Thinker&walker

简单的验证码识别

扩展

验证码又称全自动区分计算机和人类的图灵测试Completely Automated Public Turing test to tell Computers and Humans Apart，简称CAPTCHA），俗称验证码，是一种区分用户是计算机和人的公共全自动程序。在CAPTCHA测试中，作为服务器的计算机会自动生成一个问题由用户来解答。这个问题可以由计算机生成并评判，但是必须只有人类才能解答。由于计算机无法解答CAPTCHA的问题，所以回答出问题的用户就可以被认为是人类。(from wikipedia)

扩展

登录名： 用户名／邮箱

密码：

忘记用户名/密码？

验证码：

验证码如何使用？

登录 快速注册

扩展

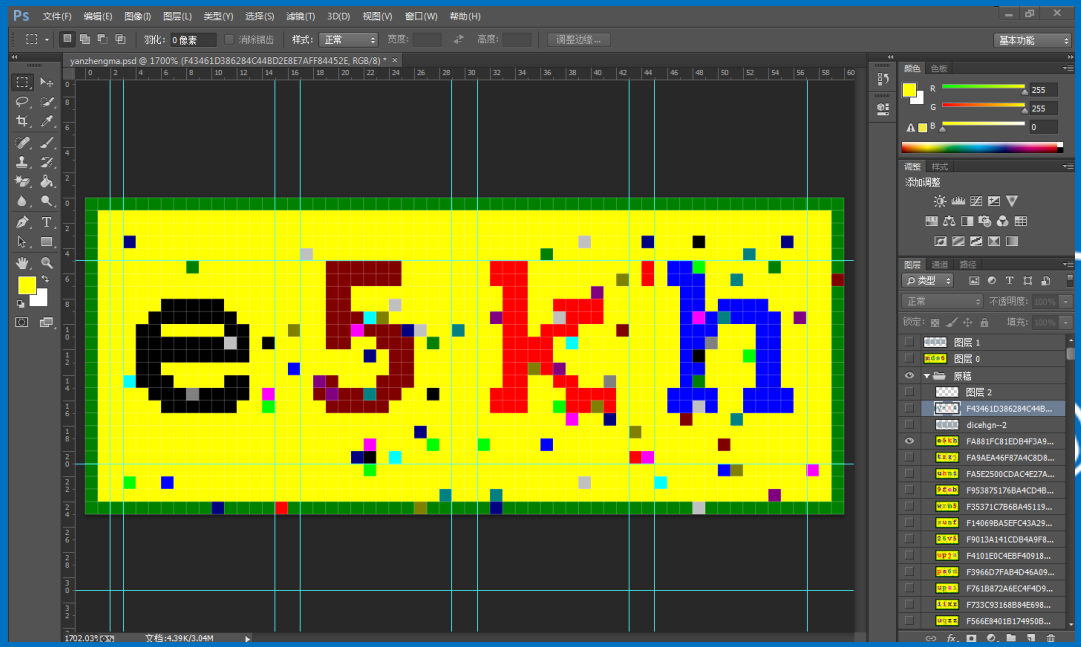

https://ipv4.google.com/sorry/IndexRedirect?continue=https://www.go
百度
导入书签
请键入下图显示的字符以继续操作：
提交
关于此网页
我们的系统检测到您的计算机网络中存在异常流量。此网页用于确认这些请求是由您而不是自动程序发出的。为什么会这样？
IP 地址：[redacted]
时间：2015-11-25T10:44:11Z
网址：https://www.google.com.hk/search?q=inurl:%22php%3Fid%3D%22&safe=strict&ei=v5BVVq7KIYKIoQSD14CABw&start=10&sa=N&biw=1366&bih=608

图片分析

--图片是不是有相同点？

工具

photoshop CC



图片分析

图片分析

1.找规律

图片尺寸大小？

字符位置？？

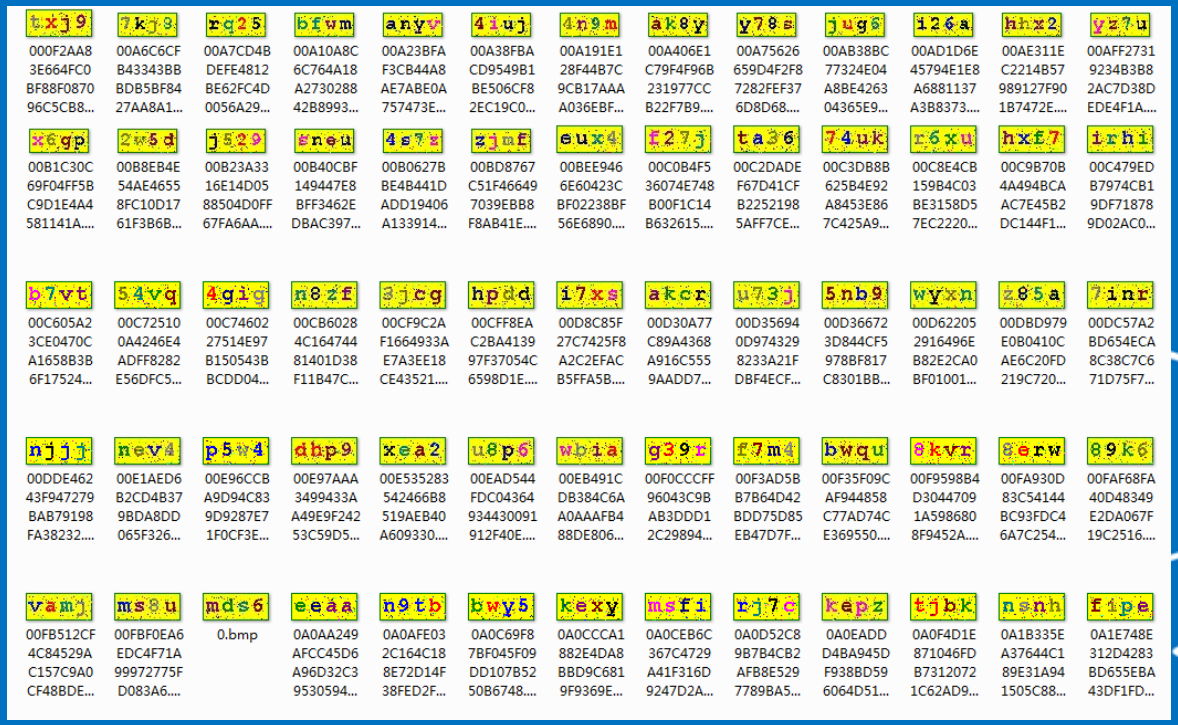
字符尺寸??

