

目录

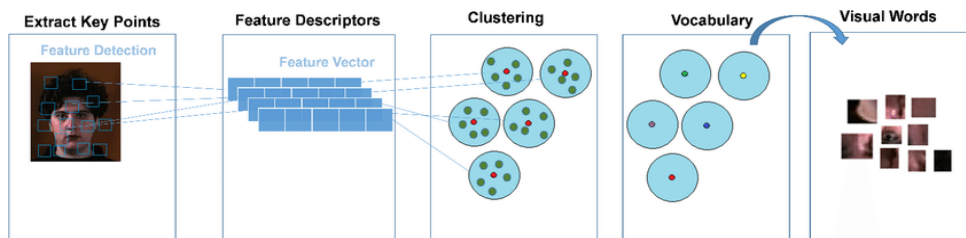
一、上机要求与目的	2
1.1 CV 与词袋模型.....	2
1.2 SIFT 算法相关	3
1.3 实验具体要求（两个数据集）	3
二、上机内容	3
2.1 方法与步骤	3
2.1.1 划分训练集与测试集；	3
2.1.2 使用 SIFT 对每幅图像进行特征采样与描述	4
2.1.3 使用 K-means 对提取特征聚类，生成 codebook.....	5
2.1.4 图像特征表示的生成	5
2.1.5 使用分类器对图像特征表示进行分类	7
2.1.6 参数调整与测试	8
2.2 结果及分析	8
2.2.1 实验参数设置	8
2.2.2 KNN 分类器准确率-K- SIFT 特征种类	9
2.2.3 图像分类准确率-SIFT 特征种类	10
2.2.4 图像分类准确率-KNN 的 K 取值	10
2.2.5 KNN 自动超参数优化	11
三、总结	13

一、上机要求与目的

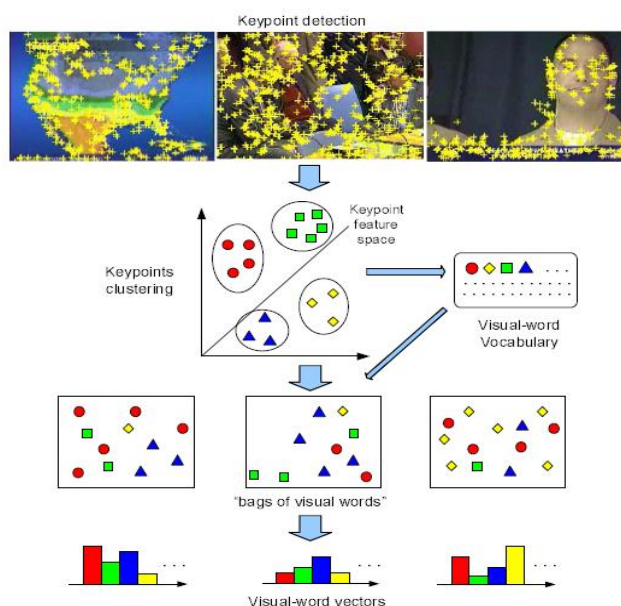
1.1 CV 与词袋模型

词袋模型最初用于文本分类中，然后逐步引入到了图像分类任务中。在文本分类中，文本被视为一些不考虑先后顺序的单词集合。而在图像分类中，图像被视为是一些与位置无关的局部区域的集合，因此这些图像中的局部区域就等同于文本中的单词了。在不同的图像中，局部区域的分布不同。因此，可以利用提取的局部区域的分布对图像进行识别。

2003 年以来，BoW 出现在 CV 中，如图像分类、图像检索等。其大概过程首先提取图像集特征的集合，然后通过聚类的方法聚出若干类，将这些类作为 **codebook**，即相当于 **words**，最后每个图像统计 **codebook** 中 **words** 出现的频数作为输出向量，就可以用于后续的分类、检索等操作。



以 sift 特征为例，假设图像集中包含人脸、自行车、吉他等，我们首先对每幅图像提取 sift 特征，然后使用如 **kmeans** 等聚类方法，进行聚类得到码本(dictionary)。之后在每一幅图像中统计 sift 特征点在码本上的频数分布，得到的向量就是该图像的 BoW 向量。最后就可以使用这些向量训练分类器，利用图像中单词的分布进行图像分类。



图像分类和文本分类的不同点在于，在文本分类的词袋模型算法中，字典是已存在的，不需要通过学习获得；而在图像分类中，词袋模型算法需要通过监督或非监督的学习来获得视觉词典。

