

第二次课程作业文档

2017211416 张志博

作业说明

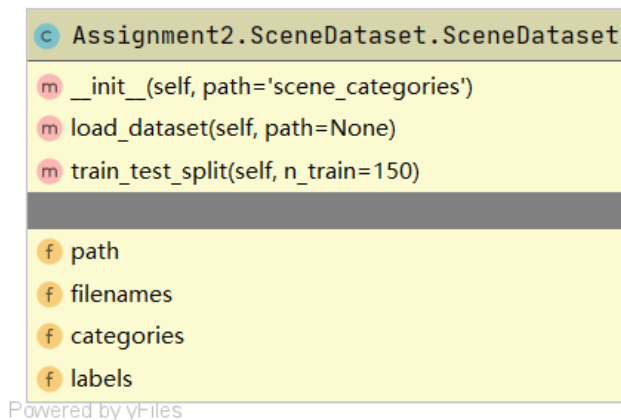
1. **任务：**编写一个图像分类系统，能够对输入图像进行类别预测。具体的说，利用数据库的 2250 张训练样本进行训练；对测试集中的 2235 张样本进行预测。
2. **数据库说明：**scene_categories 数据集包含 15 个类别（文件夹名就是类别名），每个类中编号前 150 号的样本作为训练样本，15 个类一共 2250 张训练样本；剩下的样本构成测试集合。
3. **使用知识点：**SIFT 特征、K-means、BOW、SVM

算法流程

1. **数据集预处理：**按要求读取并将数据分类成训练集和测试集。
2. **提取 SIFT 特征：**提取 2250 张训练样本的 SIFT 特征（实际使用了更好的 SURF 特征）。
3. **K-means 聚类：**因为 SIFT 特征的特征点个数对于每张图片都是不一样的，因此没有办法直接进行 SVM 分类，所以使用 K-means 方法对 SIFT 特征进行聚类，结果就作为基于 BOW 的特征提取器的词汇。
4. **SVM：**使用基于 BOW 提取的特征结果训练 SVM 分类器。
5. **分类输出：**先提取基于 BOW 的特征，再使用 SVM 进行预测即可。

函数说明

1. Python 类 Scene Dataset



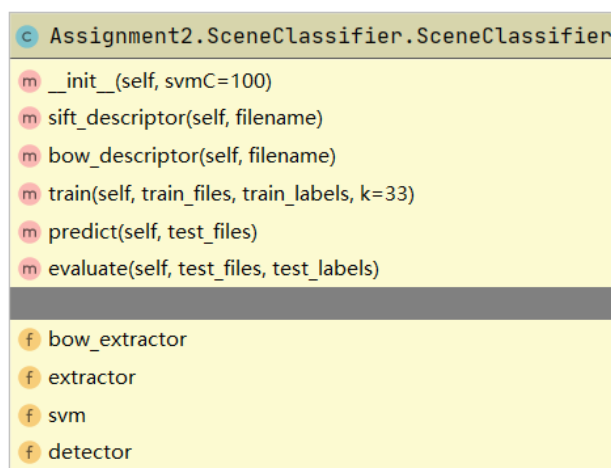
- a) 示意图：
- b) 属性

- i. **path**: 用于加载数据的 scene_categories 数据集的路径。
- ii. **filenames**: 所有数据样本的路径。
- iii. **labels**: 所有数据样本的类别，以 0 到 14 这 15 个数字表示。
- iv. **categories**: 数据样本类别的具体名称。

c) 方法

- i. **load_dataset**
 - 1. 输入参数: 用于加载数据的 scene_categories 数据集的路径 path。
 - 2. 功能: 根据所给路径, 按指定格式处理, 得到并返回 filenames、labels、categories。
- ii. **train_test_split**
 - 1. 输入参数: 训练集的大小 n_train。
 - 2. 功能: 按题目要求, 将每类的前 n 个样本 (filenames 以及 labels) 划分至训练集, 其他划分至测试集, 返回训练集和测试集。

