

目录

数字字符识别——模板匹配.....	1
1 问题描述.....	1
2 解决方案分析.....	1
2.1 整体思路.....	1
2.2 模式获取.....	1
2.3 测试图片二值化处理.....	1
2.4 数字匹配识别.....	2
2.5 判定识别结果.....	2
2.6 划痕与噪声图片的处理.....	3
2.7 标记已经处理过的区域.....	5
3 实验结果与分析.....	6
3.1 实验结果.....	6
3.2 结果分析.....	6
4 补充题目.....	6

数字字符识别——模板匹配

1 问题描述

图像中有一串由 0-9 组成的数字序列，请设计一个程序，能够自动识别出图中的数字列，并给出字符的所在位置。

提供数据：本次作业提供两组数据，分别存放在 `train` 文件夹和 `test` 文件夹中。请利用给出的图像模板对测试图像进行数字字符识别。

2 解决方案分析

2.1 整体思路

采用扫描窗的方式对测试图像进行扫描，对每个扫描窗进行模板匹配，若最接近某个模板，就认为是模板对应的数字。采用的匹配特征是图像的像素分布，即将图像转为向量，利用余弦相似度来衡量匹配程度。为了提高模板匹配的正确率，先对图像进行二值化处理。

整体流程如下：

- 1) 读取训练数据，统一尺寸（为程序处理方便），二值化后得到模板。
- 2) 对测试图片二值化处理。
- 3) 采用扫描窗的方式对测试图像进行扫描，对每个扫描窗进行模板匹配。用余弦相似度衡量匹配程度。
- 4) 设置相应阈值，将满足要求的扫描窗提取出来。
- 5) 结合先验知识，对匹配的结果进行分析验证。

下面分别分析各个流程。

2.2 模式获取

“train/”目录下给出了各个数字的模板，对这些图片直接采用 `im2bw` 函数将其二值化即可。因为图像较小，图像亮度响度均匀，不用再局部自适应二值化。

由于各个模板的大小不相同，这给后续的程序处理带来一定麻烦，可以的解决方案是将所有模板统一到相同的尺寸（最大的模板大小）。处理过程中，我遇到了这样一个问题：

Q: 统一到最大模板大小时，应该在图片四周补充空白，还是按照最大模板的大小，对各个模板进行 `resize`（通过插值实现）？

A: 实验中，我对两个方法均进行了测试，发现将模板 `resize` 的效果一般。因为这么处理的话，实际上相当于改变了模板本身的形状（长宽比），这很容易对后续的匹配造成不利影响。在图片四周补充空白的效果比较好，但需要注意的是，因为模板周围出现了空白区，为了能成功匹配测试图像中边缘区的数字，应当同样用空白对测试图像四周进行延拓。

按照上述方法，提取的数字模板如图 1 所示。

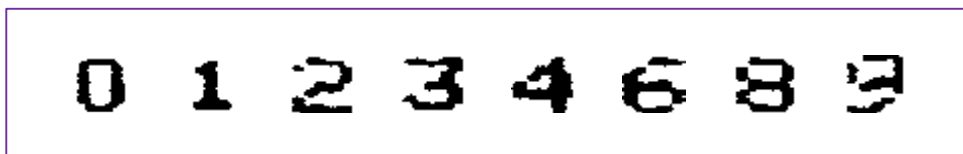


图 1 提取出的数字模板

2.3 测试图片二值化处理

与数字模板的二值化处理不同，测试图片的不同区域平均亮度不同，直接整幅图像二值化后效果很不理想（如图 2 所示）。因此，考虑采用滑窗的方式，对图像的局部进行自适应

二值化。

处理过程中，遇到了一个这样的问题：

Q: 局部处理时，采取的局部块多大合适？

A: 我最初采用了很小的滑窗进行处理，心想着这样的效果必定是极好的，但得到的结果却比较糟糕（如图 3 所示）。分析后发现，滑窗过小的话，可能滑窗内根本没有数字，这样很可能将背景色全部二值化为黑色。考虑到图像比较“扁长”，采取与图像高度相等的滑窗效果会更好一些。

