模式识别作业 1

基于模板匹配的手写数字识别

姓名：罗雁天

院系：清华大学电子系

学号：2018310742

日期：March 25, 2019

目录

第 1 章 问题描述

本次作业使用模板匹配的方法来对手写数字字符进行识别。提供两组数据，分别存放在 train 文件夹和 test 文件夹中.

- train 文件夹中存放已单独分割出来的 0-9 数字图像模板，模板已统一到相同的尺度，每个模板为对应的数字外切分割结果；
- test 文件夹中有 8 张用于测试的图像，其中 6 张正常尺度 (与模板尺度相同)，一张存在划痕，一张有噪声。

本次作业便利用此给出的图像模板对测试图像进行数字字符识别。

1.1 模板图像预处理

为了获得更好的识别效果和识别的鲁棒性，我们对训练集中的 8 张图片进行预处理，这里主要采用的算法便是将其二值化。我们将二值化后的图像保存到单独的文件夹中，方便我们之后进行模板匹配的操作。图1.1显示了训练集模板图片预处理前后的对比图，上面部分为训练集的原始灰度图片，下面图片为预处理之后的二值化图片。

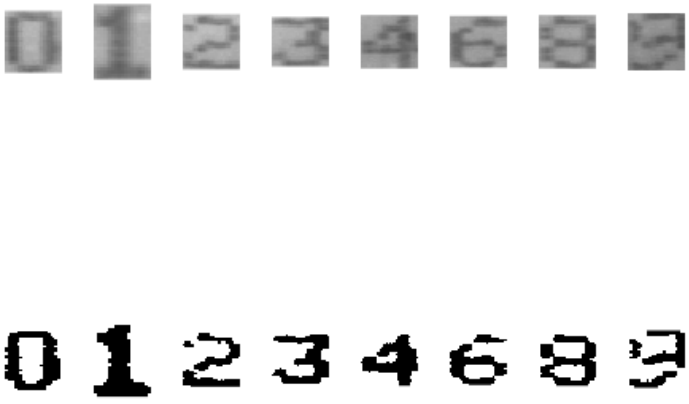

图 1.1: 训练集数据预处理前后对比图

1.2 测试集噪声图片预处理

对于含有噪声的图片，如果我们直接对其进行二值化，可以看出噪声很多，直接识别的话效果不佳，如图1.2所示。因此我们首先对其进行维纳滤波，如图1.2所示，可以看出维纳滤波可以较好的滤除了噪声。但是如果维纳滤波之后的图片直接进行二值化，可以看出左侧数字部分相比于直接二值化有了较大提升，但是右侧较暗的区域依然不足以识别。因此我们需要对其进行背景均衡化处理，具体算法为：

- 首先对图像进行模糊滤波得到背景图；
- 定义全局亮度为整张图片的亮度平均值；
- 计算 $I_{balance} = I - background(I) + \bar{I}$