字符种类？？

图片分析

图片分析

1.找规律



图片分析

图片分析

2.分析

2.分析

图片分析

图片分析

2.分析

图片尺寸大小 25*60

字符位置—固定

字符尺寸—固定

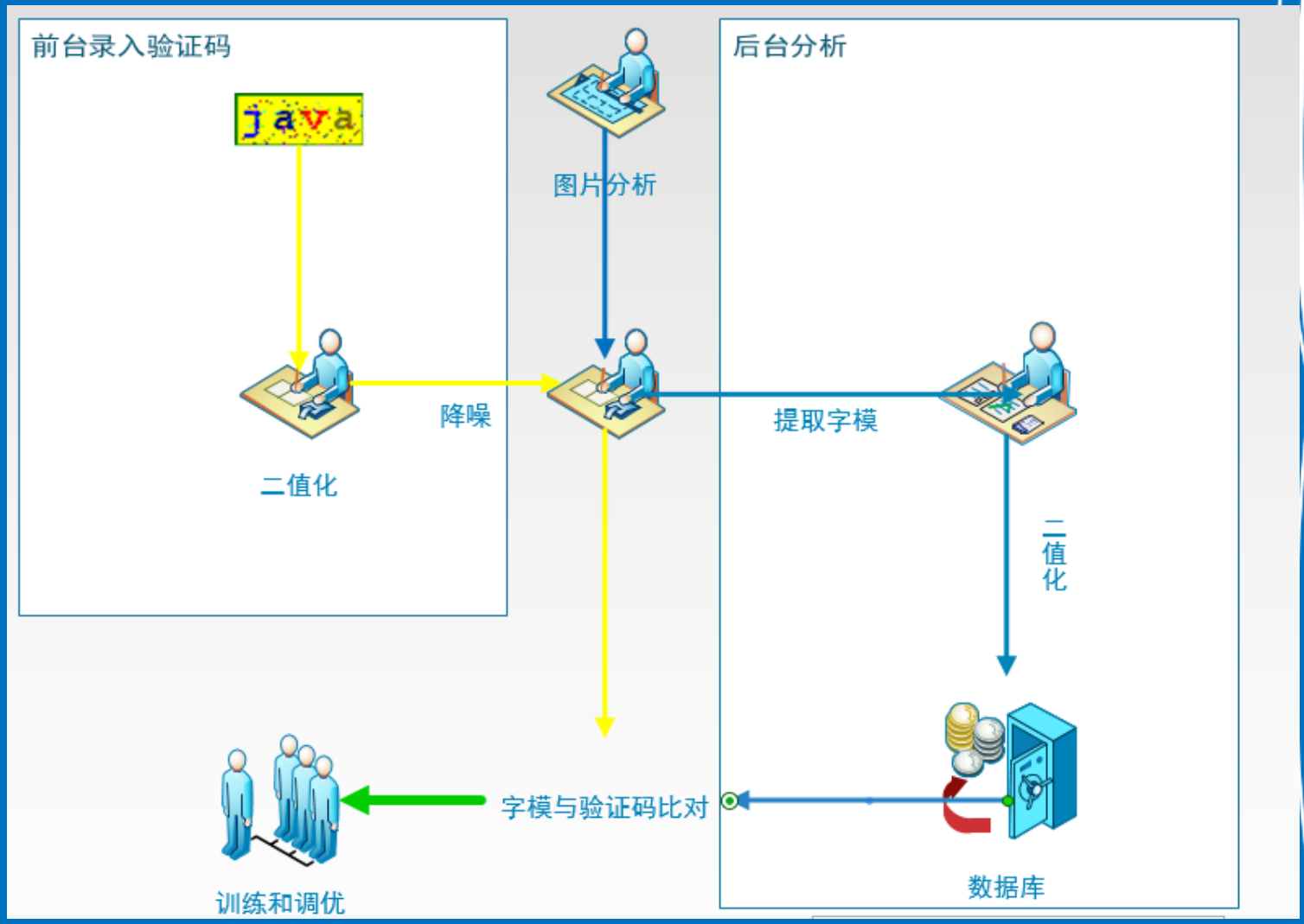
字符种类2-9a-z没有l和0（欧）

图片分析

图片分析

2.分析

处理方法
--流程



处理方法
--图片降噪

图片分析

去除无用点

工具：

1.photoshop CC

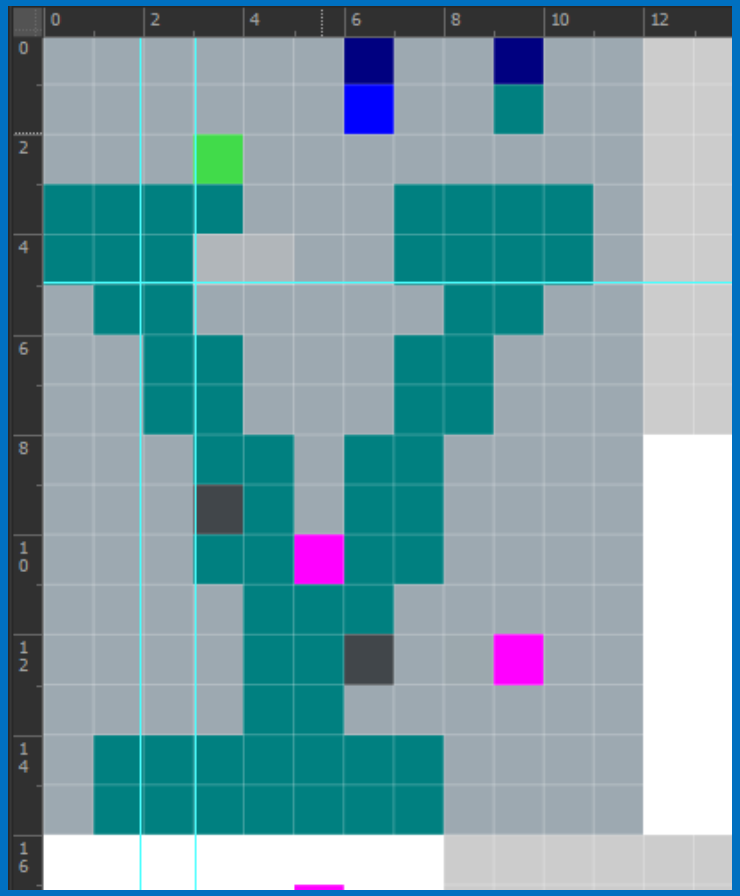
语言：

java or python

直接ps去噪点

处理方法
--图片降噪

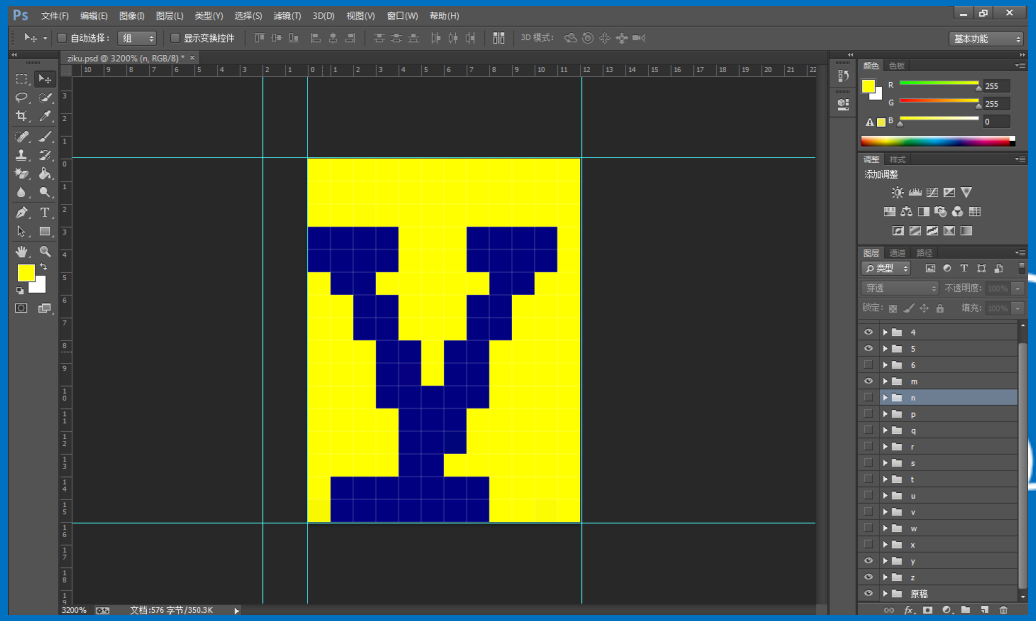
去除无用点



处理方法
--图片降噪

图片分析

去除无用点

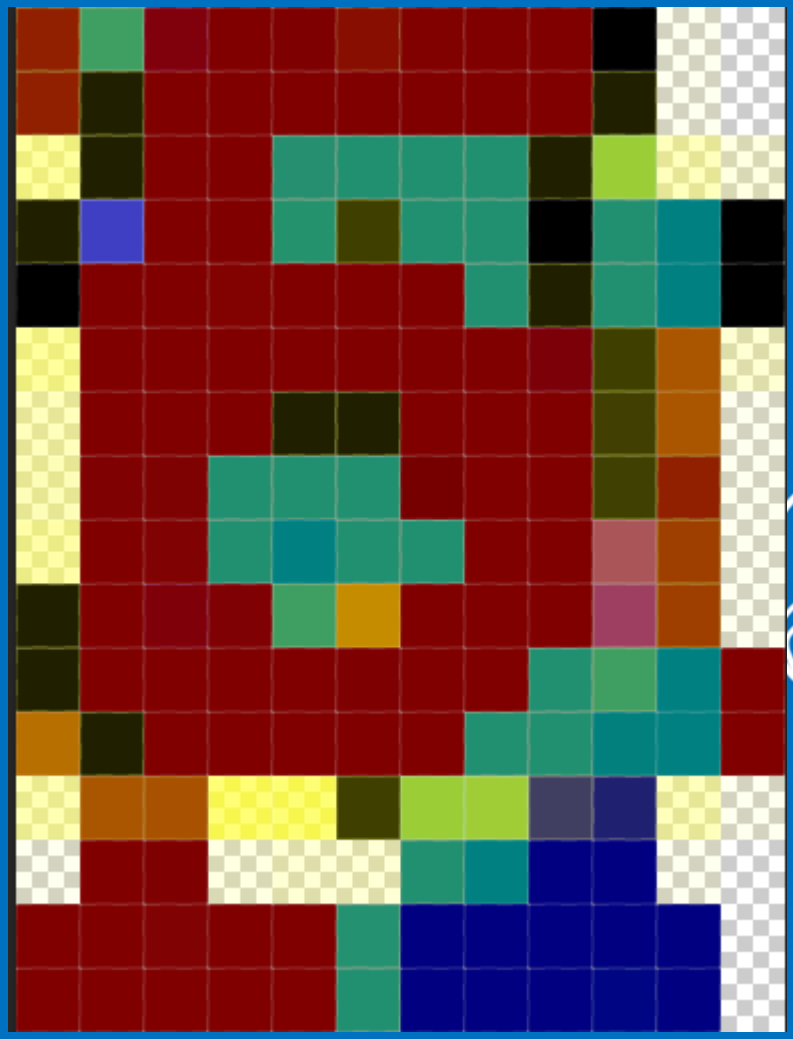


处理方法
--提取字模

去除无用点

处理方法
--提取字模

去除无用点



处理方法
--二值化

去除无用点

工具：

1.netbeans 

2.Editplus 

语言

1.java or python(PIL库)

处理方法
--二值化

图片分析

把图像转换成二维数组

```
47  public static int[][] getErZhiHua( BufferedImage bi) throws IOException {
48      /// File file = new File("E:\\png.png");
49
50  //    BufferedImage bi = ImageIO.read(file);
51      int h = bi.getHeight();
52      int w=bi.getWidth();
53      int a[][] = new int[h][w];
54      int rgb=bi.getRGB(0, 0);
55      int x=0;
56     // int arr[][]=new int[w][h];
57     // int x = bi.getRGB(2, 4);
58     // System.out.println(x);
59      for(int i=0;i<h;i++){
60      for(int j=0;j<w;j++){
61       a[i][j]=  bi.getRGB(j, i)==-256?0:1;
62
63         }
64
65      }
66      return a;
67
68
```