最终二值化的效果如图 4 所示。

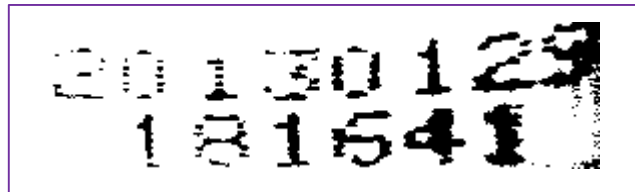


图 2 直接二值化

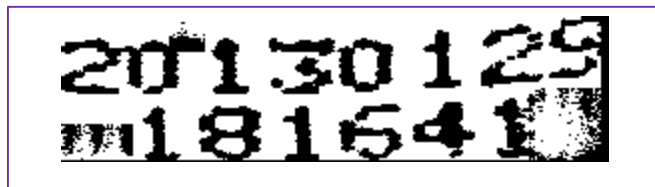


图 3 采用过小的滑窗二值化

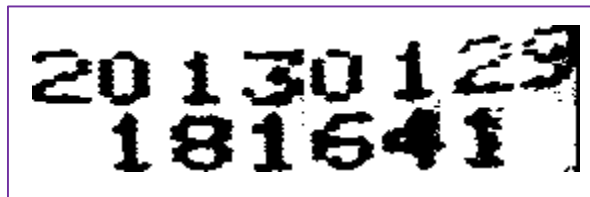


图 4 采用与测试图像等高的滑窗二值化

2.4 数字匹配识别

对于每个扫描窗，依次采用每个模板与扫描窗进行匹配测试。衡量匹配度的指标是余弦相似度，即将两幅图像均看成向量，这样一幅二值图像恰好对应一个布尔向量。余弦相似度的计算公式如(1)所示。

$$similarity = \frac{\vec{v}_1 \cdot \vec{v}_2}{|\vec{v}_1| |\vec{v}_2|} \quad (1)$$

注意到需要拒识非数字字符的窗口，因此在选取最大 similarity 的同时，需要比较最大与次大的差值，如果差值过小，很可能说明此处不是任何数字。因此，判定是否匹配至少需要为这两个参数设定阈值：

- 1) 最大匹配值
- 2) 最大匹配值与次大匹配值的差

对于每个成功匹配的扫描窗，程序中保存了其匹配的模板类型（即哪个数字）和扫描窗的位置（以扫描窗的中心点位置标记）。

2.5 判定识别结果

测试样例中，第一行的“20130129”和第二行的“181441”在图中的位置基本是固定的，因此，我利用图片编辑工具对图中的每个数字进行了定位。评估识别结果时，结合预定义的

数字区域范围和匹配的结果，对识别结果进行判定。

实验中，对各个数字区域范围的标记如表 1 所示。

数字		左上角点坐标 (x,y)		右下角点坐标 (x,y)	
		x	y	x	y
第一行	2	10	1	36	25
	0	14	34	39	56
	1	9	71	39	96
	3	11	103	38	134
	0	12	139	36	164
	1	7	178	35	202
	2	2	208	31	240
	9	1	243	32	274
第二行	1	46	38	72	60
	8	45	67	72	100
	1	45	107	72	131
	6	43	137	72	169
	4	42	175	72	208
	1	40	212	72	237

表 1 对图中各个数字区域范围的标记

2.6 划痕与噪声图片的处理

噪声图片

从原图可以看出，图中含有比较明显的椒盐噪声，而中值滤波是处理椒盐噪声很有效的方式，因此考虑先对含噪图像中值滤波，然后再进行上述的二值化、空白延拓、模板匹配、数字识别等操作。

