# 问题的描述

我们在使用爬虫技术去爬取网页信息（北京科技大学电子支付平台登陆界面https://pay.ustb.edu.cn/fontuserLogin）时总会遇到验证码，这严重阻碍了爬虫爬取信息的进程。如果程序能够自动识别验证码并且完成输入，我们的爬虫就能突破这层障碍，继续搜集我们想要得到的内容。因此，如何利用所学专业知识识别验证码则显得尤为重要。

# 问题分析及模型建立

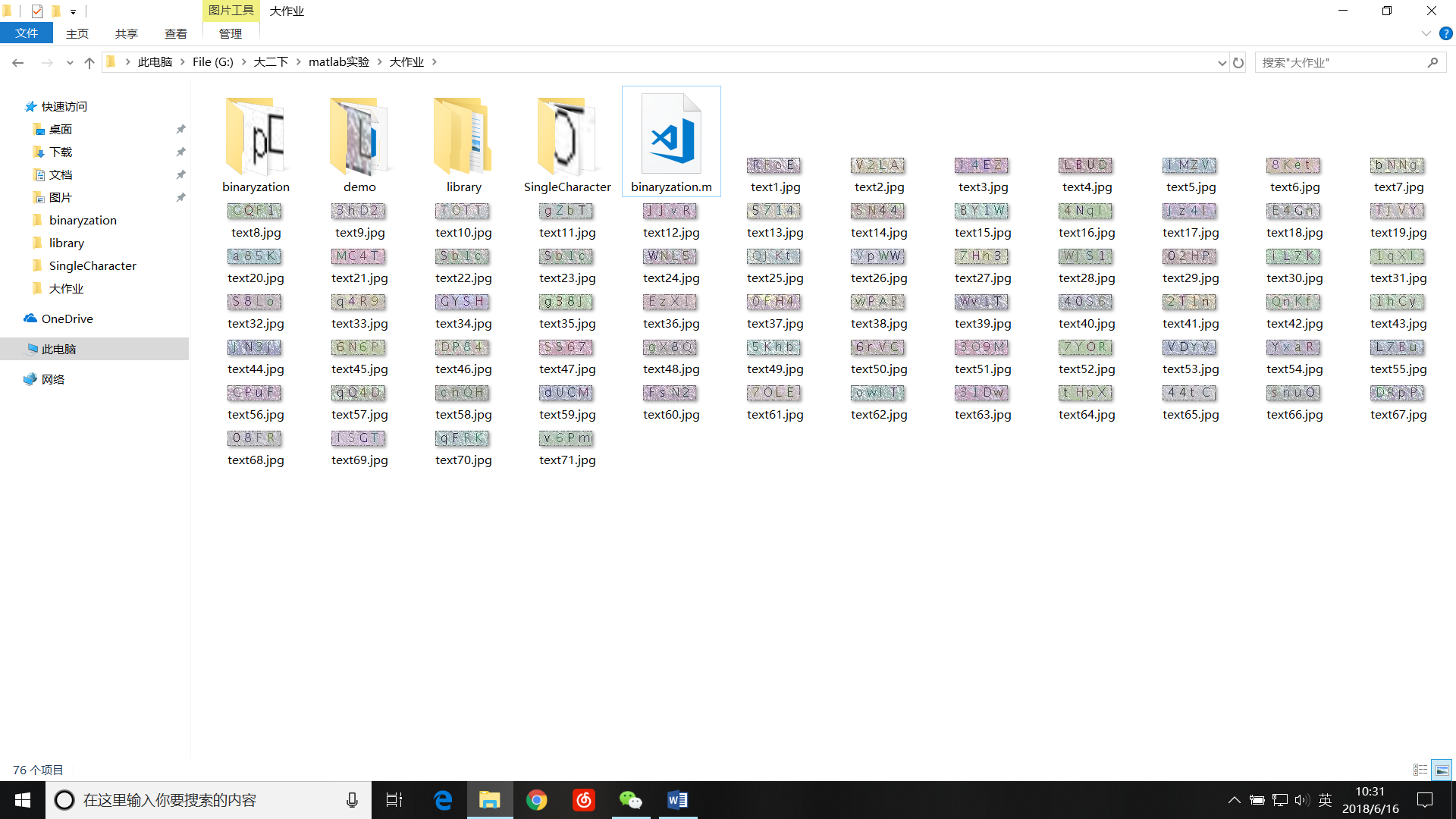
利用信号与系统所学知识，我们可以对图像的每个像素进行滤波处理，得到黑白图像，再经过进一步处理从而得到一个二值化的二维矩阵。因此按照上述流程，我们利用Matlab将所获取的验证码进行二值化、分割、对比、匹配四部操作即可完成识别工作。

