
word4=imresize(word4,[40 20]);

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

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

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

字符切割且标准化后为



## **3.5字符模板匹配识别**

目前字符识别方法主要有神经网络识别法和模板匹配法。人工神经网络是模拟人脑思维功能和组织建立起来的数学模型，但总体来说其应用还是相对复杂的。模板匹配法是将从待识别的图象或图象区域f(i,j)中提取的若干特征量与模板T(i,j)相应的特征量逐个进行比较，计算它们之间规格化的互相关量，其中互相关量最大的一个就表示期间相似程度最高，可将图象归于相应的类。也可以计算图象与模板特征量之间的距离，用最小距离法判定所属类。本系统使用的是较为简单的模板匹配法来识别字符的。

字符模板匹配法的流程为

建立模板字库表

读取分割出来的字符

第一个字符与模板中的汉字模板进行比对，对应的字库下标为37~40

第二个字符与模板中的字母模板进行比对，对应的字库下标为11~36

后五个字符与模板中的字母与数字模板进行比对，对应的字库下标为1~36

求带识别字符与模板字符的相关系数，相关系数越大，说明其相似度越大

识别完成，输出此模板对应的值

源程序代码为

Modelword=char(['0':'9' 'A':'Z' '粤京桂川']);%建立模板字符列表

SubTu=zeros(32,16); %建立一个32\*16的零数组，记录两图相减后的结果

L=1;

for n=1:7 %n记录1~7个字符

ns=int2str(n);

t=imread([ns,'.jpg']);%读取分割后的字符

P\_word=imresize(t,[32 16],'nearest');

P\_word=double(P\_word)>20;

if L==1 %如果L==1，则识别车牌上的第一个字符（汉字），汉字在模板字符列表的序号是37~40

wmin=37;

wmax=40;

elseif L==2

%如果L==2，识别第二个英文字符，英文在模板字符列表的序号是11~36

wmin=11;

wmax=36;

else L>=3

%%如果L>=3，识别的字符可能是数字或字母，在模板字符列表的序号是1~36

wmin=1;

wmax=36;

end

for W=wmin:wmax %当确定检验字符的类型后，在其序号范围内一个一个字符的匹配

fname=strcat('E:\MATLAB\字符模板\',Modelword(W),'.bmp');

%strcat连接字符串，得到第W个模板字符的文件名

M\_word=imread(fname); %读取模板字符

M\_word=double(M\_word)>1;

for a=1:32

%字符车牌图片统一化为40\*20后，从每一行，每一列开始一一与模板对应的位置像素值比较（相减）

for b=1:16

SubTu(a,b)=P\_word(a,b)-M\_word(a,b);

end

end

num=0;

for c=1:32

for d=1:16

if (SubTu(c,d) > 0 | SubTu(c,d) <0 )

%当SubTu!=0时，即分割后的字符有一个点与模板不同

num=num+1; %num记不同点的数量

end

end

end

Error(W)=num;

%将分割后的字符和第W个模板字符不同点数存在Error向量中

end

Error0=Error(wmin:wmax); %将Error向量中的值存在Error0中

MinError=min(Error0); %找出不同点中最少的那一个数值

findc=find(Error0==MinError);

%找出差值最小的一个模板字符的W（序号），即最符合的模板字符

Code(L\*2-1)=Modelword(findc(1)+wmin-1);

%将找的要输出的字符存放在code数组中

Code(L\*2)=' '; %输出的车牌号每个字符间留一个空格

L=L+1; %匹配下一个字符

end

figure(10),imshow(CP),title(['车牌号码：',Code],'Color','r');