1.2 SIFT 算法相关

SIFT 算法的核心是将一幅图像用描述子表示，这些特征点具有尺度不变性，就相当于图像本身，但是相对于图像文件本身，在计算机中处理起来方便的多了。详细的 SIFT 实现过程分为如下四个步骤：

- 1、检测尺度空间的极值点；
- 2、抽取稳定的关键点；
- 3、为每个关键点指定一个或者多个方向；
- 4、生成特征点描述子。

1.3 实验具体要求（两个数据集）

1. 测试不同的 SIFT 特征提取方式测试图像分类准确率；
SIFT 特征种类：sift, dsift, msdsift
2. 测试 codebook 维数设置为 50, 100, 150, 200, 300, 500 时测试图像分类准确率；
3. 测试 SVM 分类器不同核函数对应的图像分类准确率；
SVM 核函数种类：LINEAR, RBF, Chi²
4. 测试 KNN 分类器不同 K 值对应的分类准确率；（K 可取 1, 5, 10, 20 等）

二、上机内容

2.1 方法与步骤

2.1.1 划分训练集与测试集；

在./handsonbow/img 中存放数据集（4_ObjectCategories 和 15_ObjectCategories）

- 4_ObjectCategories 数据集包含 4 个类别：airplanes, cars, faces 和 motorbikes, 每个类别包含 450 张图像。
- 15_ObjectCategories 数据集包含 15 个类别：bonsai, butterfly, crab, elephant..... 最少的类别包含 64 张图像。

按照 8：2 的比例划分训练集与测试集：

数据集	num_train_img/classes	num_test_img/classes
4_ObjectCategories	360	90
15_ObjectCategories	52	12

参数设置位于 `exercises_solutions.m` 的第 74-77 行：

```
74 % number of images selected for training (e.g. 30 for Caltech-101)
75 num_train_img = 360;
76 % number of images selected for test (e.g. 50 for Caltech-101)
77 num_test_img = 90;
```

2.1.2 使用 SIFT 对每幅图像进行特征采样与描述

功能函数：`extract_sift_features(dirname,file_ext)`

可以选择的三种 SIFT 特征种类：

`sift`(SPARSE SIFT), `dsift`(DENSE SIFT), `msdsift`(MULTI-SCALE DENSE SIFT)

参数设置位于 `exercises_solutions.m` 的第 37-39 行

```
37 % desc_name = 'sift';
38 % desc_name = 'dsift';
39 desc_name = 'msdsift';
```

SIFT 特征提取主要步骤描述：

（代码详见 `detect_features_dsift.m`）

1. 尺度空间极值检测：搜索所有尺度上的图像位置。通过高斯微分函数来识别潜在的对于尺度和旋转不变的兴趣点。
2. 关键点定位：在每个候选的位置上，通过一个拟合精细的模型来确定位置和尺度。关键点的选择依据于它们的稳定程度。
3. 方向确定：基于图像局部的梯度方向，分配给每个关键点位置一个或多个方向。所有后面的对图像数据的操作都相对于关键点的方向、尺度和位置进行变换，从而提供对于这些变换的不变性。
4. 关键点描述：在每个关键点周围的邻域内，在选定的尺度上测量图像局部的梯度。这些梯度被变换成一种表示，这种表示允许比较大的局部形状的变形和光照变化。

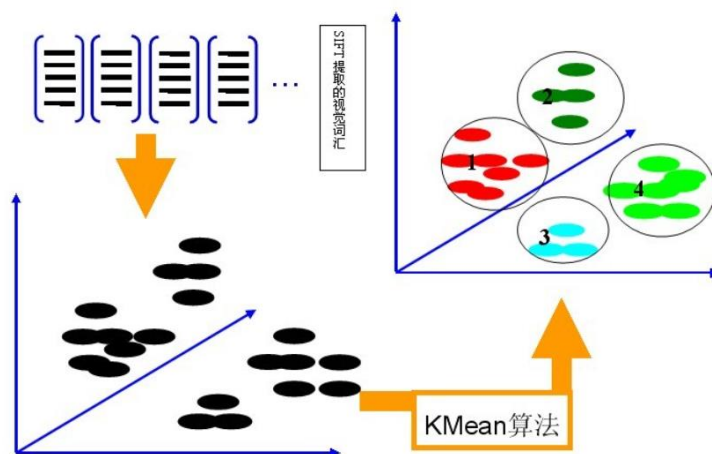
加载训练集与测试集的 SIFT 特征，可视化效果如下：



2.1.3 使用 K-means 对提取特征聚类，生成 codebook

利用 K-Means 算法构造 codebook。K-Means 算法是一种基于样本间相似性度量的间接聚类方法，此算法以 K 为参数，把 N 个对象分为 K 个簇，以使簇内具有较高的相似度，而簇间相似度较低。SIFT 提取的视觉词汇向量之间根据距离的远近，可以利用 K-Means 算法将词义相近的词汇合并，其聚类中心作为 codebook 中的基础词汇。