如果不滤波，直接二值化的结果如图 5 所示；滤波后再二值化的结果如图 6 所示。

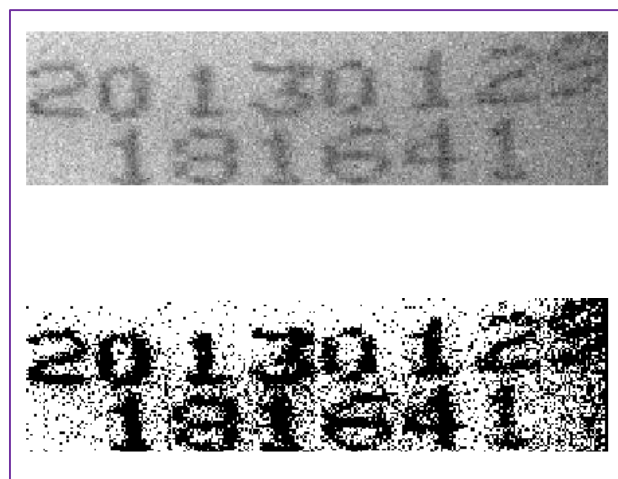


图 5 无滤波二值化结果

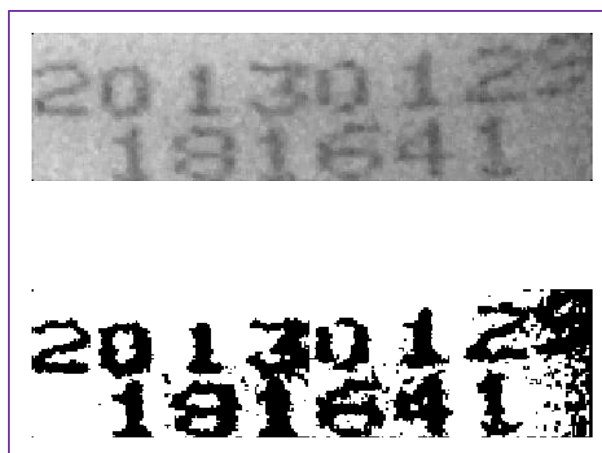


图 6 中值滤波后二值化结果

划痕图片

实际上，划痕对模板匹配没有很大的影响。由于划痕的存在，数字模板与扫描窗的匹配度肯定稍有降低，但如果匹配到正确数字时，相似度也会有一个峰值。因此，在识别过程中，适当降低匹配要求的阈值，仍可以达到比较好的结果。

此外，我发现如果同样先对划痕图片进行中值滤波，再匹配的效果会更好一些。想必划痕也是一种“噪声”，不过这种差别通过图像来看并不太明显。图 7 是没有中值滤波的情况，图 8 是中值滤波后的情况。