模型：每张验证码包含四个字符，通过分割图像而将每个字符单独提取出来进行比对。

# 问题求解

1. 首先我们需要获取到足够多的验证码

通过Chorme的F12调用出浏览器调试模式，找到获取北京科技大学支付界面验证码的url为 <https://pay.ustb.edu.cn/authImage>



这里我们获取了71张验证码，依次命名为textx.jpg。他们都是70X20的jpg图像。

1. 批量处理所获取的验证码

通过分析我们发现北京科技大学电子支付平台的验证码共包含四个字符，他们可以是大写字母、小写字母和数字。

二值化：

分割图形：

通过分析我们发现，验证码中所包括的四个字符位置是固定的，因此我们按照固定位置进行分割

代码：

global MaxPictureNum;%处理的验证码数量

MaxPictureNum=71;

for i=1:MaxPictureNum

s=['text',num2str(i),'.','jpg'];

img = imread(s);

% figure,imshow(img);

%二值化处理

img = im2bw(img);%图像转换为2值图

%处理边界像素点

for j=1:70

img(1,j)=1;

img(20,j)=1;

end

for j=1:20

img(j,1)=1;

img(j,70)=1;

end

% img=medfilt2(img,[1 1]);%中值滤波，这里处理后效果较好不需要使用

% figure,imshow(img);

%保存二值化后的文件

s=['.\binaryzation\','text',num2str(i),'.','jpg'];

imwrite(img,s);

%切割单个字符

for j=1:4

temp=img(:,(j-1)\*15+6:(j-1)\*15+19);

s=['.\SingleCharacter\',num2str(i),'\_',num2str(j),'.','jpg'];

imwrite(temp,s);

end

shuzhi=zeros(20,70);

for a=1:20

for b=1:70

if(img(a,b)==1)

shuzhi(a,b)=0;

else

shuzhi(a,b)=1;