从图1.2最后的结果看出，使用此种算法处理之后的二值化图片具有较好的直观效果。

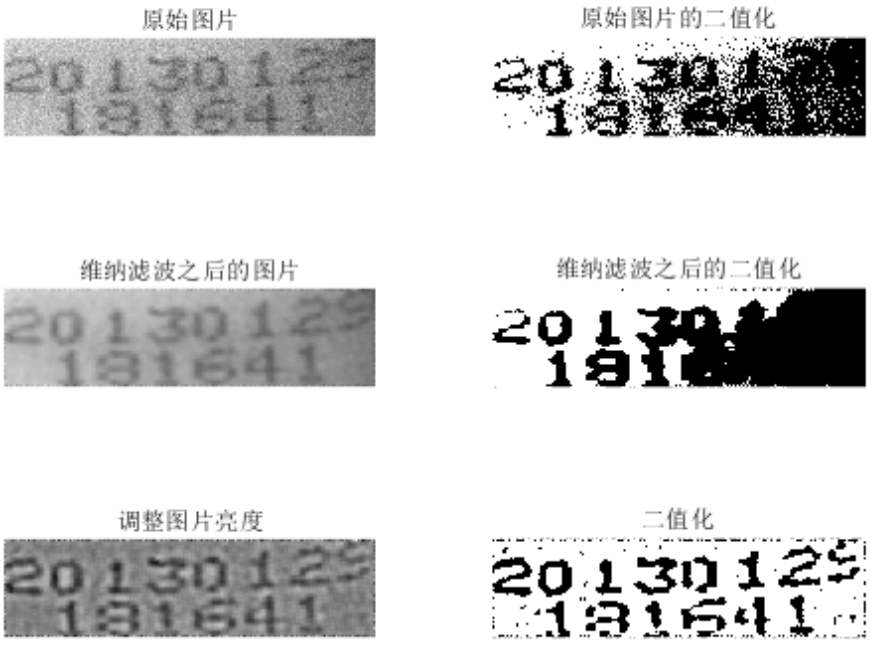

图 1.2: 噪声图片预处理对比图

1.3 测试集划痕图片预处理

如图1.3所示，划痕图片中划痕的灰度值明显较小，因此我们可以设置一个阈值 (在本实验中设置为 90)，将灰度值小于 90 的直接置为缺失值，得到中间图

所示的情况，可以看出划痕部分已经完全检测出来了。然后我们使用插值的算法填充缺失值即可得到下面图所示的结果，从中可以看出，使用此算法可以较好的去除划痕。

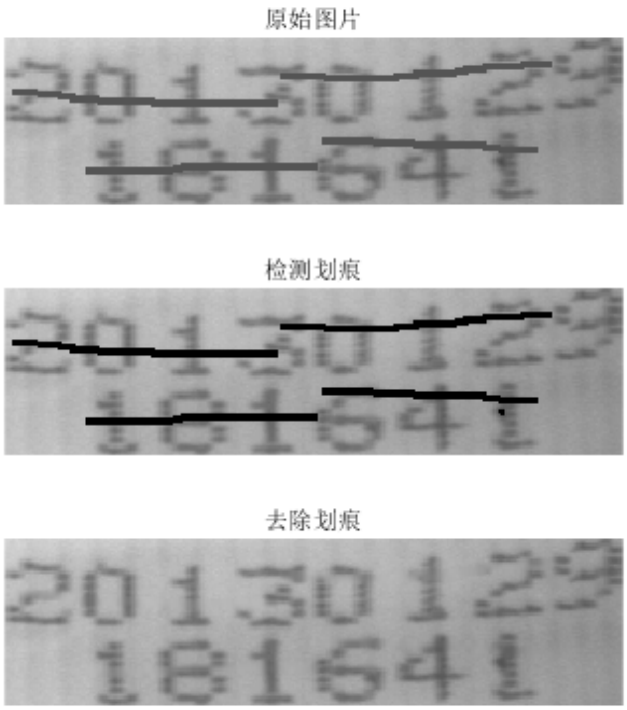

图 1.3: 划痕图片预处理对比图

第 2 章 简单模板匹配

可以用扫描窗的方式对测试图像进行扫描，每个扫描窗进行模板匹配，若最接近某个模板，就认为是该模板对应的数字。同时也需注意拒识非数字字符的窗口。同时观察测试图片发现可以对其进行分割，使得分割之后的每一块都包含一个数字字符，然后与模板进行匹配，认为匹配度高的便是该模板对应的数字。本章节对以上两种方法都进行尝试，并给出识别的结果。并且在 test 文件夹中还有两张进行过放缩处理的图片，对其我们便不能直接使用滑窗对其进行判别，因此我们对其使用多尺度扫描的方式进行检测。

2.1 基于分割的模板匹配

观察测试集的图片可以看出每一张测试图片都有多个数字组成，我们首先可以对齐进行分割，分割成小的图像块，然后对每一个图像块进行裁剪、二值化的预处理之后与预处理之后的训练集进行模板匹配操作，得到匹配值最大的数便认为是识别出来的字符。对于平均灰度值超过一定阈值 (在本实验中设置为 0.95) 的图像块，便认为该图像块没有数字。

2.1.1 正常图片识别结果

对 6 张正常图片进行识别，识别输出一个 2×8 的矩阵，矩阵每一个元素代表该图像块识别出来的数字字符，inf 代表该图像块没有数字。识别结果如图2.1所示。

2.1.2 噪声图片识别结果

首先对噪声图片进行第1.2节介绍的预处理操作，然后再对其进行模板匹配操作，得到如图2.2所示的识别结果。

2.1.3 划痕图片识别结果

首先对划痕图片进行第1.3节介绍的预处理操作，然后再对其进行模板匹配操作，得到如图2.3所示的识别结果。

../test/1.bmp的识别结果为:

2	0	1	3	0	1	2	3
Inf	1	8	1	6	4	4	Inf

../test/2.bmp的识别结果为:

2	0	1	3	0	1	2	3
Inf	1	0	1	6	4	1	Inf

../test/3.bmp的识别结果为:

2	0	1	3	0	1	2	9
Inf	1	8	1	6	4	1	Inf

../test/4.bmp的识别结果为:

2	0	1	3	0	1	2	3
Inf	1	0	1	6	4	1	Inf

../test/5.bmp的识别结果为:

2	0	1	3	0	1	2	4
Inf	1	0	1	6	4	4	Inf

../test/6.bmp的识别结果为:

2	0	1	3	8	1	2	9
Inf	1	8	1	6	4	4	Inf

图 2.1: 正常图片识别结果

噪声图片识别结果为:

2	0	1	3	0	1	2	9
Inf	1	8	1	6	4	1	Inf

图 2.2: 噪声图片识别结果

图 2.3: 划痕图片识别结果

2.1.4 结果总结

我们总结了此种方法下各个字符的召回率和虚警率如表2.1所示。从表中我们可以看出，使用此种算法对 8 和 9 的识别率较低，并且将 9 经常误判为 3，8 误判为 0。从原图中我们可以看出，9 的位置比较倾斜，并且连接有点模糊，因此将 9 误判为 3 是可以理解的，同样的道理，图片中 8 在二值化处理之后中间线较模糊，可能这也是导致 8 会误判为 0 的原因。值得注意的是，对于噪声图片和划痕图片的识别率都是 100%，这也可以说明我们的预处理算法效果是比较好的。

表 2.1: 基于分割模板匹配的数字字符识别结果统计

字符	召回率 TP	虚警个数
0	15/16	3
1	37/40	0
2	16/16	0
3	8/8	3
4	8/8	5
6	8/8	0
8	5/8	0
9	5/8	0

2.2 基于滑动窗的模板匹配

基于滑动窗的模板匹配，对于同尺度的图片而言，便是利用提供的模板图像对测试图像进行滑动相关，判断结果，最后对检测结果进行融合得到最终的结果。其流程图如图2.4所示。