%输出车牌图像和车牌号，字体为红色

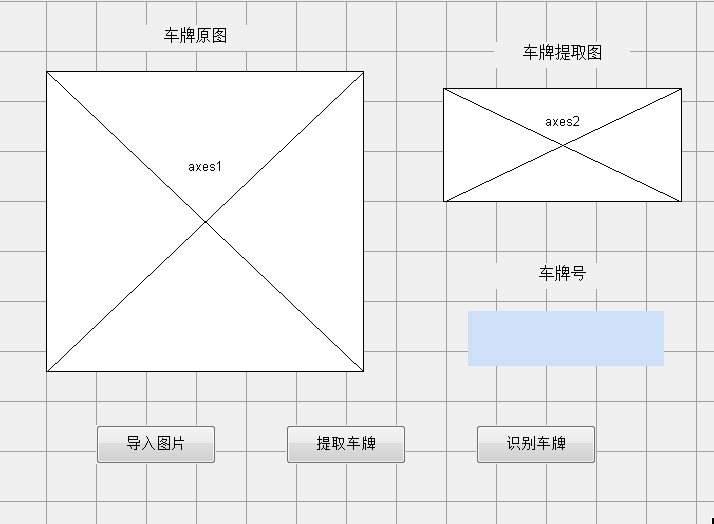
运行结果为



# 四.系统界面化

为了实现人机交互界面，本设计采用Matlab中的GUI设计，GUI的广泛应用是当今计算机发展的重大成就之一，它极大地方便 了非专业用户的使用。人们从此不再需要死记硬背大量的命令，取而代之的是可以通过窗口、[菜单](http://baike.baidu.com/view/102616.htm" \t "_blank)、按键等方式来方便地进行操作。

通过直接运用Matlab中给出的控件，我们设计出如下界面



完成布局后我们写出每个控件的回调函数：

1.导入图片：我们可以将上面讲到的车牌读取的内容写到按钮“导入图片”的回调函数下，并将读取到的原图设置为全局共享，最后将读取的图片显示在坐标轴1上：

注：axes(handles.axes1);%将当前使用的坐标设为坐标1

setappdata(handles.figure1,'I',I);%将图片I全局共享

2.车牌提取：我们可以将上面说的车牌预处理和车牌提取的相关代码写到按钮“提取车牌”的回调函数下，并将提取的车牌最后设置为全局变量，以便后续其他控件进行处理时方便调用。最后将提取到的车牌显示在坐标轴2上。

注：I=getappdata(handles.figure1,'I'); %读取全局共享的预处理后得车辆图像

axes(handles.axes2);%将当前使用的坐标设为坐标2

3．识别车牌：我们可以将上面说到的车牌图像预处理，字符分割和字符模板匹配等过程的程序代码写到按钮“识别车牌”的回调函数下，并将最后识别的结果显示在静态文本框中

4．菜单栏：同理将需要显示的图像进行全局处理后，将显示函数写在相应控件的回调函数下，最后显示在相应的界面即可。

运行结果为：



# 五.补充代码

在上述描述中以展示了本设计的核心代码，以下为一些补充代码；

1. 在字符分割的时候用到的两个函数分别为

（1）%子函数，提取字符

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

[m1,n1]=size(temp);

if wide<y1&&n1/m1>y2

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

if sum(sum(d))~=0

d=fenge(d);%切割出最小范围

else word=[];flag=1;

end

else

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

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

if sum(sum(d))~=0

d=fenge(d);flag=1;

else d=[];

end

end

end

result=d;

（2）%创建切割函数

function e=fenge(d)

[m,n]=size(d); %处理后的车牌图像大小为m\*n

%初始化

top=1;

bottom=m;

left=1;

right=n;

%从图像上下边分别往中间开始，检测到有字（d(x,y)=1）时停止

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

top=top+1;

end

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

bottom=bottom-1;

end

%从图像左右边分别往中间开始，检测到有字（d(x,y)=1）时停止

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

left=left+1;

end

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

right=right-1;

end

%切割字符的长宽

dd=right-left;

hh=bottom-top;

%返回切割后的图像

e=imcrop(d,[left top ddhh]); %imcrop('图象名',[x起点，y起点，x宽度，y宽度]) %imcrop(x,[20 20 50 50])从（20，20）开始剪下50\*50的区域

菜单栏的视图的回调函数

原图像预处理

I=getappdata(handles.figure1,'I');

I1=getappdata(handles.figure1,'I1');

I2=getappdata(handles.figure1,'I2');

I3=getappdata(handles.figure1,'I3');

I4=getappdata(handles.figure1,'I4');