过程如下图所示：



参数 K 的设置位于 exercises_solutions.m 的第 79 行

```
78 % number of codewords (i.e. K for the k-means algorithm)
79 nwords_codebook = 500;
```

对应代码部分位于 exercises_solutions.m 的 169-203 行。

核心函数在 kmeans_bo.m 中实现

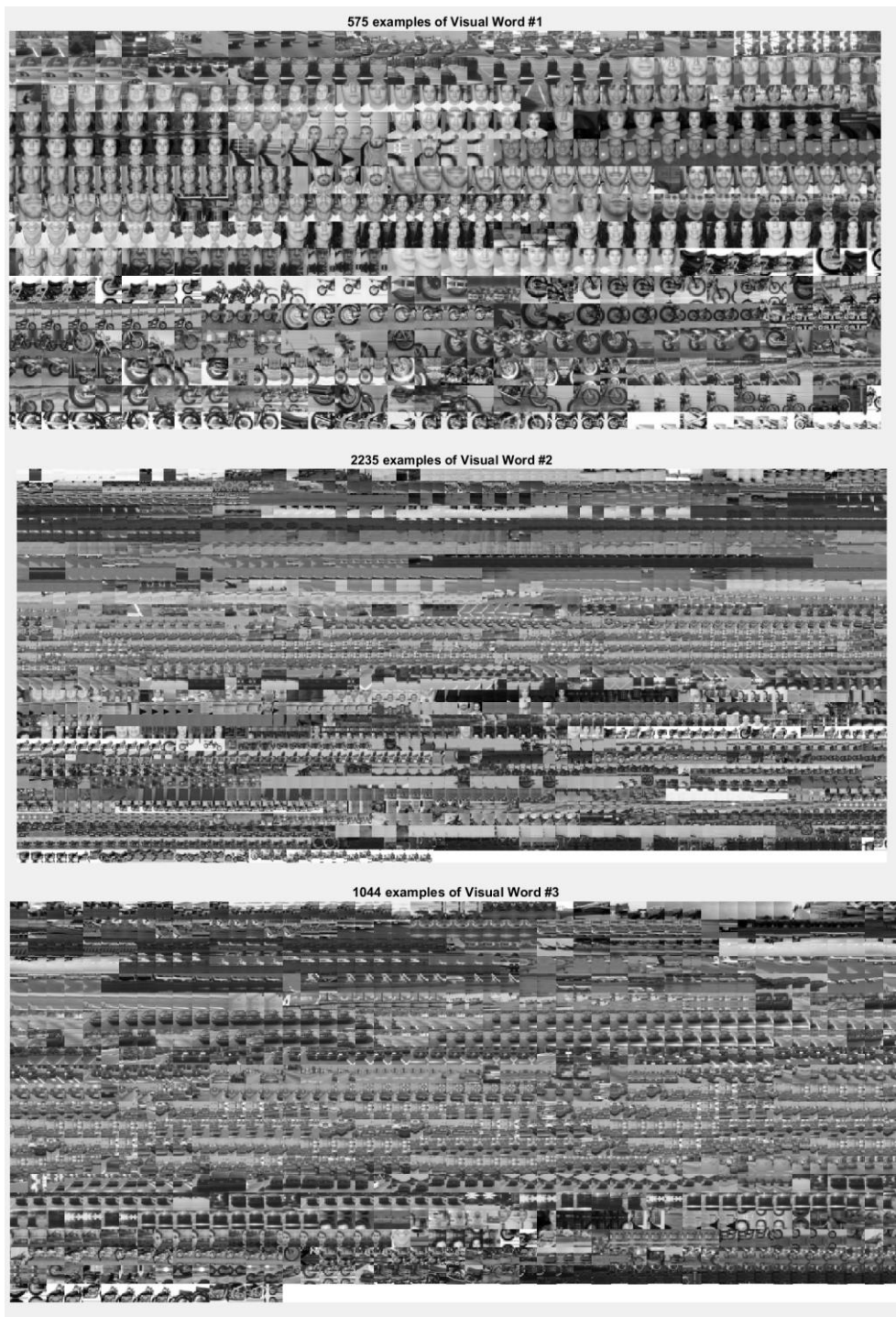
2.1.4 图像特征表示的生成

计算每幅图像的 SIFT 特征与 codebook 中每个特征的欧式距离，依据图像中每个特征距离 codebook 中特征的最近距离，给所有的图像的 SIFT 特征定量。