2.2.1 正常图片识别结果

采用滑动窗的模板匹配对正常图片进行识别，结果如图2.5所示

2.2.2 噪声图片识别结果

首先对噪声图片进行第1.2节介绍的预处理操作，然后再对其进行滑窗模板匹配操作，得到如图2.6所示的识别结果。

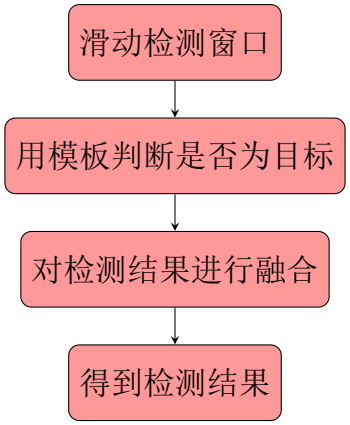

图 2.4: 基于滑动窗的同尺度模板匹配示意图

图 2.5: 采用滑动窗模板匹配的正常图片识别结果

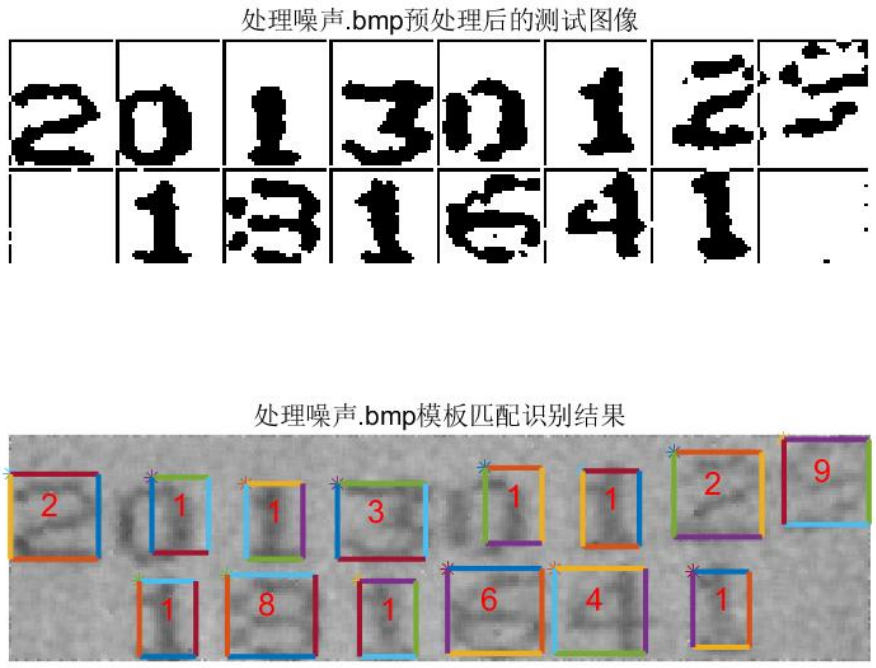

图 2.6: 噪声图片识别结果

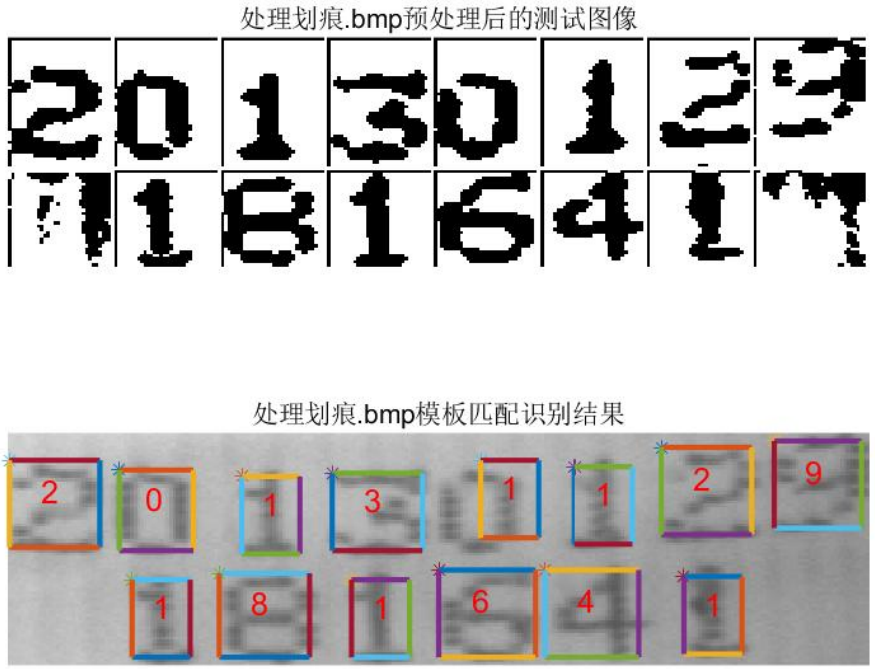

图 2.7: 划痕图片识别结果

2.2.3 划痕图片识别结果

首先对划痕图片进行第1.3节介绍的预处理操作，然后再对其进行滑窗模板匹配操作，得到如图2.7所示的识别结果。

2.2.4 结果总结

我们总结了此种方法下各个字符的召回率和虚警率如表2.2所示。从表中可以看出，基于滑窗的模板匹配主要出错的点是将 0 判别为 1 了，这是由于在滑动的过程中，模板先滑到了 0 的左边界，与 1 的相关系数较高，因此判别为 1 也不足为奇了。

表 2.2: 滑动窗匹配的数字字符识别结果统计

字符	召回率 TP	虚警个数
0	13/16	0
1	40/40	4
2	15/16	0
3	8/8	0
4	8/8	0
6	8/8	0
8	8/8	0
9	8/8	0