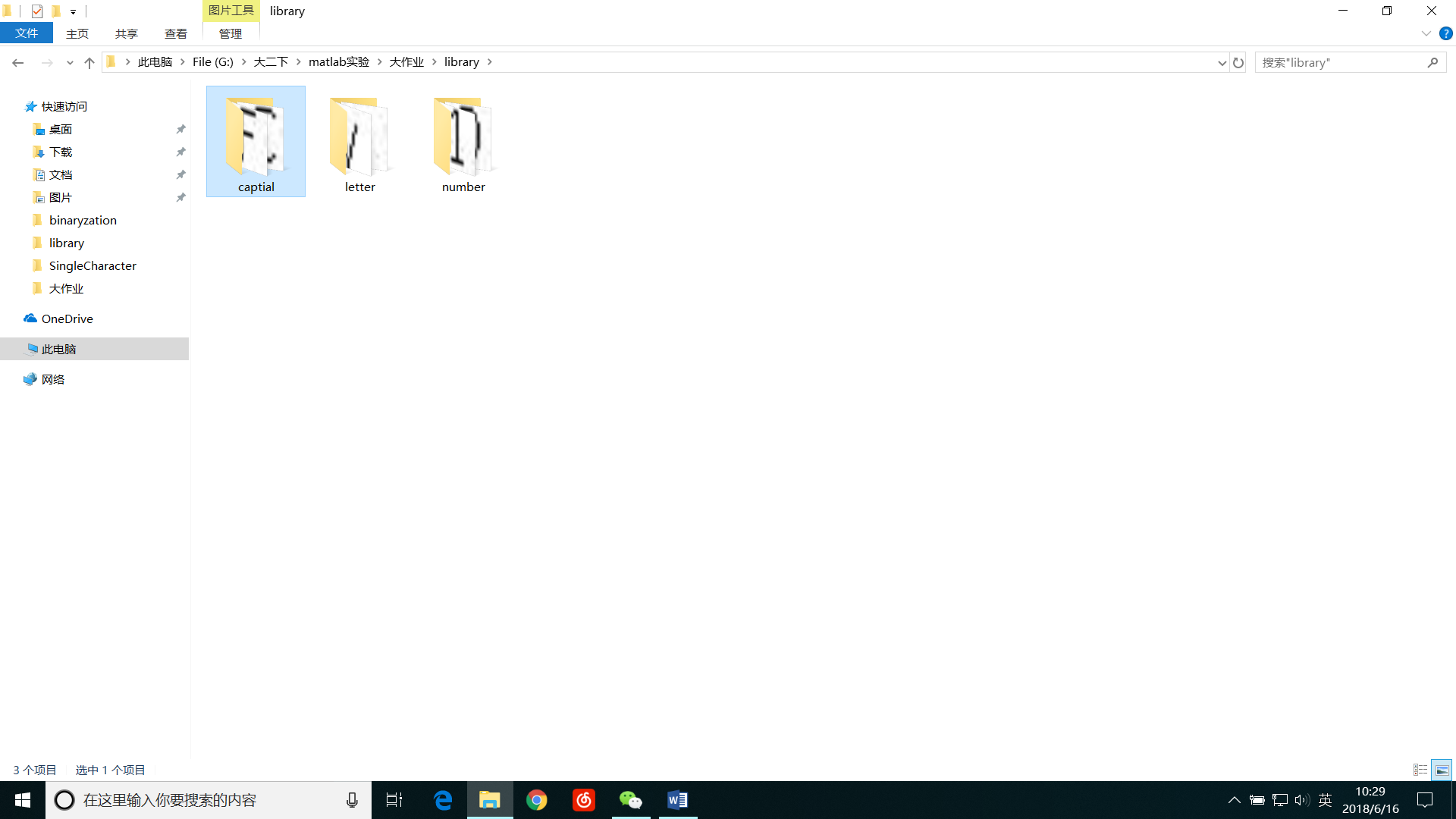
end

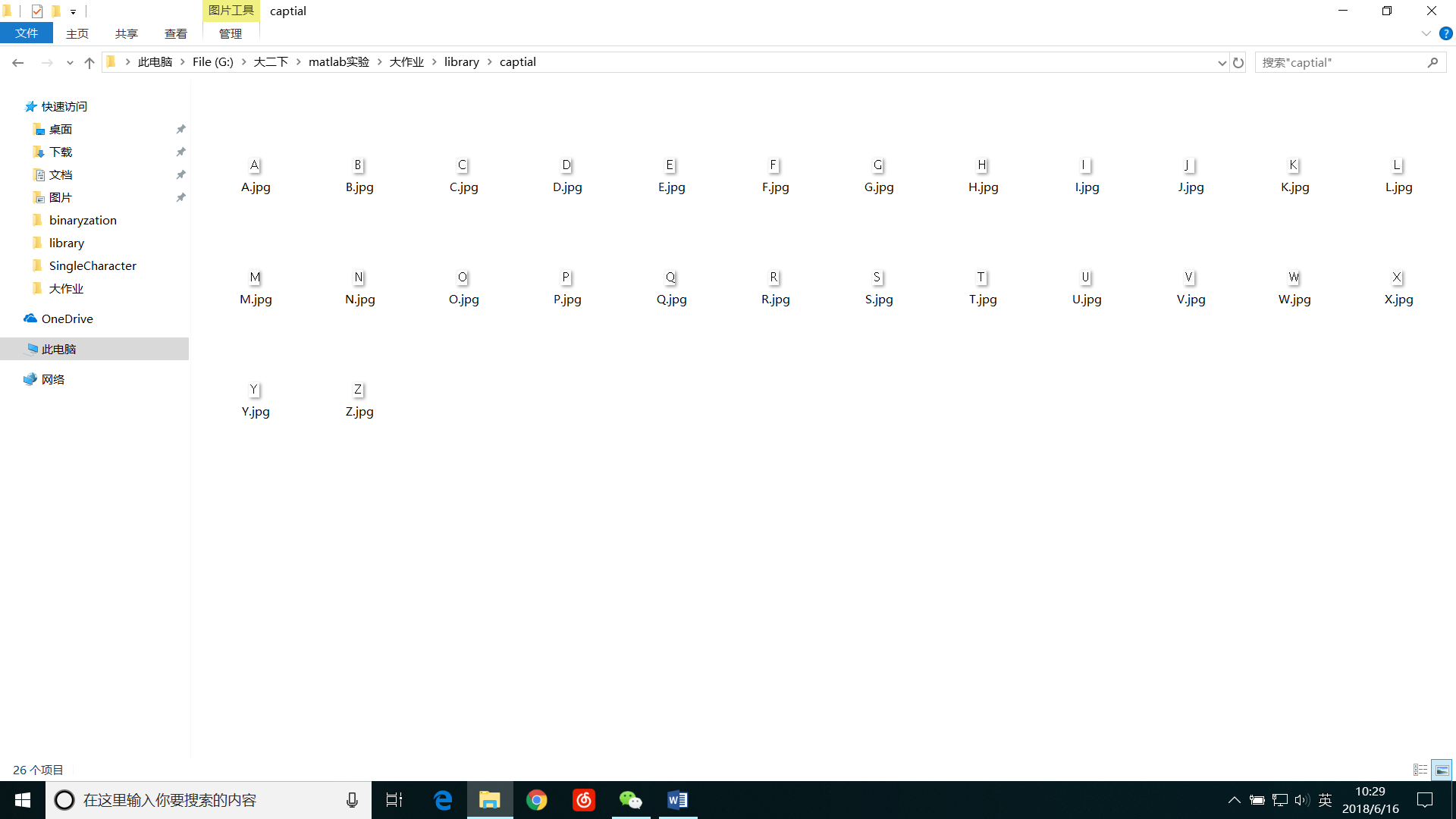
end

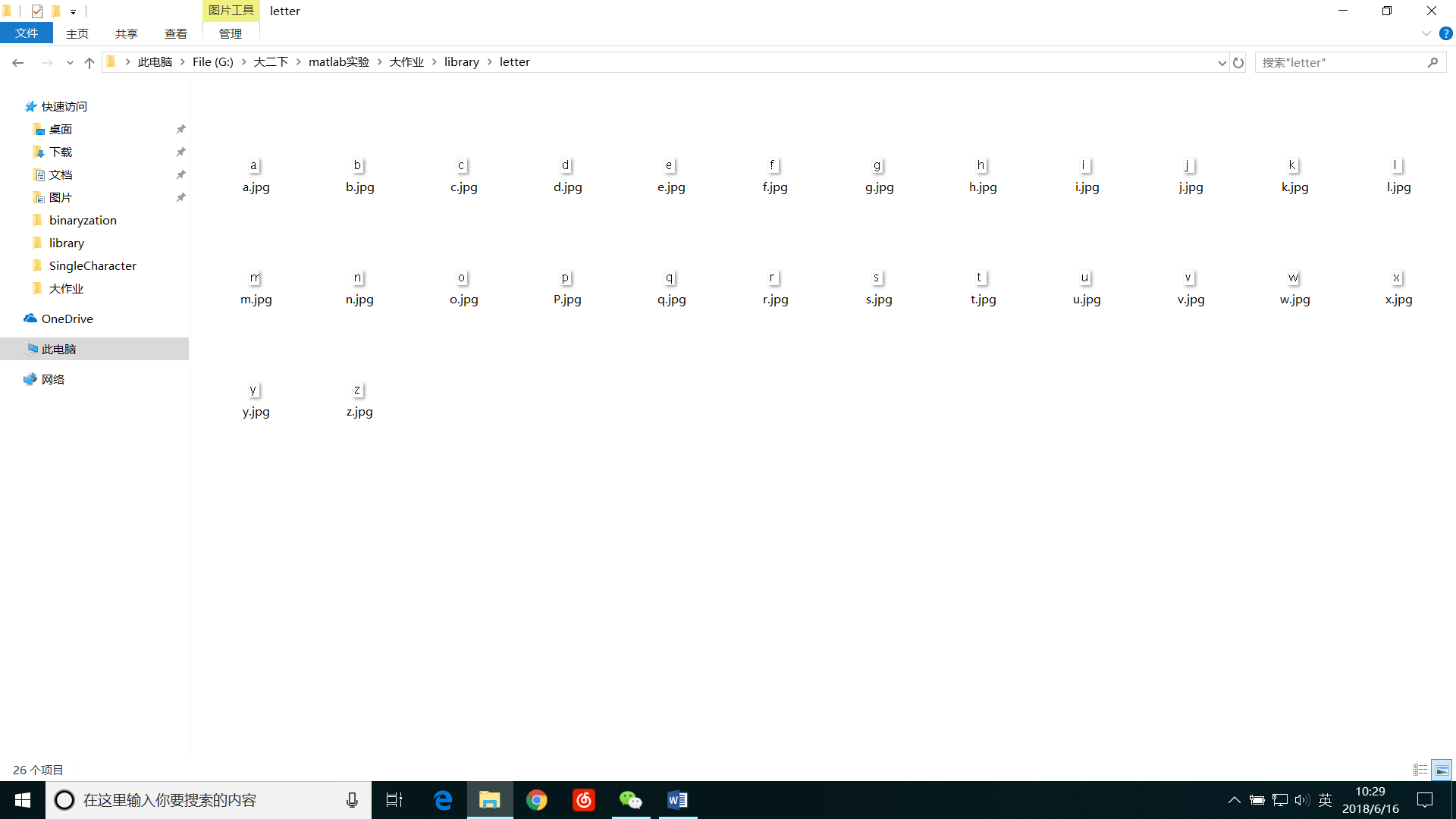
end

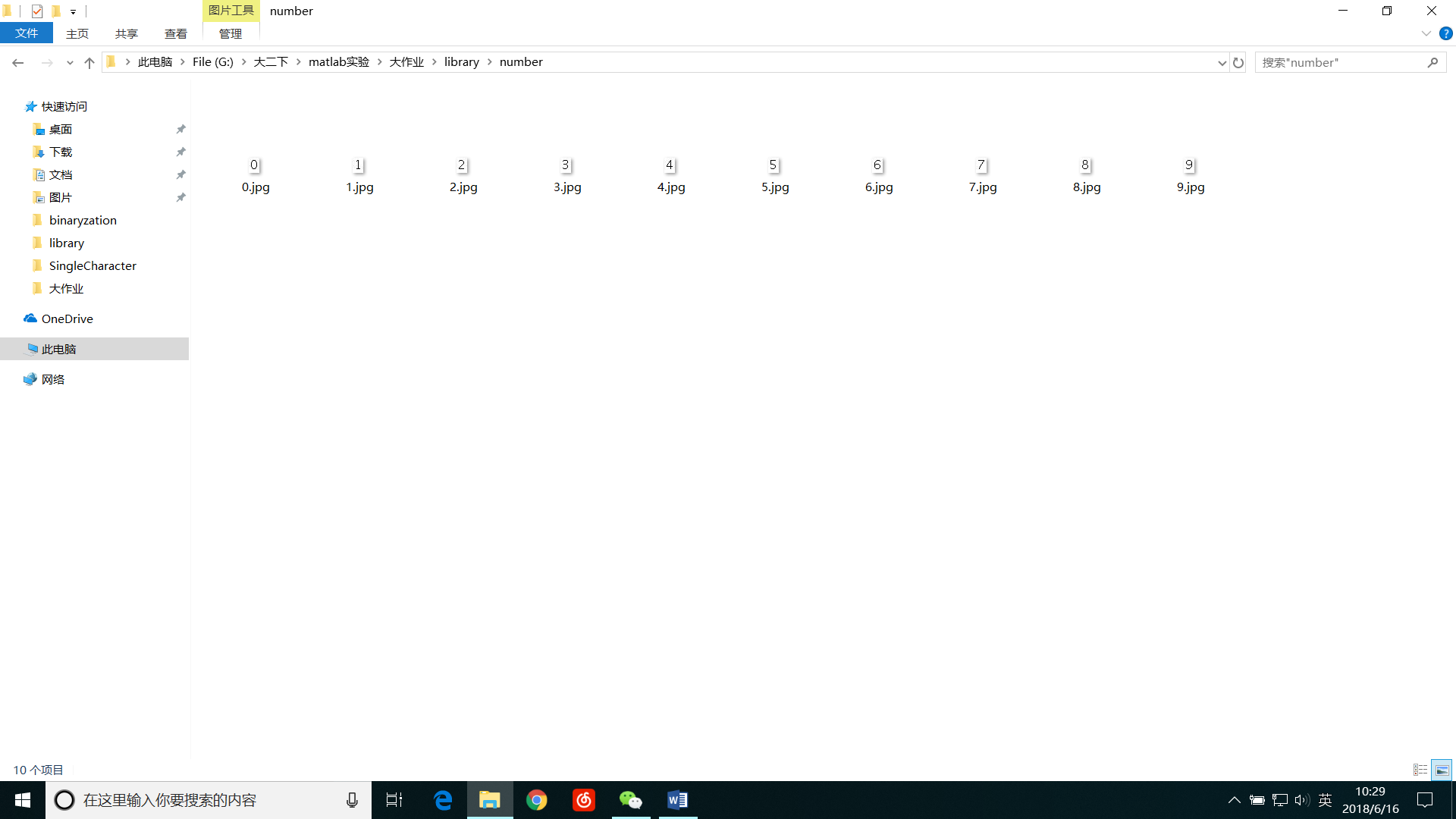
end