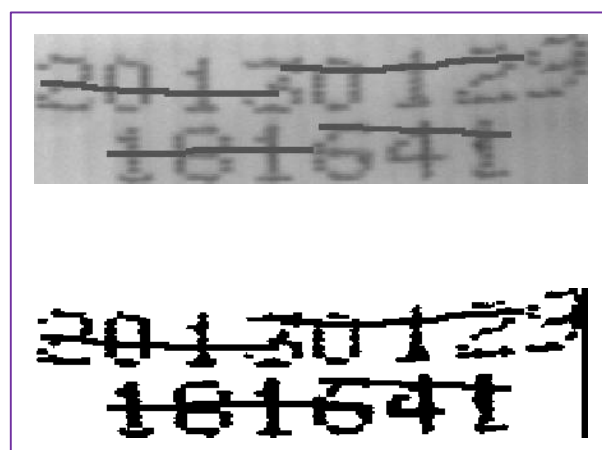


图 7 无中值滤波

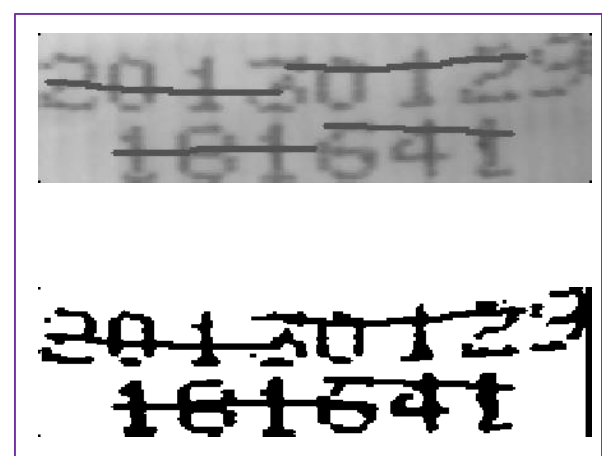


图 8 中值滤波处理结果

2.7 标记已经处理过的区域

如果直接对整幅图片逐像素滑窗，有可能会遇到这样一个问题：

Q: 如果某个扫描窗与某个数字模板成功匹配，那么其临近的扫描窗也往往能够匹配成功，但这些区域实际上只有一个数字。如果对此不加处理的话，有可能误识别出多余的数字，如图 9 所示。

A: 我想了两个方法解决该问题。

- 1) 判定全部匹配成功的扫描窗的重叠程度，可以设定一个阈值，重叠程度高于该阈值的即认为是同一个数字。此时，也可以通过这些重叠区的识别结果进行相互验证，看看它们是否匹配的同一数字。
- 2) 在识别过程中构造一个与测试图像等大小的 mask 模板（全 0），如果某一扫描窗成功匹配，则将其在 mask 里相对应的部分全部置 1。当再次匹配时，如果发现该区域对应的 mask 已经有较多 1（通过阈值限制），则认为是同一个识别区域，放弃对该扫描窗的匹配。

