
Project3

手写数字的检测与识别

12330346 - 谢志旺 - 2015年6月22日



摘要

这个project是使用 adaboost 对图片进行检测。目的是检测出图片上的数字。然后利用 svm 算法进行数字的识别。在我的代码中，adaboost 和 svm 都是使用 opencv 自带的实现。我将图片的检测和识别结果直接显示在图片中。

数字检测

样本

对图片中的手写数字的检测是使用 opencv 的 adaboost 算法。正样本使用的是 minst 数据集。负样本是自己从网上找到的以白纸为背景，纸上为手写的汉子或英文的图片。共 500 张。然后我从这 500 张图片中随机取出一定大小的不重合的 $40 * 40$ 的小图像块作为负样本。到最后一共得到 10000 张负样本。

特征

因为 opencv 的 adaboost 只支持 haar 特征，所以这里使用 haar 特征进行检测。

训练

这里使用 10000 个正样本和 5000 个负样本对模型进行训练，得到描述文件。

检测

使用训练好的模型对一张输入图片进行检测，将检测的结果作为识别器的输入。

数字识别

样本

对手写数字的识别采用的是有 opencv 实现的 svm 算法。样本采用的是 minst 数据集。一共使用 60000 个样本进行训练。

特征

一开始采用的方法是直接将所以像素点的像素值作为特征值进行训练，这个时候就有 $28 * 28$ 个特征。但是发现训练的时间较长并且模型的准确度较差，基本上只有 72%。然后我才用的方法是，先获取图片的数字部分，比如如下图片：



Fig 1: 手写数字原图

处理后变成如右：



Fig 2: 手写数字处理图

然后再将图片正规化，也就是变为正方形。结果为：



Fig 3: 手写数字处理图

最后将图片压缩成 8*8 大小，这是根据字体形状来确定的，在经过上面的处理后，所有的字体都处于下面图的中央：



Fig 4: 数字位置模板

因此我的处理特征的方法是，取一个字，选取一个包含该字的正方形区域，将这个正方形区域分割成8*8个小格，统计每个小格中像素的数量，以这些数量为维度进行映射。

训练

使用 minst 数据集进行训练。在如下计算机配置下训练时间是：1 min 28.14 s

计算机配置：

处理器：2.3GHz intel Core i7

内存：16GB 1600 MHz DDR3

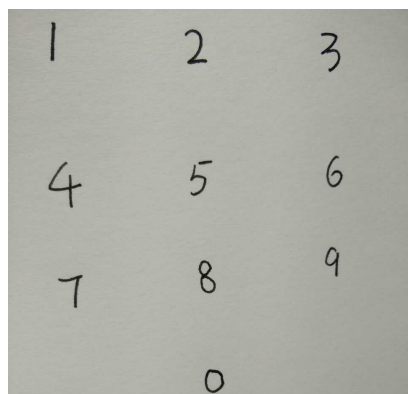
识别

直接将经过特征处理的图片特征作为输入，模型会进行分类得到结果。经过 minst 的测试数据集的测试，准确度为 93%。

手写数字的检测与识别

样本

将图片放在 image 文件夹下，然后运行程序，就可以在 result 文件夹下看到结果。结果的形式是用绿色椭圆将图片中的数字框出来，并在旁边显示识别器得到的值。比如如下：



检测和识别情况为：

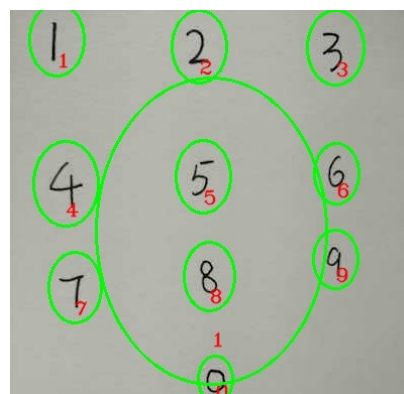


Fig 5: 图像检测和识别情况

具体运行情况和文件夹分布情况请看 readme 文件。