3）根据上一步骤得到的单个字符来建立一个包含所有大小写字母和数字的字符库，以供给对比、分析使用









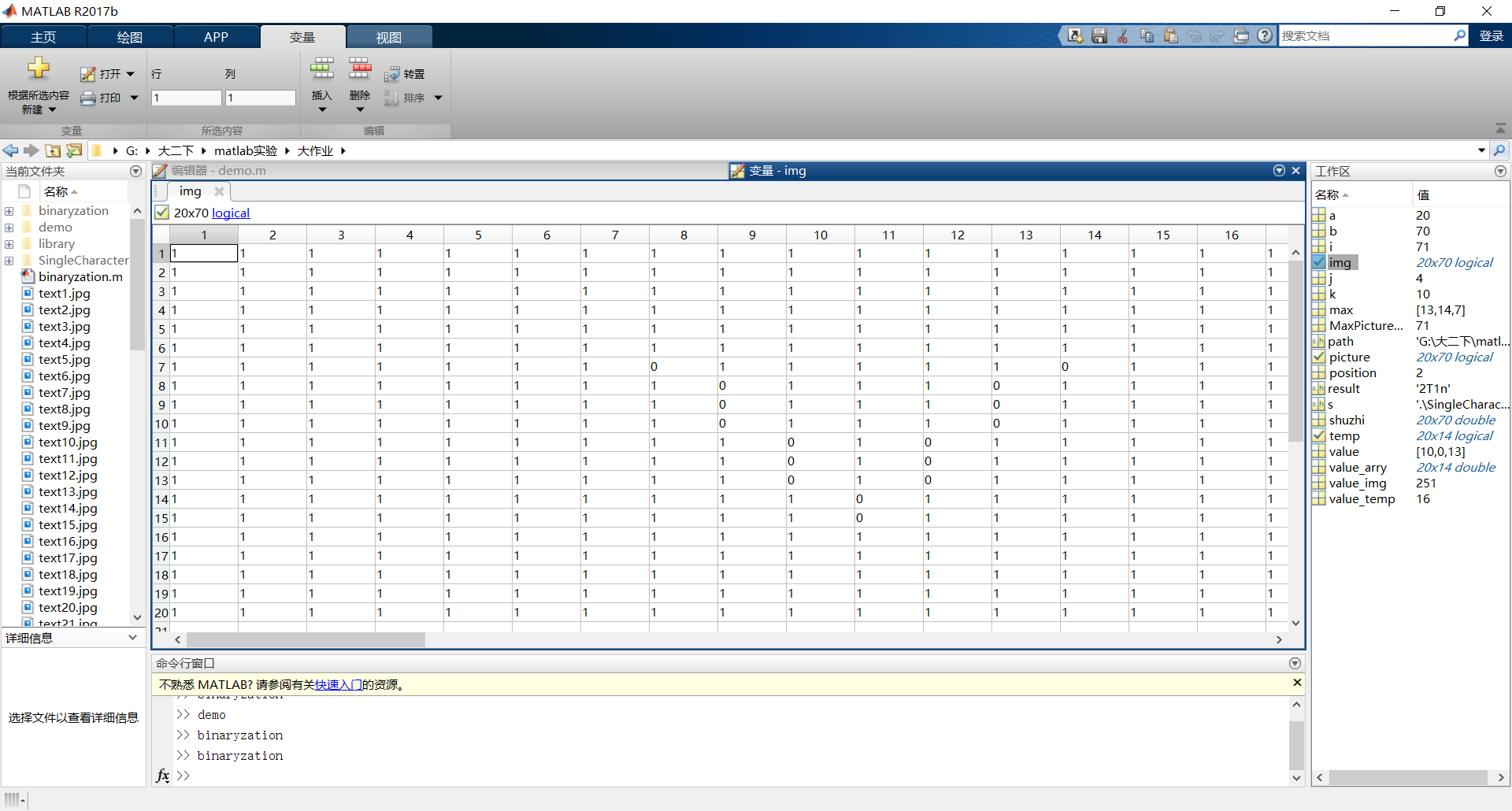
1. 对比匹配，找出最符合的字符

流程分析：

首先将待识别验证码进行二值化、分割处理以得到单个字符，再将得到的单个字符与上一步所建立的字符库里的文件进行一一比对，得到的最佳匹配项即为识别结果。

关键算法-比对匹配：

经过处理后得到的单个字符是二维矩阵的形式，同时字符库里的文件读取出来也是二维矩阵的形式，且他们都是二值化的（非0即1）。



通过将两个数组进行比对，记录其中的相同元素个数，即可得到他们的相似程度。最后根据相似程度来判定待识别字符究竟是什么。并将结果保存在字符串result中。

代码：

result='0000';

s='text.jpg';

temp=zeros(20,14);

max=[0 0 0];

value=[100 100 100];

value\_arry=zeros(20,14);

picture=imread(s);

picture=im2bw(picture);

%处理边界像素点

for j=1:70

picture(1,j)=1;

picture(20,j)=1;

end

for j=1:20

picture(j,1)=1;

picture(j,70)=1;

end

%切割单个字符

for j=1:4

max=[0 0 0];

value=[100 100 100];

temp=picture(:,(j-1)\*15+6:(j-1)\*15+19);%当前待识别字符

s=['.\singleCharacter\',num2str(j),'.','jpg'];

imwrite(temp,s);