对应代码部分位于 exercises_solutions.m 的 214-252 行。

为了在视觉上验证上面计算的结果，可以显示与同一 visual words 相对应的 patches。

对应代码部分位于 exercises_solutions.m 的 256-287 行。



在 `codebook` 中的特征直方图上记录对应的特征数目，为每幅图像生成直方图，并使用 `L1` 范数对结果进行正则化，最终结果作为该图像的特征表示。

对应代码部分位于 `exercises_solutions.m` 的 302-339 行。

注：Matlab 的函数 `'histc'` 可以用来计算直方图

2.1.5 使用分类器对图像特征表示进行分类

本次实验依据实验要求采用 K-NN 分类器，使用欧氏距离计算，权重均等。

K-NN 分类器在接受到训练数据后，只是对训练数据进行简单的存储，并不构造分类模型，在接受到待分类数据时，KNN 通过计算待分类数据 X 与所有训练数据之间的距离，选择前 K 个距离 X 最近的数据，并将这 K 个距离最近的数据中出现次数最多的类属性赋给 X。

对应代码部分位于 exercises_solutions.m 的 379-406 行。

参数 K 的设置位于 exercises_solutions.m 的第 389 行

关键代码部分如下所示：

```
K = 20;
rng(1);
kNNClassifier = fitcknn(bof_train, labels_train, 'NumNeighbors', K);
% kNNClassifier = fitcknn(bof_train, labels_train, 'NumNeighbors', K, 'OptimizeHyperparameters', 'auto', ...
% 'HyperparameterOptimizationOptions', ...
% struct('AcquisitionFunctionName', 'expected-improvement-plus'));
bof_l2lab = predict(kNNClassifier, bof_test);

method_name = 'NN L2';
acc = sum(bof_l2lab == labels_test) / length(labels_test);

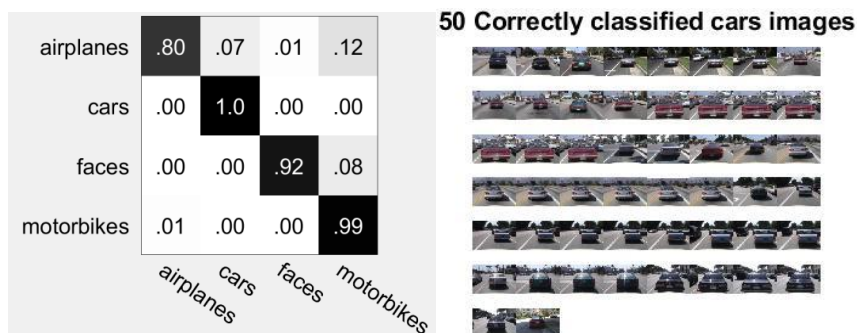
fprintf('\n*** %s ***\nK=%d\nAccuracy = %1.4f%% (classification)\n', method_name, K, acc*100);
```

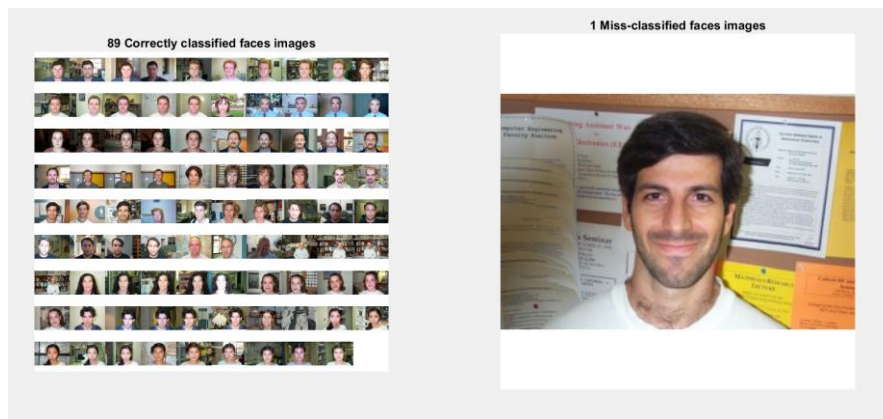
fitcknn 函数主要参数设置如下：

kNNClassifier	
1x1 ClassificationKNN	
Property ^	Value
NumNeighbors	20
Distance	'euclidean'
DistParameter	[]
IncludeTies	0
DistanceWeight	'equal'
BreakTies	'smallest'
NSMethod	'exhaustive'