I5=getappdata(handles.figure1,'I5');

figure(5),subplot(2,3,1),imshow(I);title('1.原始车牌图像');

figure(5),subplot(2,3,2),imshow(I1);title('2.灰度图');

figure(5),subplot(2,3,3),imshow(I2);title('3.robert算子边缘检测');

figure(5),subplot(2,3,4),imshow(I3);title('4.腐蚀后的图像');

figure(5),subplot(2,3,5),imshow(I4);title('5.平滑图像的轮廓');

figure(5),subplot(2,3,6),imshow(I5);title('6.从图像中删除小对象');

车牌信息

globalBlue\_x;

globalBlue\_y;

I1=getappdata(handles.figure1,'I1');

CP=getappdata(handles.figure1,'CP');

I6=getappdata(handles.figure1,'I6');

I7=getappdata(handles.figure1,'I7');

figure(2),subplot(2,3,1),imshow(I1);title('灰度图');

figure(2),subplot(2,3,2),imhist(I1);title('灰度直方图');

figure(2),subplot(2,3,4),imshow(CP);title('车牌原图');

figure(2),subplot(2,3,5),plot(Blue\_x),title('行向蓝色点统计');

figure(2),subplot(2,3,6),plot(Blue\_y),title('纵向蓝色点统计');

车牌预处理

b=getappdata(handles.figure1,'b');

d=getappdata(handles.figure1,'d');

d1=getappdata(handles.figure1,'d1');

d2=getappdata(handles.figure1,'d2');

CP=getappdata(handles.figure1,'CP');

figure(8),subplot(2,2,1),imshow(CP),title('1.车牌原图像');

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

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

figure(8),subplot(2,2,4),imshow(d2);title('4.从图像中删除小对象 ');

字符分割

word1=getappdata(handles.figure1,'word1');

word2=getappdata(handles.figure1,'word2');

word3=getappdata(handles.figure1,'word3');

word4=getappdata(handles.figure1,'word4');

word5=getappdata(handles.figure1,'word5');

word6=getappdata(handles.figure1,'word6');

word7=getappdata(handles.figure1,'word7');

figure(9),subplot(1,7,1),imshow(word1),title('1');

figure(9),subplot(1,7,2),imshow(word2),title('2');

figure(9),subplot(1,7,3),imshow(word3),title('3');

figure(9),subplot(1,7,4),imshow(word4),title('4');

figure(9),subplot(1,7,5),imshow(word5),title('5');

figure(9),subplot(1,7,6),imshow(word6),title('6');

figure(9),subplot(1,7,7),imshow(word7),title('7');

# 六．总结

本此课程设计为基于MatlabGUI的车牌识别系统，在本文中简述了有关车牌识别技术的四个方面的内容：

1. 在第一部分内容中简单介绍了车牌识别在现实生活中对交通管理领域的重要意义，现如今国内外对车牌识别技术的研究成果，以及车牌识别技术未来的发展方向，使车牌识别准确度进一步提高，让其使用过程简单有效。
2. 在第二部分中本文概述了车牌识别的简易流程：**一**个完整的车牌号识别系统要完成从图像采集到字符识别输出，过程相当复杂，基本可以分成硬件部分跟软件部分，硬件部分包括系统触发、图像采集，软件部分包括图像预处理、车牌位置提取、字符分割、字符识别四大部分。
3. 第三部分内容详细的介绍了车牌识别过程的具体步骤以及其中需要用到的函数和具体源程序代码，在Matlab中提供了许多库函数以供用户使用，我们可以直接调用函数进行图像的处理，其中我们需要了解每个函数的基本用法和参数设定，通过不断调整参数达到最佳效果。
4. 第四部分内容中简单介绍了Matlab的GUI用法，将上面的车牌识别过程的源代码封装到可视化界面中，实现便于操作的人机交互界面。

本次课程设计总体来说比较成功，通过本次设计让我初步掌握了有关使用Matlab进行图像处理的知识，了解了车牌识别的具体过程及原理。在整个系统设计过程中会遇到大大小小各种问题，例如对于处在特定环境的车牌我们要找到一套适合的参数进行处理，需要不断修改参数达到最佳效果。还有就是在模板匹配时有些模板并不适合会出现一些识别错误，需要我从新定制模板等等一些问题。但通过解决不断解决这些问题，让我对图像处理有了更深的认识的同时锻炼了自己处理问题的能力。