Thinker&Walker

处理方法
--二值化

图片分析

把图像转换成二维数组

```
000000000000
000000000000
001000000000
111100011110
111100011110
011000001100
001100011000
001100011001
000110110000
000110110000
000111110000
000011100000
000011100000
000011000000
011111110000
111111110010
```

处理方法
--二值化

图片分析

把图像转换成二维数组

处理方法
--二值化

把图像转换成二维数组

处理方法
--二值化

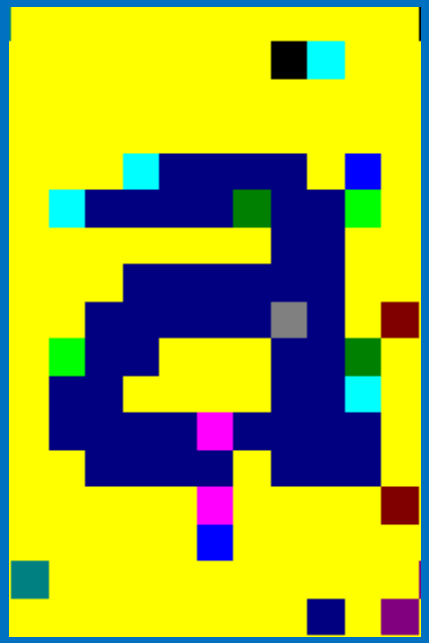
图片分析

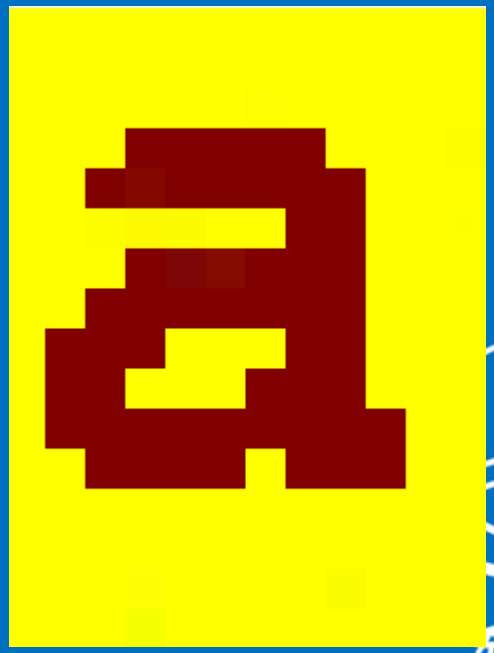
把图像转换成二维数组

```
148    public static final int ZI_7 [][] = {
149    {0,1,1,1,1,1,1,1,1,0,0,0},
150    {0,1,1,1,1,1,1,1,1,0,0,0},
151    {0,1,1,0,0,0,0,1,1,0,0,0},
152    {0,0,0,0,0,0,1,1,1,0,0,0},
153    {0,0,0,0,0,0,1,1,0,0,0,0},
154    {0,0,0,0,0,0,1,1,0,0,0,0},
155    {0,0,0,0,0,1,1,1,0,0,0,0},
156    {0,0,0,0,0,1,1,0,0,0,0,0},
157    {0,0,0,0,0,1,1,0,0,0,0,0},
158    {0,0,0,0,1,1,0,0,0,0,0,0},
159    {0,0,0,0,1,1,0,0,0,0,0,0},
160    {0,0,0,0,1,1,0,0,0,0,0,0},
161    {0,0,0,0,0,0,0,0,0,0,0,0},
162    {0,0,0,0,0,0,0,0,0,0,0,0},
163    {0,0,0,0,0,0,0,0,0,1,0,0},
164    {0,0,0,0,0,0,0,0,0,0,0,0}
165    };
```

