

CV_HW2 Report

马逸川 520030910393

日期: 2022/10/17

1 简介

按照作业要求实现了所需的各项功能, 下面是我各项函数的设计思路。提交的代码文件包括 *homework.py* 和 *homework.ipynb*, 前者是完整的实现, 后者展示了代码的各项运行结果。

2 TODOS

2.1 TODO1

第一部分: 实现了特征提取和整合。这一部分的实现较为简单, 首先调用 *cv2.SIFT_create()* 得到每张输入图片的特征信息, 记录 *data.train_lb[i]* 为训练数据所在的类别, 使用 *extend* 函数依次将对应的 *kpts* 和 *des* 信息添加到列表中即可。

2.2 TODO2

第二部分: 在每个类别中选择相同数目的特征点进行聚类。直接将特征点添加在列表中, 结果得到了一个大小为 (344658, 128) 的特征数据集。

2.3 TODO3

第三部分: 设置聚类数目, 将得到的特征进行聚类处理, 得到聚类特征 (即图像的 word)。在本次实验中, 总共尝试了 30, 50, 100, 150, 200, 300, 500 几种聚类数目, 其中 *N_clusters = 200* 时得到了最佳的分类结果。总体的分类结果展示在报告的最后部分。

2.4 TODO4

第四部分: 构建训练特征向量。在本部分中, 首先利用之前得到的 *kmeans* 聚类, 调用 *kmeans.predict()* 得到训练数据中每个维度的聚类, 从而构建图像的特征向量。调用 *cv2.cv2.normalize* 函数将得到的图片特征向量归一化到 (0,1) 范围内。得到训练数据的特征向量后, 调用 *SVM* 进行分类, 代码如下:

```
1
2 from sklearn import svm
3 classifier = svm.SVC(probability=True)
4 classifier.fit(hist_vector, data.train_lb)
```

2.5 TODO5

第五部分: 训练并验证。在本部分中, 通过同样的方式将图像特征映射到聚类特征上, 得到测试数据的特征向量, 使用 `classifier.predict` 得到训练的 SVM 对测试数据的分类结果。在 `homework.py` 中简单尝试了用多层感知机完成分类任务, 结果并不是很理想, 训练中观察到 loss 在下降到 3.64 左右后不再变化。由于时间原因, 在这个思路上没有再做更多的尝试。

3 Result

训练结果如图。

num_clusters	accuracy
30	0.2941
50	0.3262
100	0.3626
150	0.3688
200	0.3852
300	0.3684
500	0.3797

最好结果: 使用 SVM, `N_clusters` 值设置为 200。最终最好结果为 0.3852。`n=200` 时最佳, 我认为原因在于:`n` 太小时, 特征不足以反映完整信息, `n` 太大时, 维度太高, 数据量不足难以分类, 导致错误率的提升。所以 200 时得到最佳的性能。

然而模型的性能并不好, 我认为原因是提取的总特征量不够, 导致聚类结果并不能完整反映图像信息导致的。