2.3 多尺度模板匹配

对于图像与模板不匹配的情况，我们不能直接使用图2.4所示的流程图进行模板匹配，我们需要对模板进行多尺度变换之后与测试图像进行匹配。算法流程图如图2.8所示。

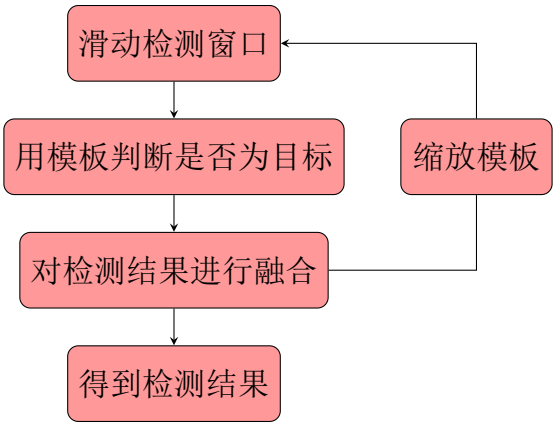

图 2.8: 基于滑动窗的多尺度模板匹配示意图

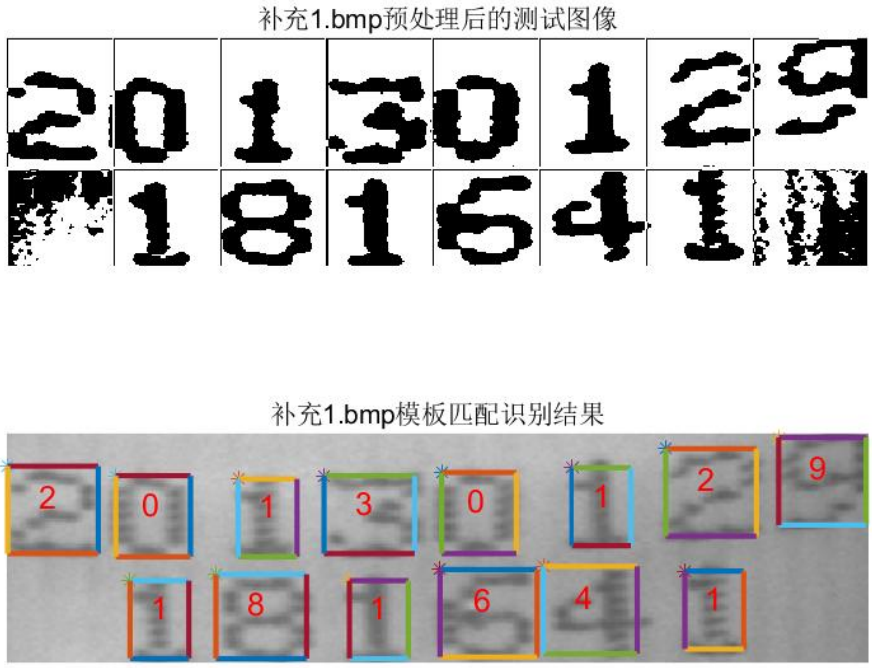

图 2.9: 放大图片识别结果

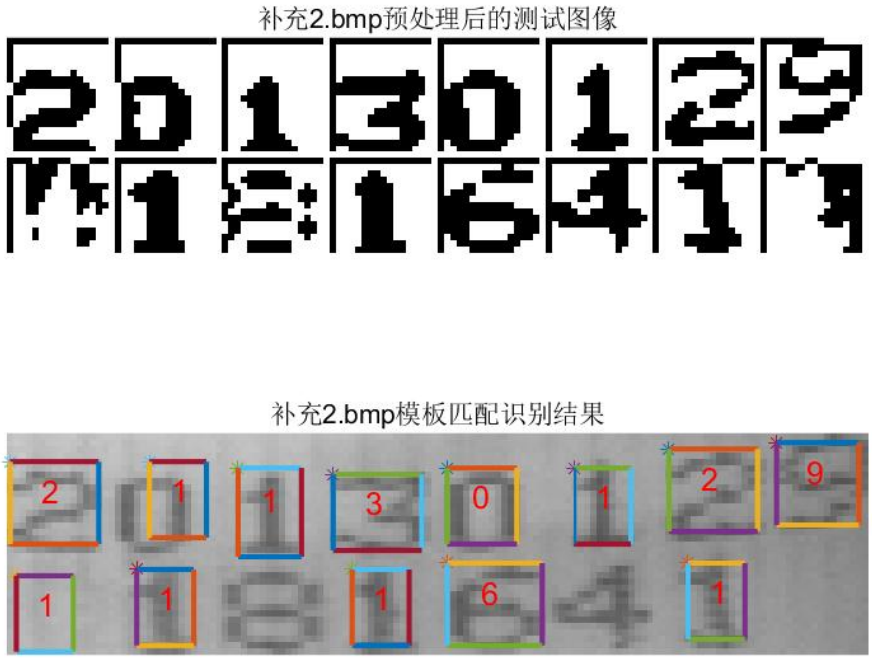

图 2.10: 缩小图片识别结果

分别对补充 1 和补充 2 的图片首先进行预处理 (去除光照影响、二值化等)，然后分别将模板图片进行多尺度缩放，之后再与预处理之后的测试图片进行滑动相关，取结果最大的数字字符作为识别结果。如图2.9和图2.10所示，分别为放大的图片和缩小图片的识别结果，从图中可以看出，对于尺度增大的图片识别率为100%，由此可以说明，使用多尺度的滑动窗模板匹配算法对放大的图片效果较好，但是对于尺度缩小的图片，有 1 个识别错误，2 个未识别出来，原因在于将模板缩小尺度的过程中图片会有失真，因此导致检测效果不好。