2. Python 类 Scene Classifier



a) 示意图:

b) 属性

- i. **detector**: 用于提取关键点。
- ii. **extractor**: 用于提取描述符 (特征)。
- iii. **bow_extractor**: 基于 BOW 的描述符提取器。
- iv. **svm**: 分类器。

c) 方法

- i. **sift_descriptor**
 - 1. 输入参数: 需要分析样本的路径 filename。
 - 2. 功能: 返回描述符。
- ii. **bow_descriptor**
 - 1. 输入参数: 需要分析样本的路径 filename。
 - 2. 功能: 返回基于 BOW 的描述符提取器计算的描述符。
- iii. **train**
 - 1. 输入参数: 训练数据集 train_files, 训练标签集 train_labels, K-means 参数 k。
 - 2. 功能: 提取数据集 SIFT 特征, 进行 K-means 聚类获得 BOW 的词汇, 再提取基于 BOW 的描述符, 最后配合利用标签集训练 SVM。

- iv. **predict**
 1. 输入参数：测试数据集 test_files
 2. 功能：提取基于 BOW 的描述符，再使用 SVM 对其分类，返回测试标签集。
- v. **evaluate**
 1. 输入参数：测试数据集 test_files，测试标签集 test_labels
 2. 功能：根据测试集的预测结果计算出混淆矩阵，plots 输出混淆矩阵图像，打印输出 macro、weighted、micro 总 3 种 F1-score，返回混淆矩阵。

实验设置

注：下面表格中 F1-Score(micro)即为论文中使用的 Accuracy-Score，未标注的则默认采用 F1-Score(micro)，即 Accuracy-Score。

1. SIFT or SURF

a) 实验设置及结果

- i. 无关变量：SVM 使用 RBF 核，参数 C=100，参数 Gamma=1

	K-means 的参数 K	F1-Score(macro)	F1-Score(micro)
SIFT	100	0.491198869	0.523937360
SIFT	1000	0.549631654	0.568680089
SURF	100	0.508513569	0.546756152
SURF	1000	0.560643730	0.597762864

- ii. 结论：SURF 效果优于 SIFT（时间效率上也是）。

2. K-means 的参数 K

a) 实验设置及结果

- i. 无关变量：使用 SURF 特征，SVM 使用 RBF 核，参数 C=100，参数 Gamma=1

K-means 的参数 K	F1-Score(macro)	F1-Score(micro)
33	0.480419618	0.515436242
50	0.473135600	0.511409396
100	0.508513569	0.546756152
500	0.540352371	0.578970917
1000	0.560643730	0.597762864

- ii. 结论：K=33 时可以取得局部较优的结果，K 远大于 33 时，K 越大效果越好。

3. SVM

a) 实验设置及结果

- i. 无关变量：使用 SURF 特征

- ii. 注：此部分比较方式不够严谨，仅供大致比较和示意。

K-means 的参数 K	SVM 核	SVM 的参数 C	SVM 的参数 Gamma	F1-Score(macro)	F1-Score(micro)
100	RBF	100	1	0.508513569	0.546756152
1000	RBF	100	1	0.560643730	0.597762864
100	LINEAR	100		0.493116707	0.535570470

1000	LINEAR	100		0.545162083	0.578970917
100	RBF	10	1	0.472162080	0.513646532
100	RBF	1000	1	0.469760993	0.504697987
100	RBF	100	0.01	0.324843290	0.355704698
100	RBF	100	0.1	0.471007479	0.511856823
100	RBF	100	10	0.478184389	0.511409396