显然，从复杂度考虑，方法(2)更为简便一些。因此实验中采用该方法来标记已经处理过的区域。处理后的结果如图 10 所示。

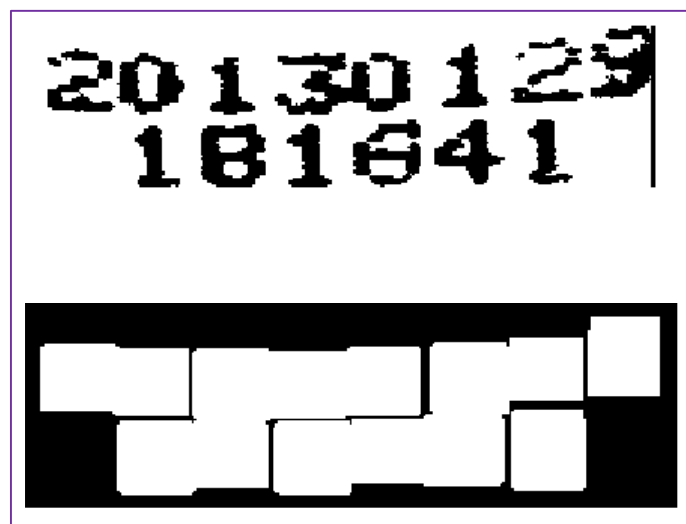


图 9 匹配结果（可以看出，同一数字附近有多个扫描窗成功匹配）

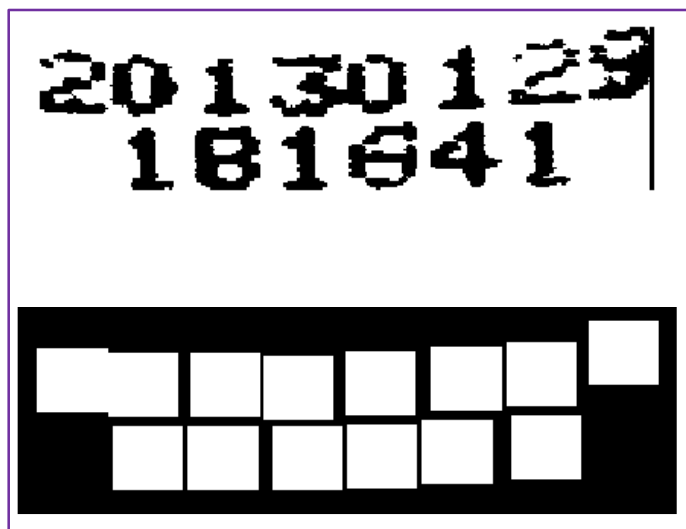


图 10 对已识别区域添加标记后的识别结果

3 实验结果与分析

3.1 实验结果

经过对参数阈值的调整、测试，目前得到的比较好的匹配结果如表 2 所示，识别结果中的“□”表示该位置未识别出数字。

测试图片	识别结果	识别错误个数	备注
1.bmp	201301□9 181641	1	无滤波 使用相同的阈值参数
2.bmp	20130124 181641	1	
3.bmp	20130124 181641	1	
4.bmp	20130129 181641	0	
5.bmp	201301□9 181641	1	
6.bmp	201301□9 181641	1	
划痕.bmp	20□30129 181641	1	中值滤波 匹配参数阈值最小
噪声.bmp	20130124 181641	1	中值滤波 匹配参数阈值次小

表 2 识别结果

3.2 结果分析

表 2 中的结果是多次对阈值参数进行调试后得到的一个比较好的结果。通过对比不同的操作流程和参数设置，我对结果的分析如下：

- 1) 前 6 幅测试图像的阈值参数基本一致，也不用中值滤波。因为这些图片相对来说没有噪声，不必要的滤波反而可能损害图片的特征（实验中，滤波后的效果确实不升反降）。
- 2) 对于带椒盐噪声的图片，中值滤波后的效果明显好于没有滤波的效果。尽管滤波对带匹配数字本身有一定损害，但这种损害远小于消去噪声的影响而带来的效果的提升。当然，由于图像本身受噪声、滤波的影响，匹配阈值较前 6 幅图应有所降低。实验中，匹配阈值参数从 0.73 调到了 0.72。
- 3) 对于带划痕的图片，首要需要处理的是对阈值的调整。因为划痕的存在，匹配时的最大匹配系数会进一步降低，但匹配的方法不用调整，因为正确数字的匹配程度肯定依然高于不匹配的数字。实验中，匹配阈值参数进一步降低到了 0.7。此外，对图像先滤波，也会带来一定程度上的效果提升。
- 4) 在对含噪声、划痕的图片进行滤波时，我采用的是中值滤波，而不是均值滤波。一方面，根据数字图像处理的知识，消去椒盐噪声的最好滤波方式是中值滤波；另一方面，均值滤波会进一步模糊真实数字的边缘，会给模板的匹配带来更不利的影响。

4 补充题目

尺度与模板不一致

解决思路：

假如我们已经有了与模板匹配的图像大小这个先验信息，可以直接将测试图像放缩至适配大小（Matlab 中的 `imresize` 函数）。此后的操作流程就与之前的完全一样了。

实验中，我测试了这种方法，设置的阈值参数是

- 1) $\max_match \geq 0.72$
- 2) $\max_match - \text{second_max_match} \geq 0.05$

得到的识别结果如所示。

测试图片	识别结果	识别错误个数	备注
补充 1.bmp	20130129 181641	0	无滤波
补充 2.bmp	20130129 181641	0	

表 3 补充图像的识别结果

但在实际的模式识别问题中，需要将测试样本（或是模板）放缩多少，往往是无法准确预知的。针对此，我有如下两个思路：（时间原因，本实验没有再对这两种思路进行实现）

- 1) 先根据图像的大小，选择一个初始值作为模板的放缩大小，然后用滑窗匹配，随后逐步调整模板的大小，直至能够识别出较为满意的效果。
- 2) 更换提取的特征。本实验中，我是把二值化后的图像看成一个向量，利用余弦相似度衡量两个向量的相近程度。如果更换特征，比如可以考虑将图像灰度值的累积分布函数作为待匹配的特征，由于灰度上的频率分布与图像尺寸大小并无关系，因此可以较好地解决尺度与模板不一致的问题。