分类结果包括：

1. KNN 分类器准确率
2. 各类图像的分类情况、准确率与错误分布





2.1.6 参数调整与测试

在整个实验过程中，有一些重要的参数直接影响到了最后分类的效果。故对这方面进行了大量的实验分析，实验数据及结论分析详见结果与分析部分。

- 测试不同的 SIFT 特征提取方式测试图像分类准确率；
- 测试 KNN 分类器不同 K 值对应的分类准确率。

2.2 结果及分析

2.2.1 实验参数设置

	4_ObjectCategories	15_ObjectCategories
num_train_img	360	52
num_test_img	90	12
k-means 最高迭代次数	50	50
number of codewords	500	500

改变三种 SIFT 特征种类：

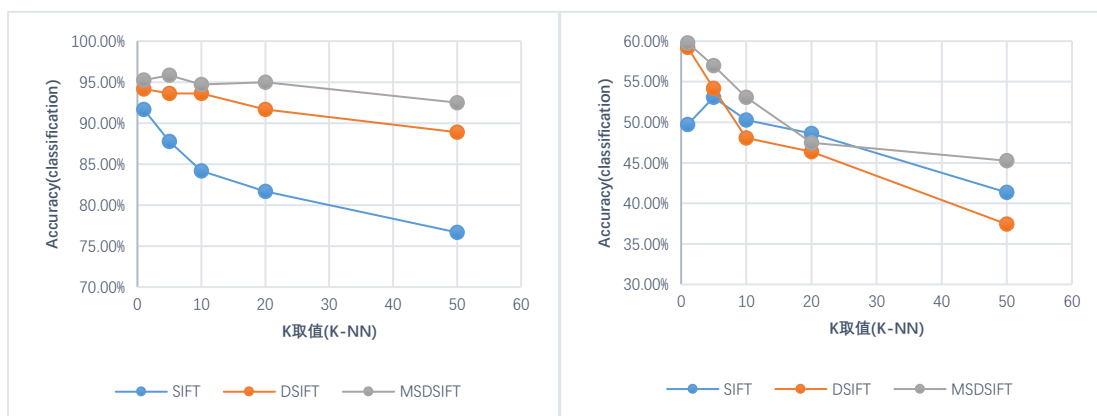
sift(SPARSE SIFT), dsift(DENSE SIFT), msdsift(MULTI-SCALE DENSE SIFT)

改变 K-NN 分类器的 K 值：

1, 5, 10, 20, 50

2.2.2 KNN 分类器准确率-K- SIFT 特征种类

Dataset: 4_ObjectCategories				Dataset: 15_ObjectCategories			
K\SIFT 种类	SIFT	DSIFT	MSDSIFT	K\SIFT 种类	SIFT	DSIFT	MSDSIFT
1	91.67%	94.17%	95.28%	1	49.72%	59.22%	59.78%
5	87.78%	93.61%	95.83%	5	53.07%	54.19%	56.98%
10	84.17%	93.61%	94.72%	10	50.28%	48.04%	53.07%
20	81.67%	91.67%	95.00%	20	48.60%	46.37%	47.49%
50	76.67%	88.89%	92.50%	50	41.34%	37.43%	45.25%



由实验数据可知，对于数据集 4_ObjectCategories 的分类准确率普遍在 75%以上，明显优于数据集 15_ObjectCategories（低于 60%），这是因为数据集 4_ObjectCategories 每一类别中包含的图像更多，且只有四个图像类别，使得训练集/测试集更大，训练更为充分；

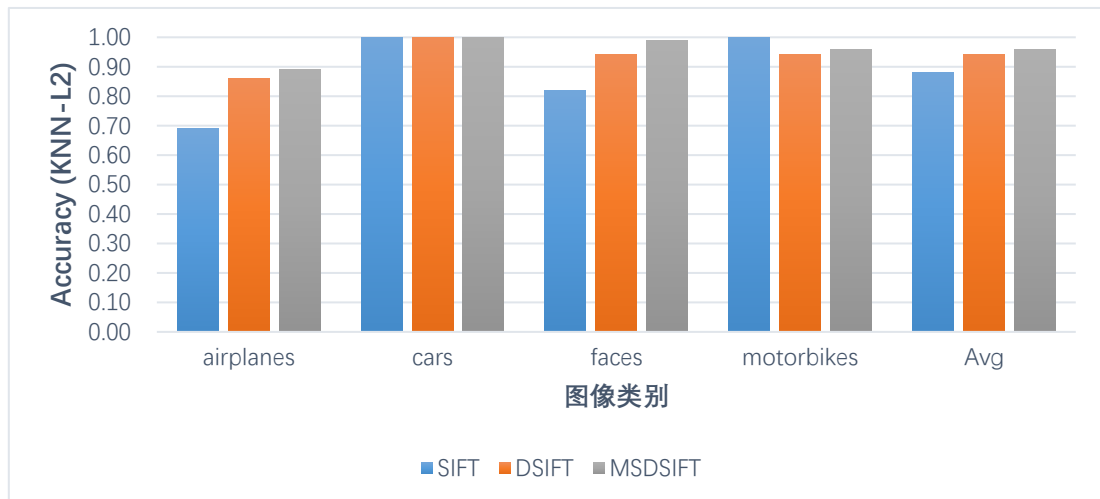
对于两个数据集，三种 SIFT 特征种类对比来看，MSDSIFT > DSIFT > SIFT，其中 MSDSIFT 的效果最佳，对于数据集 4_ObjectCategories 的分类准确率可达 95.83%（K=5）；