iii. 结论：SVM 中，RBF 核优于 LINEAR 核，C=100 和 Gamma=1 是一组不错的参数。

4. 与论文

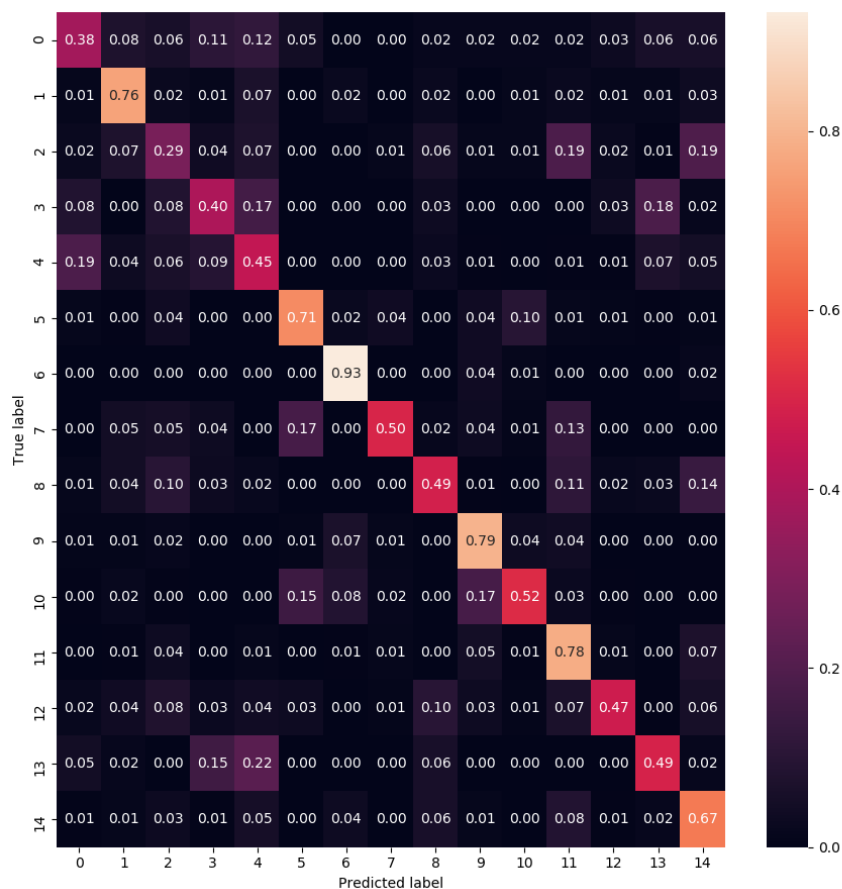
a) 结果如下表，性能略微有所提升，但在 K 较高时效果仍没有明显改善。

K (或 M)	我的版本	论文版本	复现效果
16	0.476	0.543	87%
200	0.558	0.722	77%
400	0.571	0.748	76%
1000	0.598		

b) 结论：我的版本在 K 相同时性能与论文有差距，这个差距会随 K 增大而增大。

结果分析

1. 混淆矩阵示例



2. 实验中各个环节和各个参数对最终性能的影响

a) SIFT or SURF

- i. SURF 相较于 SIFT，特征类似，但鲁棒性更强，最重要的是速度快，在大规模运算时相比 SIFT 具有压倒性的优势，实际测试也是如此。

b) K-means

- i. K-means 的参数 K 对试验结果影响很大，本以为可以找到一个合适的数值，比如 $(\text{训练样本数量}/2)^{0.5}$ 即 33 或 34，可以保证 k 较小即计算量较小的情况来得到较好的实验结果，但在实验中发现 K 几乎是越大效果就越好，尽管这会花费更多训练时间。如果想尝试改善性能，还可以考虑其他聚类方法。
- ii. 但是在论文中，使用了一个更小的训练集(每类 100，不清楚是否是随机划分)，K 从 16 提升至 200 就有一个很大的提升，但我的实现中 K 从 16 提升至 1000 的提升幅度仍很有限，暂不清楚是否有其他影响因素。

c) SVM

- i. 在 SVM 的 Kernel 中，RBF 核是广泛使用且效果较好的核，但也有说法称当样本特征较多时，LINEAR 核的计算更加高效，但经过测试，发现计算量减少难以掩盖训练效果较差的结果，因此最终还是选择使用了 RBF 核。参数 C 经实验发现在 100 时效果较好，Gamma 则保持在默认的 1 即可。如果需要更加好的参数，可能需要自动化的参数寻优方法。

d) Level 与 Pyramid

- i. 因为基本的性能并未达标，且暂未找到与论文性能存在差异的原因，为减少其他变量以便测试，再加上以原论文中所述，要想提升性能，增加特征量是最有效的，所以暂未实现这两项功能。