**数字图像处理 作业三 实验报告**

**姓名：周敏苑 学号：161220180 邮箱：[161220180@smail.nju.edu.cn](mailto:161220180@smail.nju.edu.cn)**

**运行说明：由于裁剪是按照测试样例的规格来的，所以只支持同测试样例同样规格的图片输入**

**实验要求**

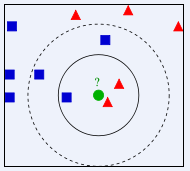
**字符识别：从图片中计算加减算式结果**

**实验内容**

1. **数字识别**
2. **使用knn分类模型识别样本数字：**
3. **KNN原理:**

如下图所示，有两类不同的样本数据，分别用蓝色的小正方形和红色的小三角形表示，而图正中间的那个绿色的圆所标示的数据则是待分类的数据。也就是说，现在， 我们不知道中间那个绿色的数据是从属于哪一类（蓝色小正方形or红色小三角形），下面，我们就要解决这个问题：给这个绿色的圆分类。

　　我们常说，物以类聚，人以群分，判别一个人是一个什么样品质特征的人，常常可以从他/她身边的朋友入手，所谓观其友，而识其人。我们不是要判别上图中那个绿色的圆是属于哪一类数据么，好说，从它的邻居下手。但一次性看多少个邻居呢？从下图中，你还能看到：



* 如果K=3，绿色圆点的最近的3个邻居是2个红色小三角形和1个蓝色小正方形，少数从属于多数，基于统计的方法，判定绿色的这个待分类点属于红色的三角形一类。
* 如果K=5，绿色圆点的最近的5个邻居是2个红色三角形和3个蓝色的正方形，还是少数从属于多数，基于统计的方法，判定绿色的这个待分类点属于蓝色的正方形一类。

​ 于此我们看到，当无法判定当前待分类点是从属于已知分类中的哪一类时，我们可以依据统计学的理论看它所处的位置特征，衡量它周围邻居的权重，而把它归为(或分配)到权重更大的那一类。这就是K近邻算法的核心思想。

​ KNN算法中，所选择的邻居都是已经正确分类的对象。该方法在定类决策上只依据最邻近的一个或者几个样本的类别来决定待分样本所属的类别。

KNN 算法本身简单有效，它是一种 lazy-learning 算法，分类器不需要使用训练集进行训练，训练时间复杂度为0。KNN 分类的计算复杂度和训练集中的文档数目成正比，也就是说，如果训练集中文档总数为 n，那么 KNN 的分类时间复杂度为O(n)。

KNN方法虽然从原理上也依赖于极限定理，但在类别决策时，只与极少量的相邻样本有关。由于KNN方法主要靠周围有限的邻近的样本，而不是靠判别类域的方法来确定所属类别的，因此对于类域的交叉或重叠较多的待分样本集来说，KNN方法较其他方法更为适合。 K 近邻算法使用的模型实际上对应于对特征空间的划分。

1. **Matlab函数介绍：**

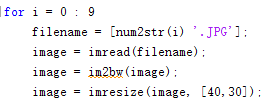
Matlab中自带knn训练函数fitcknn。严格意义上，fitcknn不是一个函数，而是一个类（class），这个类的主要作用是构建一个KNN分类器对象，接下来对本次实验使用到的构造函数和方法成员做介绍：

**构造函数： mdl = fitcknn（X，Y）**返回分类器模型对象,输入参数：X是训练数据，Y是标签。X是一个数值矩阵，每一行表示一个样本数据。Y可以是数值向量，逻辑向量，或者元组等，用于表示每个样本的标签。因此X的行数应该等于Y的长度，都表示样本的个数。

**方法成员：predict（）**使用KNN分类器模型预测标签。

1. **实现步骤：**
2. 手动裁剪训练集图片，并统一格式（40\*30的矩阵）

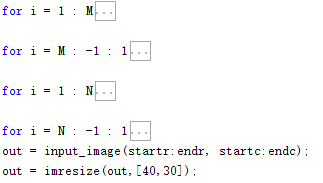




1. 生成训练集矩阵X，将每一个数字的图片转换为1\*1200的矩阵插入到训练集矩阵X中



1. 读取需要识别的图像，对该图像进行切割（除去数字周围的空白），并统一为40\*30的矩阵格式



1. 使用knn函数训练



1. 使用predict识别图片数字



1. **实验结果**

对提供的测试样例的识别结果正确率很高；

这里说一下为什么使用knn而不是其他分类方法，考虑到印刷体数字并不会有很大的变化，所以每个数字的训练集有限（其实每个数字就一张图），所以无法使用那些需要很多训练样本的方法如神经网络，SVM等等，这种情况下，KNN是最佳选择。