KNN 中的 K 值选取对 K 近邻算法的结果会产生重大影响。在本次实验中，对于确定的 SIFT 特征种类，K 取 1 或 5 时，可以达到最佳准确率，之后随着 K 的增大，分类器的准确率会降低。这是因为，当增大 K 的时候，有周围更多的样本可以借鉴，错误率会先降低，分类效果会变好。但当 K 值过大的时候，KNN 失去意义，错误率会更高。对于最佳的临界 K 点，当它继续增大或减小的时候，错误率都会上升。由于本次实验的数据集较小，所以适合的 K 值也较小。

2.2.3 图像分类准确率-SIFT 特征种类

以 4_ObjectCategories 数据集为例：

SIFT 种类\Class	airplanes	cars	faces	motorbikes	Avg
SIFT	0.69	1.00	0.82	1.00	0.88
DSIFT	0.86	1.00	0.94	0.94	0.94
MSDSIFT	0.89	1.00	0.99	0.96	0.96

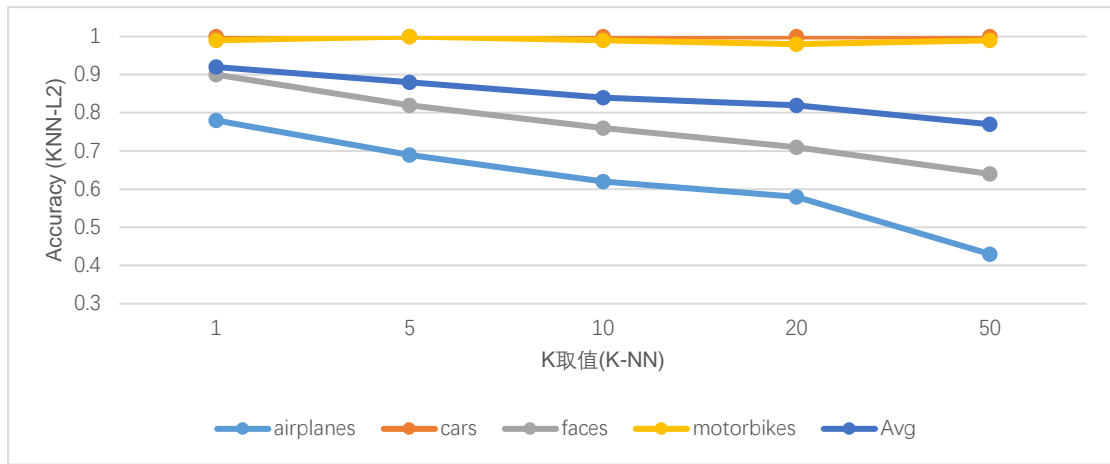


由实验数据可知，MSDSIFT 的效果总体高于其他两种 SIFT 特征种类，但 SPARSE SIFT 对于 motorbikes 的分类效果更好；cars 类别的分类效果较为理想，均为 100%，airplanes 类别的分类效果较差，采用的三种 SIFT 特征种类均未达 90%。

2.2.4 图像分类准确率-KNN 的 K 取值

以 4_ObjectCategories 数据集为例，SIFT 特征种类采用 SPARSE SIFT：

K\Class	airplanes	cars	faces	motorbikes	Avg
1	0.78	1	0.9	0.99	0.92
5	0.69	1	0.82	1	0.88
10	0.62	1	0.76	0.99	0.84
20	0.58	1	0.71	0.98	0.82
50	0.43	1	0.64	0.99	0.77



由实验数据可知，总体上分类准确率随 K 的增加而降低；ars 类和 motorbikes 类的分类准确率一直稳定维持在 98%以上，高于平均水平；airplanes 类的分类效果最不理想，最高仅为 78%。