%大写字母判定

path='G:\大二下\matlab实验\大作业\library\captial';

for k=1:26

s=[path,'\',char(k+64),'.jpg'];

img=imread(s);

img=im2bw(img);%库文件

%figure,imshow(img);

value\_arry=temp.\*img;

value\_img=sum(img (:));

value\_temp=abs(sum(value\_arry (:))-value\_img);

if (value\_temp<=value(1))

value(1)=value\_temp;

max(1)=k;

end

end

%小写字母判定

path='G:\大二下\matlab实验\大作业\library\letter';

for k=1:26

s=[path,'\',char(k+96),'.jpg'];

img=imread(s);

img=im2bw(img);

%figure,imshow(img);

value\_arry=temp.\*img;

value\_img=sum(img (:));

value\_temp=abs(sum(value\_arry (:))-value\_img);

if (value\_temp<=value(2))

value(2)=value\_temp;

max(2)=k;

end

end

%数字判定

path='G:\大二下\matlab实验\大作业\library\number';

for k=1:10

s=[path,'\',char(k+47),'.jpg'];

img=imread(s);

img=im2bw(img);

%figure,imshow(img);

value\_arry=temp.\*img;

value\_img=sum(img (:));

value\_temp=abs(sum(value\_arry (:))-value\_img);

if (value\_temp<=value(3))

value(3)=value\_temp;

max(3)=k;

end

end

position=1;

for a=1:3

if(value(a)<=value(position))

position=a;

end

end

if(position==1)

result(j)=char(64+max(position));

end

if(position==2)

result(j)=char(96+max(position));

end

if(position==3)

result(j)=char(47+max(position));

end

end

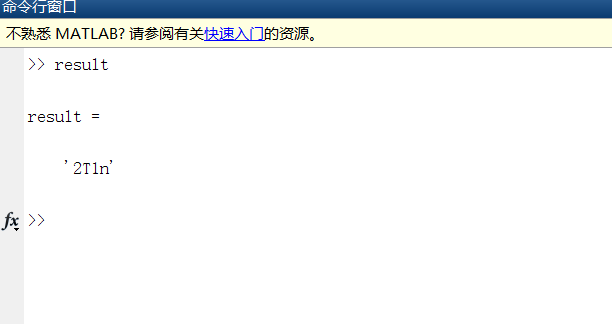
# 结果分析及心得体会

结果分析：

待识别验证码:



结果：



待识别验证码：



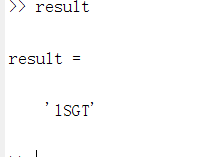
结果：



待识别验证码：



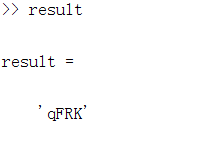
结果：



待识别验证码：



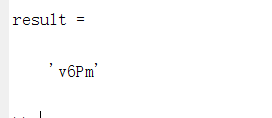
结果：



待识别验证码：



结果：



通过对71张验证码（284个字符）的识别，程序的正确识别率为97.8%，已经满足爬虫突破验证码障碍的要求。

心得体会

通过结合信号与系统以及数学实验两门课程，我们为验证码的机器识别提出了一种解决方案。我们意识到Matlab在数据处理方面的优势：语法简单、丰富的库函数可供调用、调试过程简便等等。但是我们也发现，我们所提供的验证码识别解决方案只能解决一些特定的验证码，而对于出现字符粘连、字符抖动的复杂验证码表现效果有待改善。通过查阅网上资料，我们发现人工智能领域里的深度学习能够简单高效的解决这些问题，这里笔者不再赘述。

# 参考文献

知网：

1. 验证码识别的设计与实现\_马宇卓\_哈尔滨工程大学
2. 一种扭曲粘连字符验证码识别方法\_尹龙\_模式与人工智能
3. 数字验证码识别算法的研究和设计\_王虎\_计算机工程与应用

博客：

参考众多CSDN博主的博客