1. **使用matlab内部函数ocr识别样本数字：**



一开始使用ocr函数的时候，发现它对只包含单独数字的图片无效，可能是因为空白边缘太大了，后来我在数字的前面添加了两个小黑块，如图：



就能够成功的识别出数字。尽管如此，这毕竟是一种取巧的方法，鲁棒性并不很强，换一张图片，换一个尺寸估计又不行了，所以第一种knn算法还是第一选择。

1. **符号识别**

**方法同数字识别**

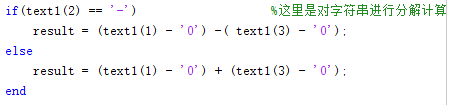
1. **识别图片并计算结果：**
2. **把整张图片切分为一格一格的小图片，每张图片包含一个算式：**



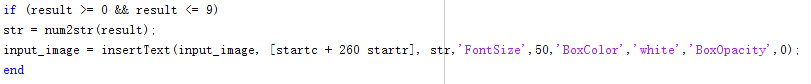
1. **可以用ocr函数识别整个式子，也可以用自己写的两个函数分别识别数字和符号（需要继续切分）：**



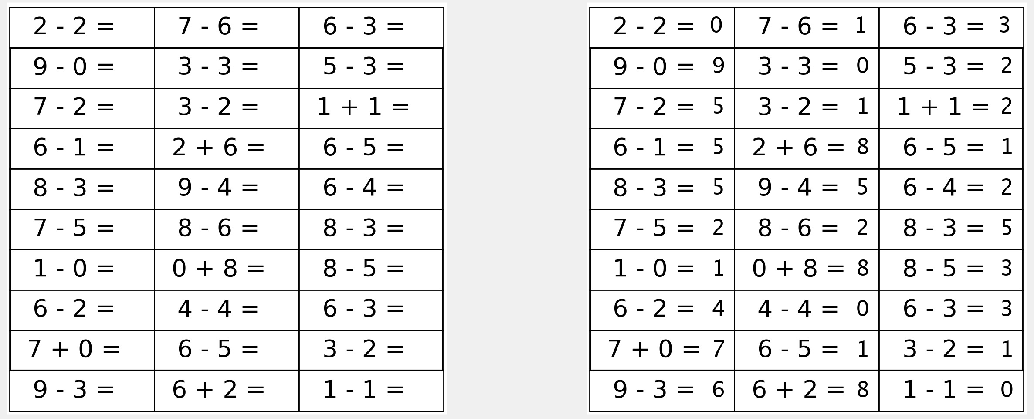
1. **根据识别出来的数字和符号，计算出结果：**



1. **将计算出来的结果添加到图片上得到最终结果图：**

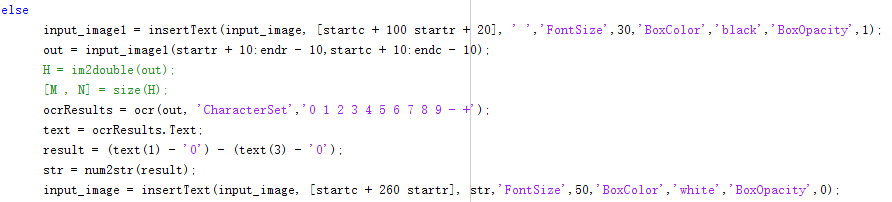


1. **最终结果图：**



**实验中遇到的问题：**

实验前面做的一切顺利，但是到了最终输出结果的时候，发现凡是包含“- 1”的式子都无法识别，于是我对包含“- 1”的式子进行了特殊处理：



特殊处理的方式，就是将算式中间的‘-’号去掉，这样就能识别出剩下的两个数字，因为已知只有‘- 1’不能被识别，那么中间的符号一定是‘-’号，这样一来，只需要将得到的两个数字相减就能得到最终的结果，然后按照之后的步骤将结果添加到图片上即可。

**实验心得**

这次实验本身是一个充满趣味的实验，需要识别数字符号，然后将结果打印到图片上。在做这个实验之前，毫无头绪，本来在数据挖掘课上讲过了knn算法，但没有实现过，也不知道到底是个什么情况。通过本次实验，虽然也是调用了matlab自带的knn函数，但不管怎样，最终通过knn实现了图片分类模型，最终利用这个模型可以识别很多数字字符，还是很有成就感的。