第 3 章 基于特征提取的模板匹配

在本章，我们将首先对图像分块提取特征，最后将特征合并在一起用于匹配，同时，我们依然使用了上一章的图像预处理的方法，为的是使我们的识别更加准确。

首先，我们采用作业中提示的方式对图像分块进行特征提取。如图3.1所示，将图像分块，每块逐行或者逐列统计黑像素点的个数作为局部特征，然后使用特征来进行模板匹配。对正常图片的识别结果如图3.2所示，对噪声图片 (使用了1.2的处理算法) 的识别结果如图3.3所示，对划痕图片 (使用了1.3的处理算法) 的的识别结果如图3.4所示。

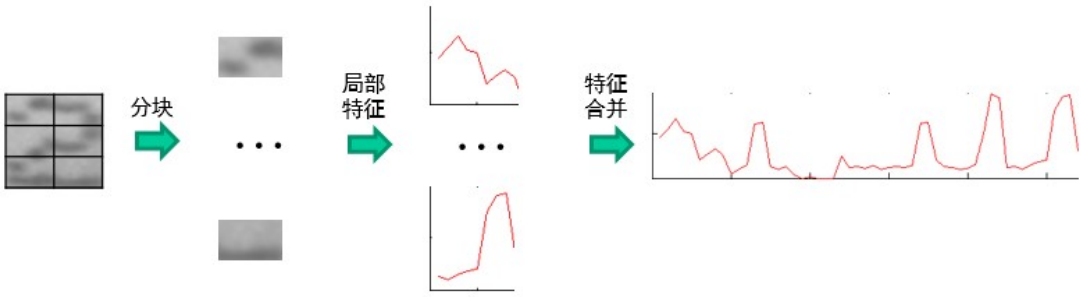

图 3.1: 图像分块计算特征

我们总结了此种方法下各个字符的召回率和虚警率如表3.1所示。从表中可以看出，基于特征提取的数字字符识别结果相比于直接的滑动窗模板匹配算法识别率更高，并且滑动窗算法中 0 被误识别为 1 的现象在特征提取的算法中也避免了，这是由于局部统计信息时 0 的边缘位置与 1 的特征相差较大，因此避免了这种现象。同时，从结果图中我们可以看出。噪声图片和划痕图片的识别率均达到了 100%，说明我们在第一章中提出的预处理算法是比较强的。

表 3.1: 基于特征提取的数字字符识别结果统计

字符	召回率 TP	虚警个数
0	16/16	0
1	40/40	0
2	16/16	0
3	8/8	0
4	6/8	0
6	8/8	0
8	8/8	0
9	8/8	1

