《计算机视觉与图形图像处理》

Programming Assignment 2:

基于 BOW 的图像分类

南京农业大学人工智能学院 黄君贤

PA2 due 2021.11.30 11:00PM

计算机视觉的一项重要应用场景在于对图片分类，本次作业采取的分类方案为 BOW（Bag of Words）。BOW 最早应用于 NLP 文档分类，BOW 的原理在于通过对词频的统计来对比文档的相似度。将 BOW 应用于图片分类，最大的改变在于，原来对单词的处理，需要变更为对图片的特征描述向量（特征描述子）的处理。
对应于 BOW 的流程，本次作业将分为三个主要部分，相关步骤你可能需要用到 sklearn, opencv 库，训练和测试的图片见 data 文件夹。

预备操作

1）将分类标签映射为数字，方便机器学习算法处理，映射关系
{'airport': 0, 'auditorium': 1, 'bedroom': 2, 'campus': 3, 'desert': 4, 'football_stadium': 5, 'landscape': 6, 'rainforest': 7}
2）从 data 文件夹中读取所有图片路径和对应的数字标签，并按照 8：2 切分训练集和测试集。定义 X_train, X_test, y_train, y_test 分别为训练集的图片路径数组，测试集的图片路径数组，训练集的对应数字标签，测试集的对应数字标签，图片路径和标签一一对应。例如'data/desert/sun_atyasoydrjavyfux.jpg' 对应数字标签 4。注意，为了防止干扰，在切分训练和测试集之前一般要先打乱数据。

1. 构建视觉码本（Visual Words Dictionary）

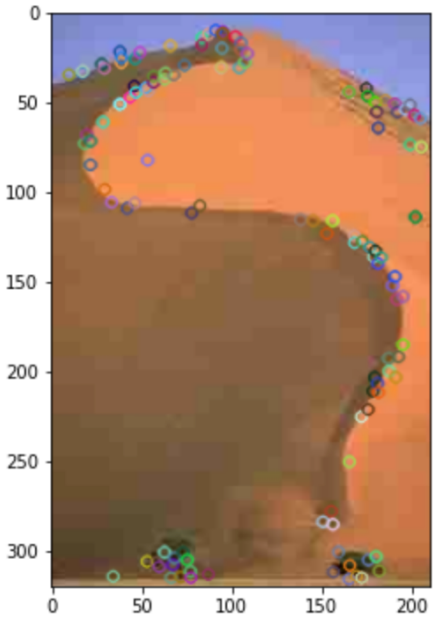
1.1 提取图片特征点

常用的图片特征提取算法有 Harris 角点检测（Harris Corner），SIFT 即尺度不变特征变换（Scale-invariant feature transform，专利到期，可商用），SURF（SIFT 改进版本，计算量小，运算速度快，提取的特征点几乎与 SIFT 相同，但专利保护），ORB（Oriented FAST and Rotated BRIEF），BRISK 等。

ORB 作为 SIFT 和 SURF 的有力替代者，非常适用于对计算实时性要求比较高的场景，而且可以免费商用。作为学术研究，本次作业可以自由选择特征提取算子（包括以上未列出的）。

为了直观感受对图片的特征提取，选取 data 文件夹中的一张图片，使用特征提取算子，提取图片中的特征点（key point）和特征子（descriptor），并将特征点绘制于原图中，

效果应当类似下图所示，对应的每个特征子都应当是相同大小的一维数组，比如 sift 提取的每个特征子是 1*128 的数组，orb 是 1*32 的数组



1.2 批量提取图片特征子

编写函数，批量提取图片特征子

def get_images_descriptors(detector, image_path_array, ori_labels)

入参：

detector 表示特征提取算子，方便后期更换算子比较效果

image_path_array 图片路径数组

ori_labels 和图片一一对应，表示图片的分类标签

返回：

=> 图片特征子集合数组，图片特征子集合是二维数组（n * dim，n 为检测到的特征点个数，dim 表示特征子维度），整体返回的应为三维数组

=> 图片标签数组，注意，算子可能无法从某个图片中找到特征点，此处要和图片特征子集合形成对应关系

1.3 堆叠特征子

编写函数，重新组织图片的所有特征子，生成二维数组，每一行为图片中的单个特征子

def vstack_descriptors(descriptors_list)

入参 descriptors_list 为图片特征子集合数组，返回堆叠后的二维数组

1.4 聚合特征子

编写函数，通过 K-means 对所有的特征子聚合

def cluster_descriptors(descriptors, no_clusters)

入参：

descriptors 表示堆叠后的特征子二维数组

no_clusters 表示需要将所有的特征子聚合成多少分组
返回
训练完成的 K-means 模型，每个分组的质心就是 Visual Word

1.5 构建视觉码本
对 1.2 步应用 X_train 数据后，再分别执行 1.3 和 1.4 步，1.4 步的执行结果 kmeans 模型中持有视觉码本数据，将该模型序列化到磁盘，以便后期直接使用。

2. 构建图片分类器
2.1 提取图片特征子在码本中的分布
定义函数，将每个图片的特征子集合转化成码本中 Visual Word 的频次集合：通过 kmeans 将每个特征子归类到所属 Visual Word 分组，并累计所属 Visual Word 的频次

def extract_features(kmeans, descriptors_list, no_clusters)
入参：
kmeans 即训练完的 kmeans 模型
descriptors_list 对应图片特征子集合数组
no_clusters 表示 kmeans 聚合时设置的分组数
返回
频次集合的数组，注意，每个图片都有一个对应的频次集合，而且大小一致，都是 no_clusters 个元素，集合的下标对应于某个 Visual Word，集合下标对应的数值即该 Visual Word 的频次。

2.2 对数据标准化处理
后一步需要对数据应用 SVM 分类算法，执行之前需要对数据的每个属性（列）分别进行标准化处理。
注意，保留应用于训练集的标准化处理对象（或参数），后期测试集应当使用同一套标准化参数预处理。

2.3 训练分类器
定义函数，使用 SVM 分类器针对已经提取出的图片码本分布，进行分类训练
def train_SVC(features, train_labels)
入参：
features 标准化处理后的频次集合数组
train_labels 对应的数字标签
返回
训练完成的 svm 分类器模型

将训练完成的模型序列化到磁盘，方便以后直接使用。

思考：SVM 分类器包含多个超参数（hyper parameter），例如核函数（可选 linear、poly、rbf、sigmoid 等），惩罚系数 C（一般 0.1, 1, 10 等），某些核函数还自带参数 gamma（一般 0.1, 0.01, 0.001, 0.0001 等），如何选择最佳的组合？选择的过程中能否直接用测试集来评估？

3. 评估分类器

3.1 计算整体正确率

对 X_test 应用 2.1 和 2.2 的步骤，用 2.3 的 svm 分类器对测试集图片预测分类，并和真实的分类进行比较，计算整体正确率。

3.2 绘制混淆矩阵（Confusion Matrix）

绘制混淆矩阵，查看单个类别的正确率，以及错误识别的部分在其他类别的分布，其效果应类似下图所示