处理方法
--识别训练

1.相似度

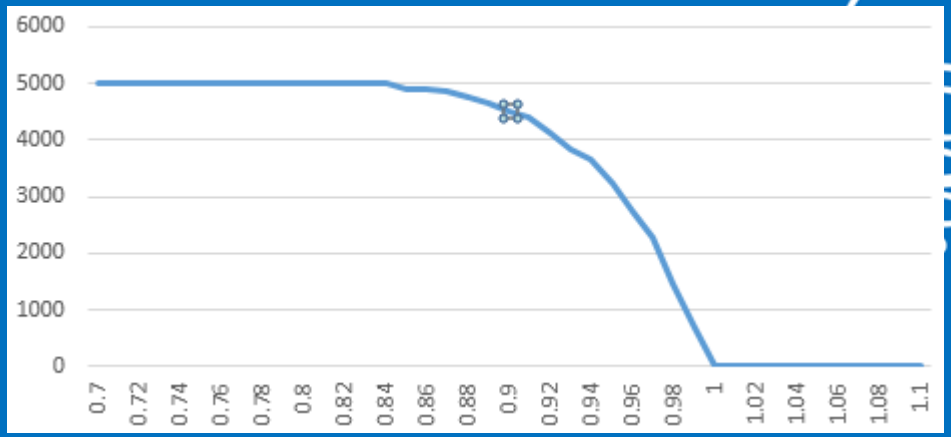




处理方法
--识别训练

1.相似度

序号	相似度	未识别	错误率
	0.88	248	
	0.89	339	
	0.9	484	
	0.88	248	



处理方法
--识别训练

1.相似度

0.89

处理方法
--识别训练

图片分析

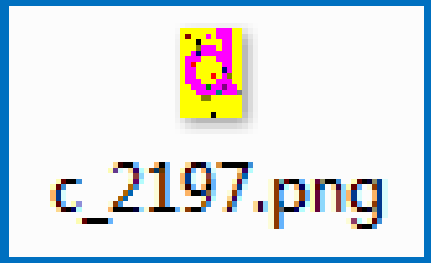
1.相似度

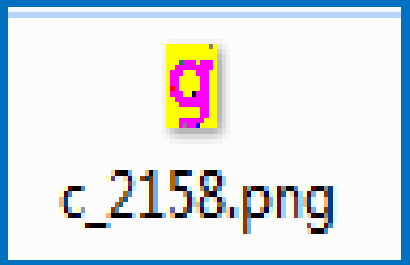
找错

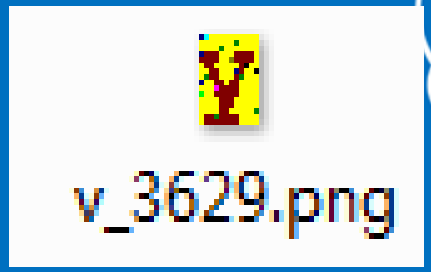
处理方法
--识别训练

1.相似度

错误字符的处理

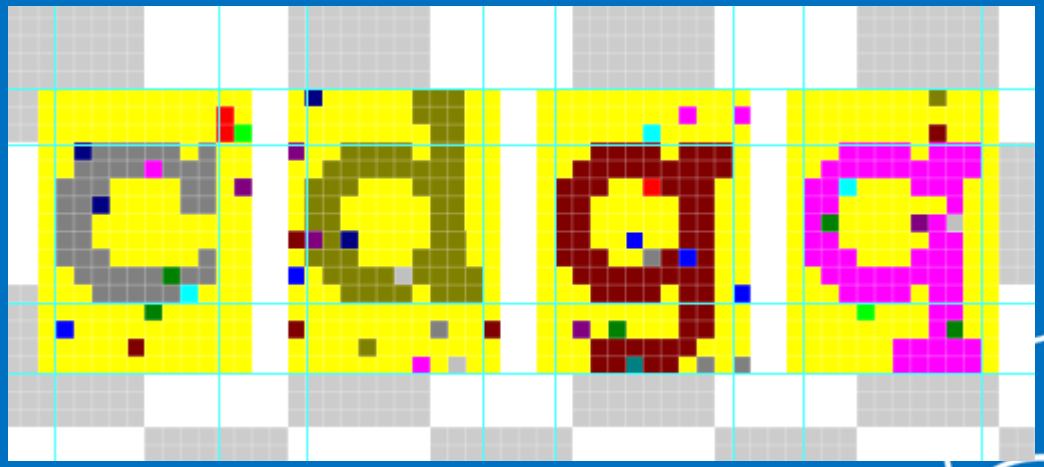


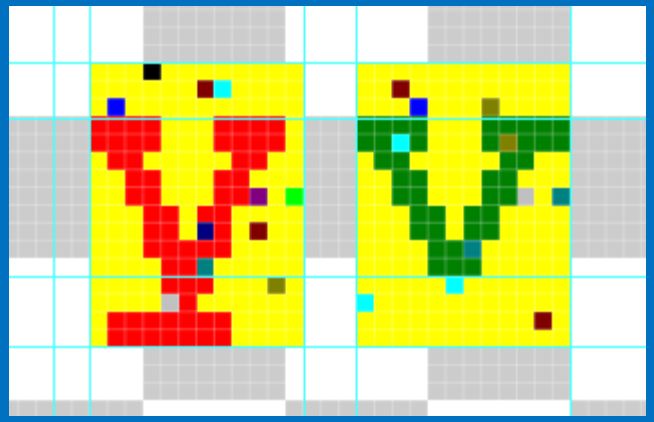




处理方法
--识别训练

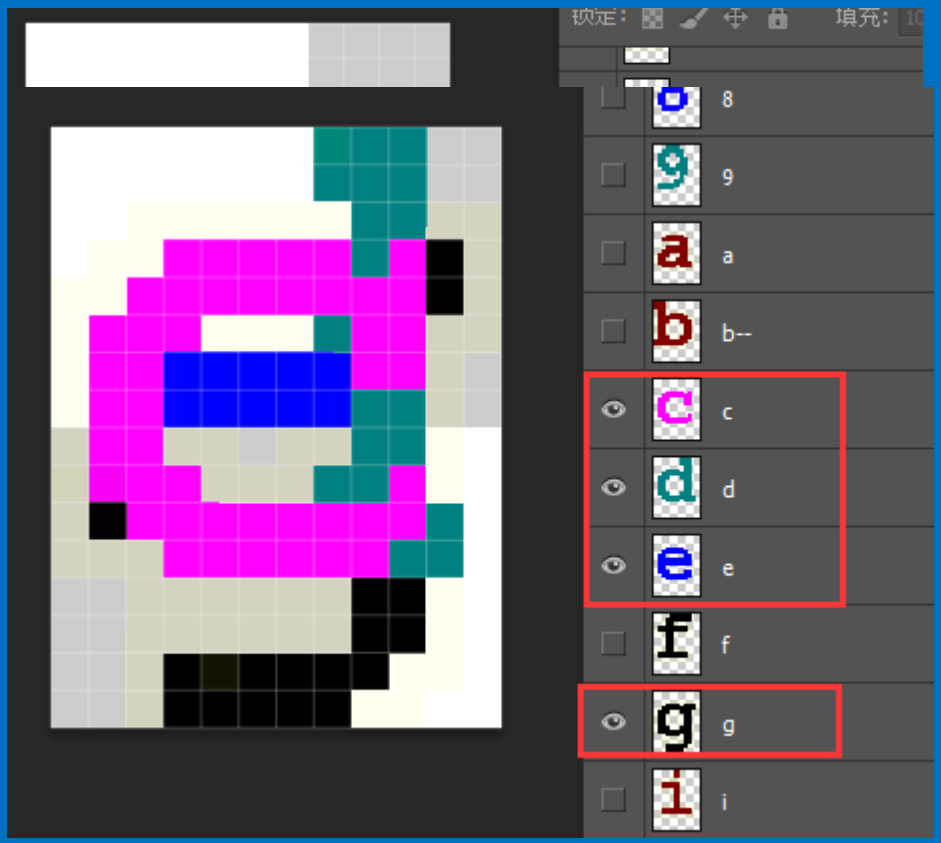
对相似的字符进行区分





处理方法
--识别训练

对相似的字符进行区分



处理方法
--识别训练

对相似的字符进行区分

```
78                  if("c".equals(next)){
79                  if((tmp>0.97)){
80                      //处理c d g  的d
81                      {
82                        if(next.equals("c")){
83                                  int m=0;
84                                 for(int k=0;k<3;k++){
85                                     for(int l =7;l<10;l++){
86                                        if(start[k][l]==1)m++;
87                                        }
88                                     }
89                      if(m>5)return "d";
90                      }
```

处理方法
--识别训练

对相似的字符进行区分

结果c d g 完全分开
YV 完全分开

处理方法
--识别训练

图片分析

对相似的字符进行区分

二次处理

处理方法
--识别训练

对相似的字符进行区分

```
188    public static String ImageOCRTwoErCi(BufferedImage img) throws IOException{
189     int start[][]=imageTest.getErZhiHua(img);
190        Set<String>Zi=ziku.ZIs.keySet();
191        Iterator<String> iter= Zi.iterator();
192        while (iter.hasNext()) {
193            String next = iter.next();
194           System.out.println("-----------"+next);
195           // ziku.Print_zi(ziku.ZIs.get(next));
196
197            int count=0;
198            int src[][]=ziku.ZIs.get(next);
199            for(int i=0;i<16;i++){
200            for(int j=0;j<12;j++){
201                if(1==src[i][j]&&1==start[i][j]){
202                count++;
203                }
204            }
205            }
```