2.2.5 KNN 自动超参数优化

fitknn 函数提供超参数优化选项：

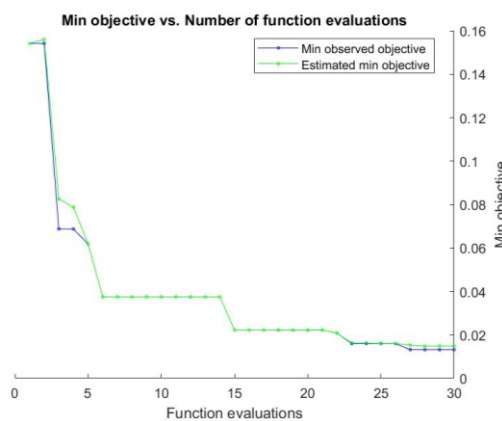
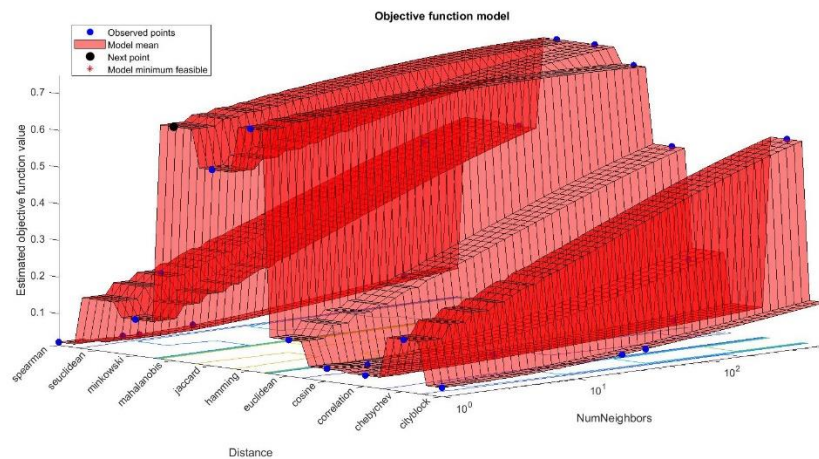
```
kNNClassifier = fitknn(bof_train, labels_train, 'NumNeighbors', K, 'OptimizeHyperparameters', 'auto', ...
    'HyperparameterOptimizationOptions', ...
    struct('AcquisitionFunctionName', 'expected-improvement-plus'));
```

- 'auto'-使用 {'Distance','NumNeighbors'}。
- 为了重现性，需设置随机种子并使用'expected-improvement-plus'采集功能。

自动超参数优化的 30 次迭代结果如下：

Iter	Eval	Objective	Objective	BestSoFar	BestSoFar	NumNeighbors	Distance
	result		runtime	(observed)	(estim.)		
1	Best	0.15417	0.33142	0.15417	0.15417	1	chebychev
2	Accept	0.20069	0.37972	0.15417	0.15602	3	seuclidean
3	Best	0.06875	0.32834	0.06875	0.082608	33	cityblock
4	Accept	0.45972	0.4263	0.06875	0.07881	272	seuclidean
5	Best	0.061806	0.31503	0.061806	0.061859	22	cityblock
6	Best	0.0375	0.30749	0.0375	0.037511	1	cityblock
7	Accept	0.043056	0.3415	0.0375	0.037507	1	correlation
8	Accept	0.079167	0.37148	0.0375	0.037507	199	correlation
9	Accept	0.043056	0.38627	0.0375	0.037505	2	cosine
10	Accept	0.21319	0.39742	0.0375	0.037508	493	cosine
11	Accept	0.11181	0.30816	0.0375	0.037508	1	euclidean
12	Accept	0.5	0.39026	0.0375	0.037512	718	euclidean
13	Accept	0.052778	0.34401	0.0375	0.037508	9	correlation
14	Accept	0.54792	1.4445	0.0375	0.037509	1	jaccard
15	Best	0.022222	0.97194	0.022222	0.022237	1	spearman
16	Accept	0.077083	0.84115	0.022222	0.022236	376	spearman
17	Accept	0.11181	0.36063	0.022222	0.022235	1	minkowski
18	Accept	0.75	9.1351	0.022222	0.022239	716	mahalanobis
19	Accept	0.67361	1.0468	0.022222	0.022243	1	hamming
20	Accept	0.5	0.45837	0.022222	0.022242	720	minkowski