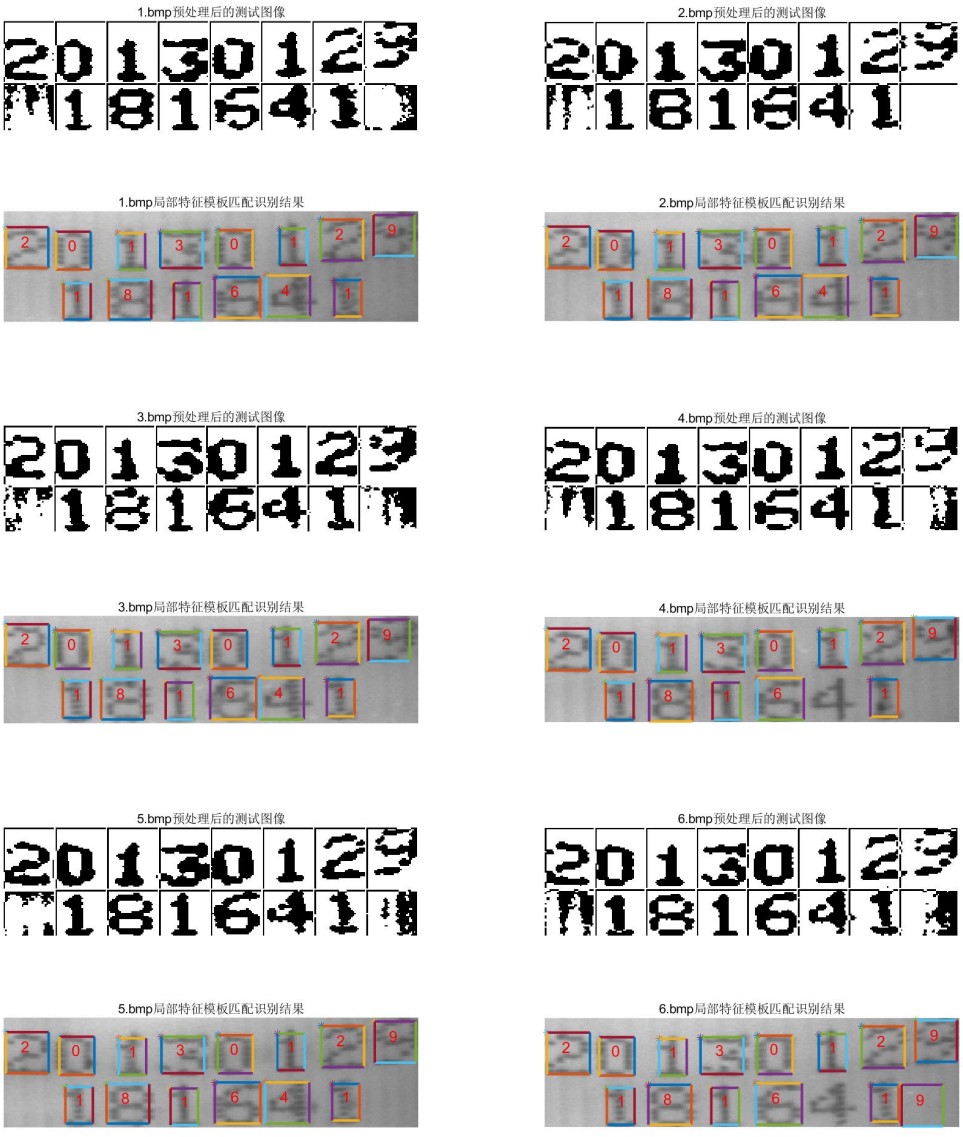

图 3.2: 采用特征提取模板匹配的正常图片识别结果

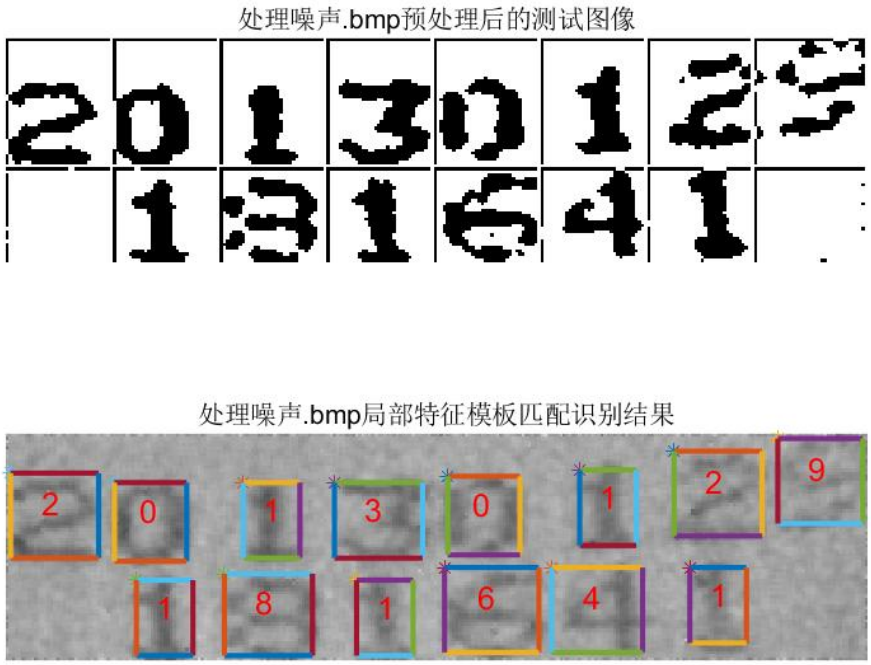

图 3.3: 采用特征提取模板匹配的噪声图片识别结果

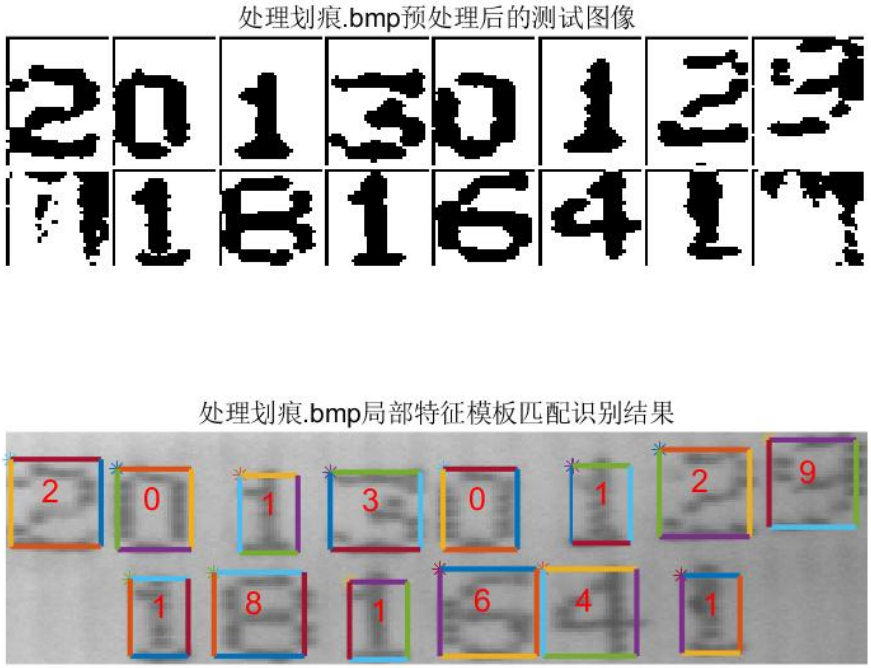

图 3.4: 采用特征提取模板匹配的划痕图片识别结果

第 4 章 总结

本次大作业使用模板匹配的方式对数字字符进行识别，我们主要采用了基于分割的模板匹配、基于滑动窗的模板匹配以及基于特征提取的模板匹配三种算法，并且使用了一些数字图像处理的算法对图像进行了一系列的预处理。

对于训练集的模板图片，我们采用二值化的方式将其转换为二值图片。对于噪声图片我们采用了维纳滤波的方式去除噪声，然后再去除光照影响，最后再二值化用于模板匹配。对于划痕图片，我们首先使用阈值将划痕检测出来，然后将其像素值置为 NaN，然后采用插值的算法填充缺失值，最后再去除光照影响、二值化。从实验结果来看，我们对于噪声图片和划痕图片的预处理是非常成功的，在基于特征提取的模板匹配和基于分割的模板匹配中均达到了 100% 的正确率。

对于基于分割的模板匹配算法，对于特定的图片算法实现简单直观，但是如果给出的测试集图片结构发生变化便需要重新修改代码重新分割；对于滑动窗的模板匹配算法，主要问题是把 0 识别为 1；而基于特征提取的模板匹配算法较好的解决了 0 误识别为 1 的现象。所以三种算法中，基于特征提取的算法更好。特征提取的方法有很多种，例如 SIFT 特征、HoG 特征等，甚至我们可以使用神经网络来提取特征再进行模板匹配，但是这样计算复杂度会增加很多，在本次大作业中没有进行尝试。

对于不同尺度的图片，我们主要采用了多尺度模板匹配的算法，对于放大的图片识别性能较好，对于缩小的图片识别性能欠佳。我们可以通过更改不同尺度下匹配结果的阈值来更加精确的调节。