处理方法
--识别训练

图片分析

对相似的字符进行区分

C和e识别有误

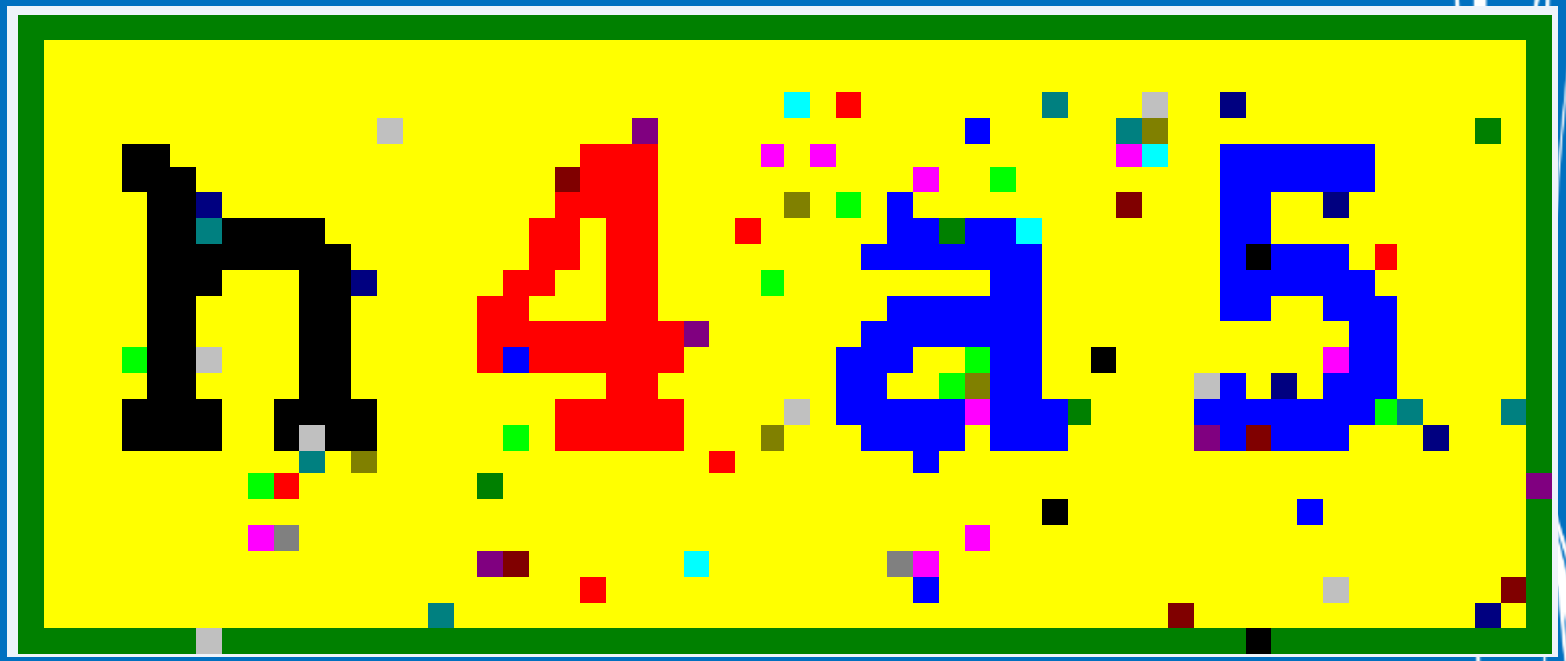
处理方法
--识别训练

对相似的字符进行区分

42207张验证码我们团队15级的看了两个多小时没发现两处错误

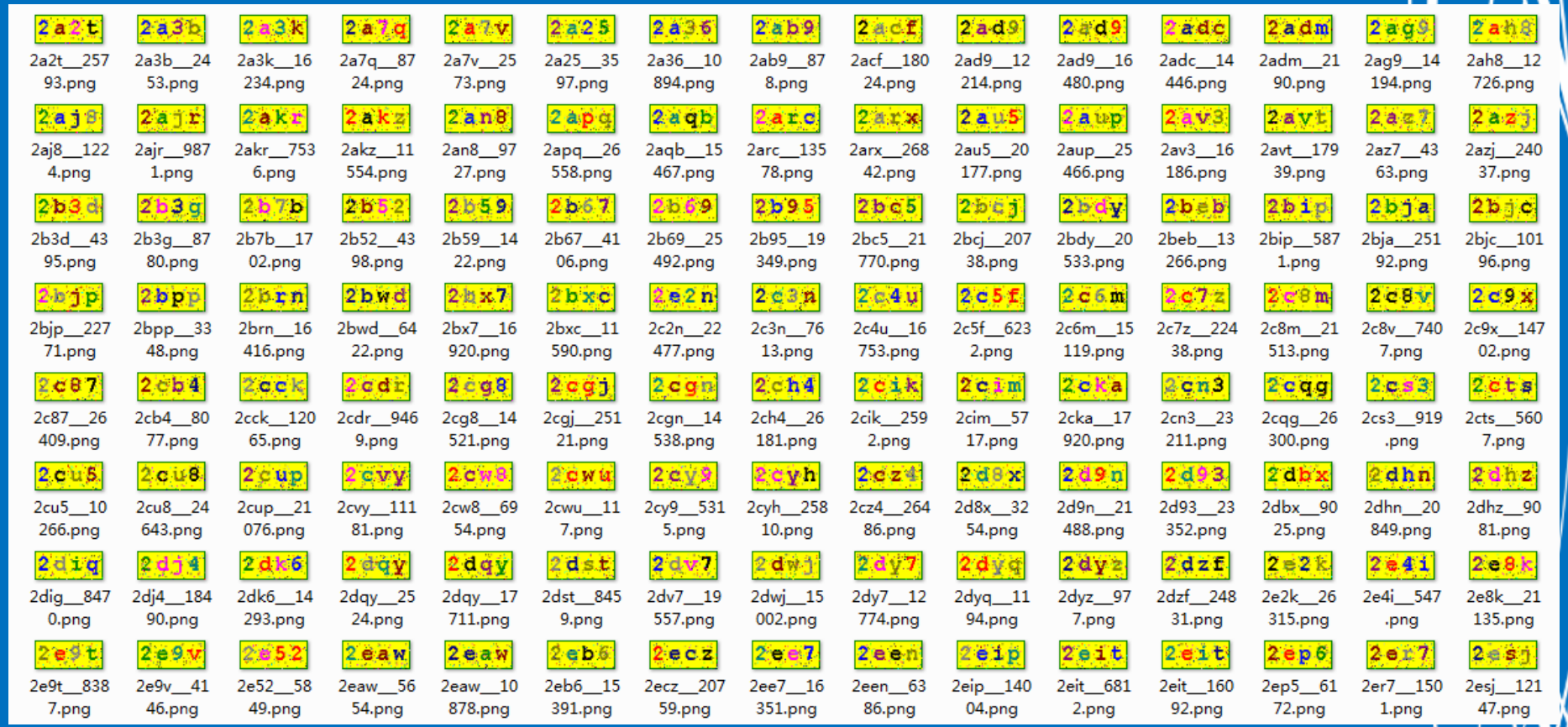
处理方法
--识别训练

对相似的字符进行区分



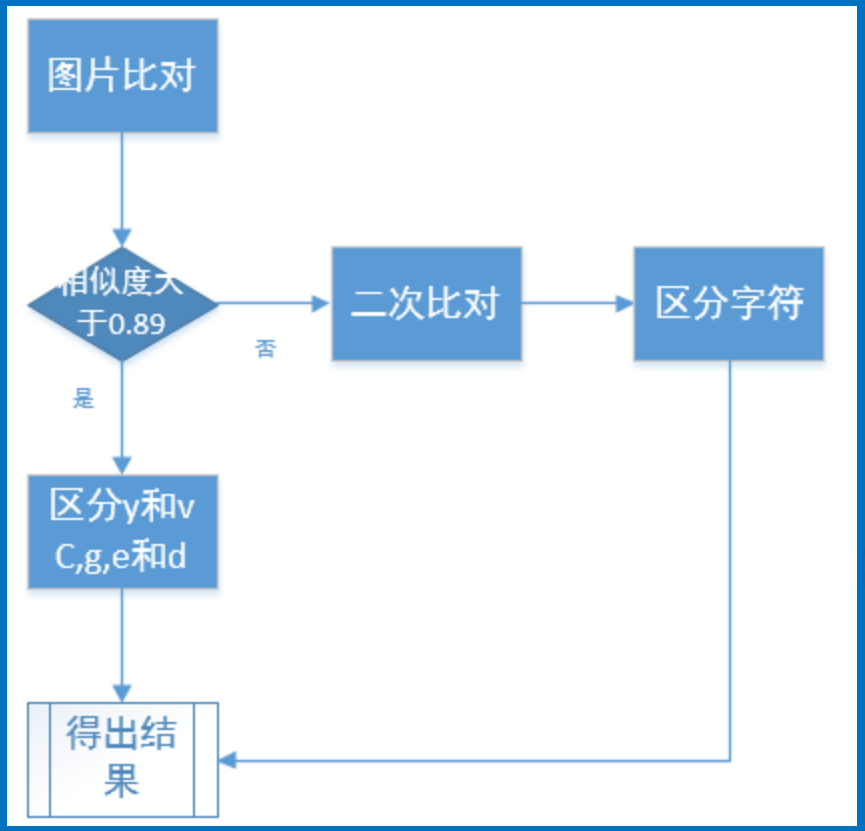
处理方法
--识别训练

对相似的字符进行区分



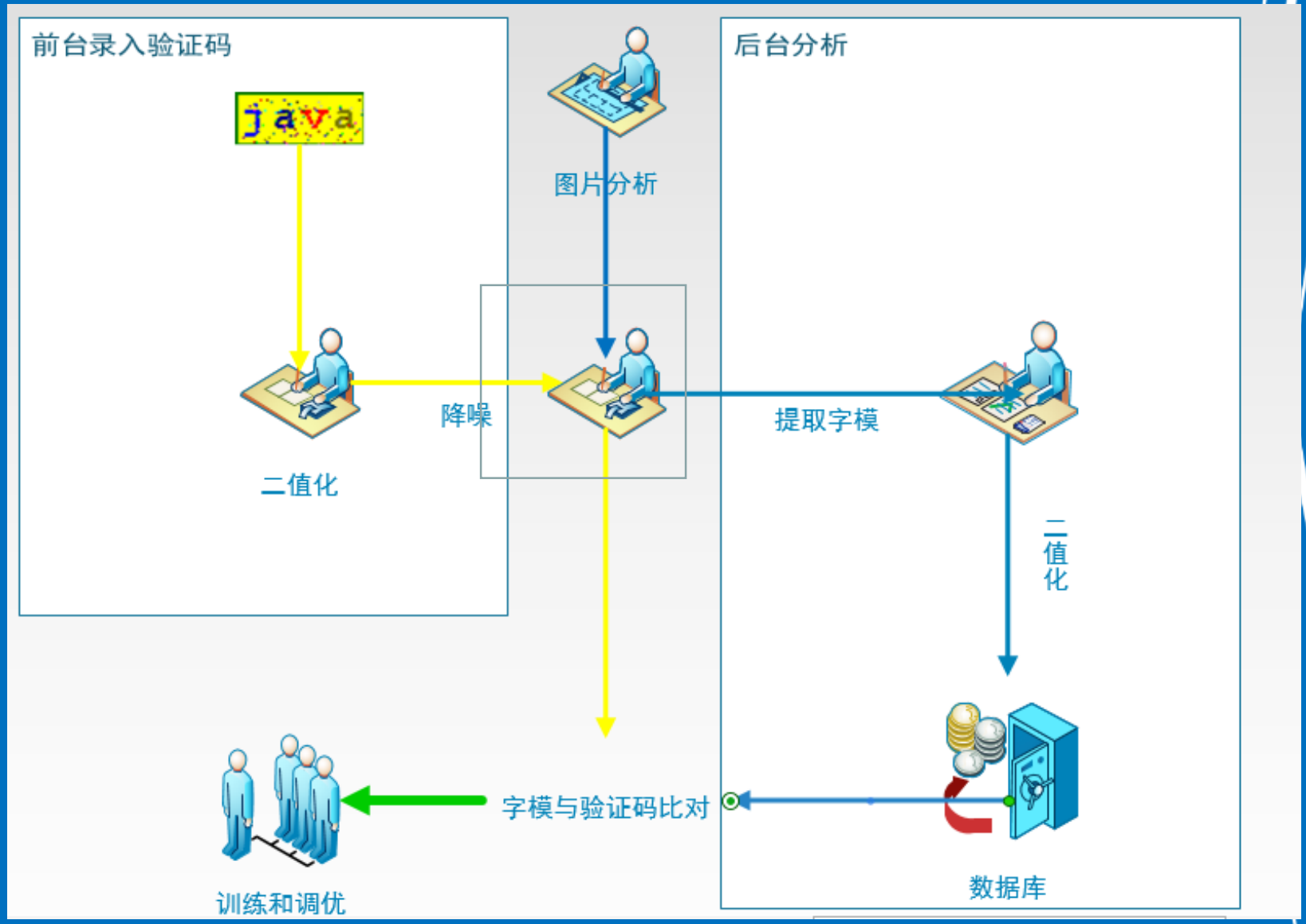
处理方法
--识别训练

对相似的字符进行区分



总结

--处理流程



总结

--所用技术

图片处理

--photoshop
--java imageIO类

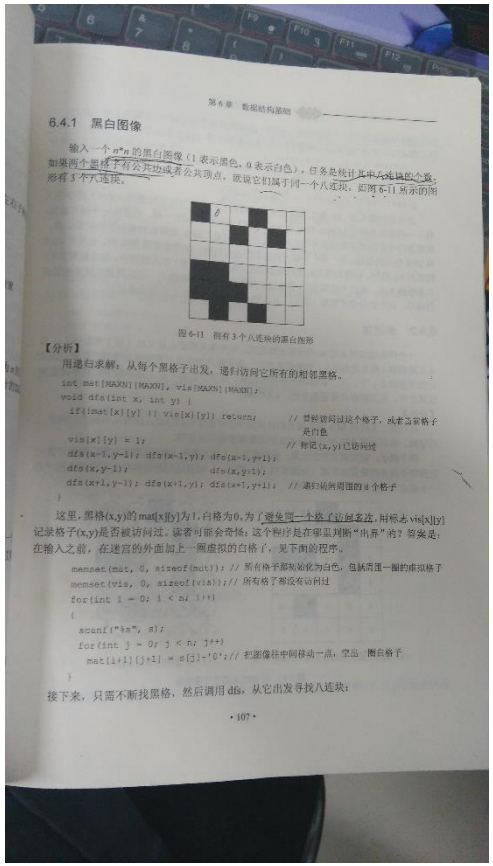
正则匹配

---editplus

1.图片降噪的算法

数据结构中八连块问题

代码优化



第6章 数据结构基础

6.4.1 黑白图像

输入一个 $n*n$ 的黑白图像（1 表示黑色，0 表示白色），任务是统计其中八连块的个数。如果两个黑格子有公共边或有公共顶点，就说它们属于同一个八连块。如图 6-11 所示的图形有 3 个八连块。

图 6-11 有 3 个八连块的黑白图形

【分析】

用通归求解，从每个黑格子出发，递归访问它所有的相邻黑格。

```
int mat[MAXN][MAXN], vis[MAXN][MAXN];