Iter	Eval	Objective	Objective	BestSoFar	BestSoFar	NumNeighbors	Distance
	result		runtime	(observed)	(estim.)		
21	Accept	0.56181	0.44122	0.022222	0.022242	720	chebychev
22	Best	0.020833	0.81433	0.020833	0.020858	10	spearman
23	Best	0.015972	0.92893	0.015972	0.016236	3	spearman
24	Accept	0.047222	0.3743	0.015972	0.016202	1	cosine
25	Accept	0.015972	0.79801	0.015972	0.016097	3	spearman
26	Accept	0.015972	0.83428	0.015972	0.016057	3	spearman
27	Best	0.013194	0.81493	0.013194	0.015311	4	spearman
28	Accept	0.013194	0.92052	0.013194	0.014843	4	spearman
29	Accept	0.75	1.8441	0.013194	0.014891	712	jaccard
30	Accept	0.70764	1.1377	0.013194	0.014926	720	hamming



Best observed feasible point:

NumNeighbors	Distance
4	spearman

Observed objective function value = 0.013194

Estimated objective function value = 0.014926

Function evaluation time = 0.81493

Best estimated feasible point (according to models):

NumNeighbors	Distance
4	spearman

Estimated objective function value = 0.014926

Estimated function evaluation time = 0.86716

由结果可知，最优超参数为{'spearman', 4}

Spearman 秩相关系数是一种无参数检验方法，用于度量变量之间联系的强弱。

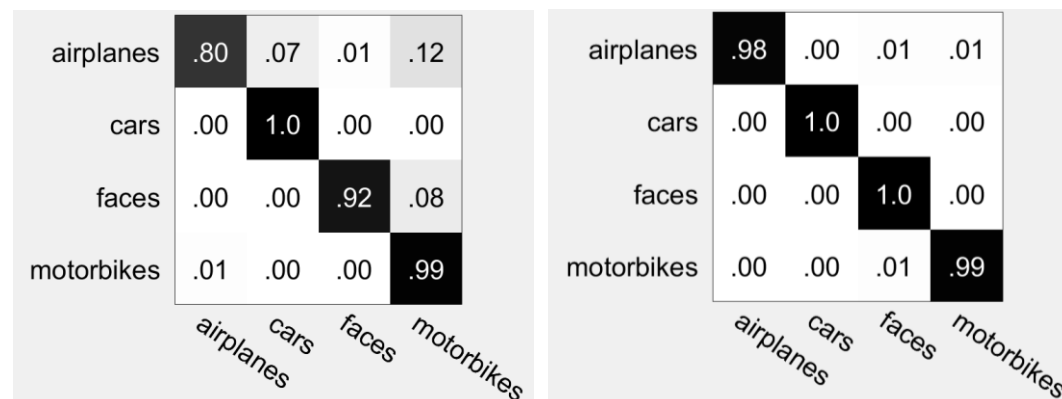
公式如下：

$$\rho = 1 - \frac{6 \sum_{i=1}^N d_i^2}{N(N^2 - 1)}$$

无参数的等级相关系数，亦即其值与两个相关变量的具体值无关，而仅仅与其值之间的大小关系有关。di 表示两个变量分别排序后成对的变量位置差，N 表示 N 个样本，减少异常值的影响。

当 K=4 时，SIFT 特征种类选取 SPARSE SIFT，比较采用 L2 与 Spearman 的不同效果：

- OVERALL KNN L2 classification accuracy: 0.9275
- OVERALL KNN spearman classification accuracy: 0.9917



注：左图采用欧氏距离，右图采用 spearman

由上图可见，KNN 分类器{'spearman', 4}的平均准确率高达 99.17%，明显高于欧氏距离计算下的 92.75%，且准确率的提升效果主要作用于 airplanes 类和 faces 类。

三、总结

通过本次上机实验，我掌握了使用视觉单词进行图像分类的方法。在实验过程中，我使用 SIFT 对每幅图像进行特征采样与描述，使用 K-means 对提取特征聚类，生成 codebook；为每幅图像生成直方图，进行正则化，将最终结果作为图像的特征表示；最后使用 K-NN 分类器对图像特征表示进行分类。

在掌握视觉单词图像分类流程之后，我依据老师的实验要求，在实验提供的两个数据集上，进行了参数调整与测试：测试了不同的 SIFT 特征提取方式测试图像分类准确率；测试了 KNN 分类器不同 K 值对应的分类准确率。记录实验数据，绘制数据表格与图像，分析相应结论。

此外，我还尝试使用了 Matlab 的自动超参数优化功能（贝叶斯优化），为 K-NN 分类器寻找最佳邻居数与距离策略，所得结果符合预期。

在这次实验中，我深刻地了解了词袋模型在 cv 领域的经典应用，对于 nlp 与 cv 领域的交融产生了一些兴趣，受到了较大的启发。