

项目名称是 testsvm

项目结构如下：

```
____testsvm
|____*.cpp
|____*.hpp
|____*.h
|____x64
|____testSet
|    |____data2
|        |____*.jpg
|        |____A4_correction    //经过 a4 纸矫正的结果
|        |____threshold       //经过二值化后的结果
|        |____row_div         //经过行划分后的结果
|        |____col_div         //经过列划分后的结果
|        |____svmmnist.xml    //svm 训练后的保存文件
|        |____data.csv        //最终输出的识别的学号手机号和身份证号
```

实验名称：手写数字检测-part2

实验要求：通过标准流程实现对手写数字的检测，输出相关数据。

实验步骤：

第一步 A4 纸矫正。结果放在 testSet/A4\_correction 目录下。

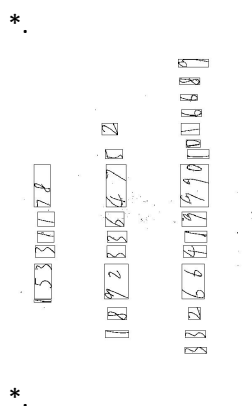
第二步二值化。结果放在 testSet/threshold 目录下。

第三步行划分。结果放在 testSet/row\_div 目录下。

第四步列划分。结果放在 testSet/col\_div 目录下。

第五步对划分出来的图像进行数字识别。结果放在 testSet/data.csv 文件里。

和 part1 相比，多了更多种类的恶性数据，像这种：



1 4 2

1 2 3 4

1 2 3 4 5 6 7 8 9 10 11 12

1 2 3 4 5 6 7 8 9 10 11 12

\*

1 2 3

1 2 3 4 5 6 7 8

1 2 3 4 5 6 7 8 9 10

1 2 | 3 4 5 | 6 7 8 9 10 11 | 12

\*

1	6	3	4	0	2	3	4				
1	3	2	8	6	8	7	7	0	9		
4	4	1	4	2	4	1	9	9	8	11	12
										6	9
										5	8

\*

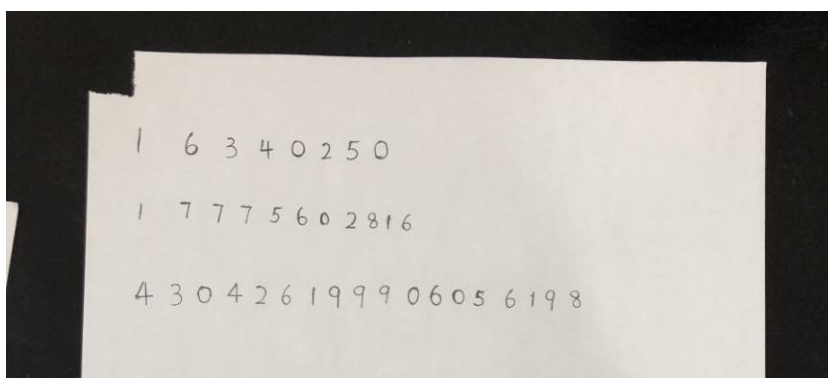
姓名： 田和宇

学号： 1 6 3 4 1 0 2 4

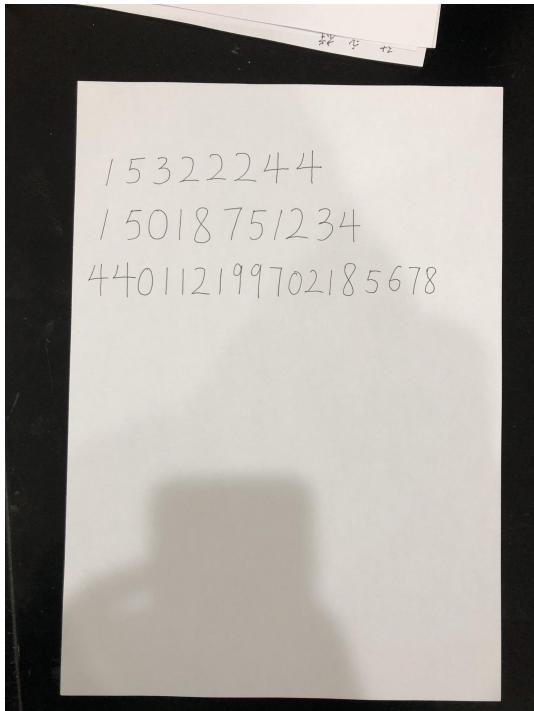
手机号： 1 5 6 2 6 2 6 4 0 2 4

身份证号： 4 1 0 8 8 2 1 9 9 8 0 8 0 8 8 0 5 3

\*



除了这么些比较个别的，其他的都能正常划分出来字符。但是识别上仍然很低。  
还有一点就是有的图片上有超过一张 A4 纸，即有另外 A4 纸的边角。



这样的图片通过影响霍夫转换，间接影响了 A4 纸矫正，对此一个处理函数，将相连的白色像素点打同样的 **tag**，筛选遗留最大面积的白块。

还有就是阴影问题，关于分区域进行分割的想法，感觉并不是很好实施，当窗口内一半是白纸，一半是阴影时，阴影部分就会出现被二值化为 **0**，合并后的黑块非常影响对数字的识别。

对此查了博客，有博客讲解了一种通过计算背景图的方法来解决光照对图片二值化的影响。博客地址忘了。

对图片进行遍历，找到每个像素点周围 (**5\*5**) 最大的 **6** 个像素点，取第 **2-6** 个像素点的平均值作为该点像素的背景颜色。

对图片进行遍历，如果改点像素值与背景像素值之差的绝对值在一个允许的误差范围内，则认为改点是背景点，置像素值为 **0**（黑色背景）。否则认为该点属于前景的点，置像素值为 **255**；

程序实现消除光照影响函数具有极大局限性，一个是速度慢，另一个是如果字体太粗，字体中间会被置为背景色。