void dfs(int x, int y) {
 if(!mat[x][y] || vis[x][y]) return;          // 曾经访问过这个格子，或者当前格子是白色
 vis[x][y] = 1;                               // 标记(x,y)已访问过
 dfs(x-1,y-1); dfs(x-1,y); dfs(x-1,y+1);
 dfs(x,y-1);               dfs(x,y+1);
 dfs(x+1,y-1); dfs(x+1,y); dfs(x+1,y+1);      // 递归访问周围的 8 个格子
}
```

这里，黑格(x,y)的 mat[x][y]为 1，白格为 0，为了避免同一个格子访问多次，用标志 vis[x][y]记录格子(x,y)是否被访问过。读者可能会奇怪：这个程序是在哪里判断“出界”的？答案是：在输入之前，在迷宫的外面加上一圈虚拟的白格子，见下面的程序。

```
memset(mat, 0, sizeof(mat)); // 所有格子都初始化为白色，包括周围一圈的虚拟格子
memset(vis, 0, sizeof(vis));// 所有格子都没有访问过
for(int i = 0; i < n; i++)
{
 scanf("%s", s);
 for(int j = 0; j < n; j++)
  mat[i+1][j+1] = s[j]-'0';// 把图像往中间移动一点，空出一圈白格子
}
```

接下来，只需不断找黑格，然后调用 dfs，从它出发寻找八连块：

· 107 ·