最后，通过本次作业学习了模板匹配的方法在模式识别中的应用，同时复习了数字图像处理的相关知识。

第 5 章 文件说明

“code/”文件夹中放置的是本次作业用到的代码文件，“train/”文件夹中放置的提供的训练集模板文件，“test/”文件夹下放置的是提供的测试集文件以及**预处理之后的划痕、噪声文件**，“train_bw/”文件夹下放置的是预处理之后的模板文件。以下将对“code/”文件夹下的主要代码进行说明：

- **main.m**：使用基于分割的模板匹配的主函数，直接执行即可得到第2.1节的结果；
- **slideMain.m**：使用基于滑动窗的模板匹配的主函数，直接执行即可得到第2.2节的结果；
- **multiscaleMain.m**：对缩小和放大的图片进行多尺度模板匹配的主函数，直接执行即可得到第2.3节的结果；
- **slideMainFeature.m**：使用基于特征提取的模板匹配的主函数，直接执行即可得到第3章的结果；
- processNoisy.m：对噪声图片进行预处理的代码，并且绘制了处理前后的对比图，并将处理后的图片保存在“test/”文件夹下；
- processScratch.m：使用腐蚀膨胀的算法对划痕图片预处理的代码，实验效果一般；
- processScratchImage.m：使用第一章提到的算法对划痕图片进行预处理的代码，实验效果较好；
- inpaintn.m：使用插值算法对图像缺失值进行填补的算法代码，使用了 Matlab 开源包里的内容；
- processTrain.m：绘制训练集模板图片预处理前后的对比图；
- predictNormalImage.m：使用基于分割的模板匹配算法对正常图片识别的函数；
- predictNoisyImage.m：使用基于分割的模板匹配算法对噪声图片识别的函数；
- predictScrachImage.m：使用基于分割的模板匹配算法对划痕图片识别的函数；
- predictNum.m：使用基于分割的模板匹配算法中预测数字的函数；