算法竞赛入门经典107页八连块问题—深度优先搜索

2.字模矩阵压缩

压缩矩阵

----参考数据结构严蔚敏版

代码优化

3.**********

还有更好的优化方法，更好的验证码识别算法

以下摘自网络

分类

纯文本型验证码

图型验证码

摘自网络

扩展

纯文本型验证码

- 1+1=?
- 本论坛的域名是?
- 今天是星期几?
- 复杂点的数学运算

摘自网络

扩展

图形验证码

识别图形验证码可以说是计算机科学里的一项重要课题，涉及到计算机图形学，机器学习，机器视觉，人工智能等等高深领域.....

简单地说，计算机图形学的主要研究内容就是研究如何在计算机中表示图形、以及利用计算机进行图形的计算、处理和显示的相关原理与算法。图形通常由点、线、面、体等几何元素和灰度、色彩、线型、线宽等非几何属性组成。计算机涉及到的几何图形处理一般有 2维到n维图形处理，边界区分，面积计算，体积计算，扭曲变形校正。对于颜色则有色彩空间的计算与转换，图形上色，阴影，色差处理等等

摘自网络

扩展

在识别验证码中需要用到的知识一般是 像素，线，面等基本 2 维图形元素的处理和色差分析。常见工具为：

- 支持向量机(SVM)
- OpenCV
- 图像处理软件(Photoshop,Gimp…)
- Python Image Library

扩展

后续知识请自行搜索

经验

1.勤用搜索引擎（学好了可以装逼）

2.多和别人交流或者听听别人（大师）的解决问题的方法

3.请尽快把想法付诸实践

!

Thinker&Walker

More than the struggle for the dream