

实验 2-图像检索

胡图图

联系方式: 翻斗大街翻斗花园二号楼 1001 室

(Dated: December 1, 2022)

1 实验目的

数据集包含 1000 张图像, 分为 250 组, 每组 4 张图像, 为同一个物体的不同视角. 如下所示:



实验目的为, 为每张图像, 在 1000 张图像中, 检索出与其相似的图像.

2 实验原理

2.1 局部特征提取

使用 SIFT 算法提取局部图像特征, 具体略.

2.2 聚类得到视觉码本

第 i 副图像得到 n_i 个 sift 特征. 所有图像共有 $N = \sum_i n_i$ 个视觉特征. 对这 N 个特征进行聚类, 得到 M 个聚类中心.

实验使用了层次 k 均值聚类. 例如首先使用 k 均值聚类将所有数据分为 10 类, 对于每个种类, 重复使用 k 均值聚类, 再将其聚为 10 类...

使用码本得到图像全局特征

在上一步中, 聚类得到 N 个聚类中心. 用一个 N 维的向量 v 表征一副图像的特征. 设第 i 副图像有 n_i 个 sift 特征, 计算每个 sift 特征和 N 个聚类中心的距离, 找到距离最近的聚类中心. 例如此特征离第 j 个中心最近, 则将 v 的第 j 个位置加 1(实际上不是直接计算和聚类中心的距离得到最近的特征, 而是按照树的结构逐层搜索查找中心). 之后, 使用某种方式 (L_1 范数或 L_2 范数) 将 v 归一化.

最后, 对于每张图像, 得到一个 N 维向量, 其第 j 个位置的分量表示第 j 个视觉单词出现的次数.

2.3 使用倒排表计算两个图像向量的相似度

使用下式, 计算两个图像特征的 p 范数距离.

$$\begin{aligned}\|\mathbf{q} - \mathbf{d}\|_p^p &= \sum_i |q_i - d_i|^p \\ &= \sum_{i|d_i=0} |q_i|^p + \sum_{i|q_i=0} |d_i|^p + \sum_{i|q_i \neq 0, d_i \neq 0} |q_i - d_i|^p \\ &= \|\mathbf{q}\|_p^p + \|\mathbf{d}\|_p^p + \sum_{i|q_i \neq 0, d_i \neq 0} (|q_i - d_i|^p - |q_i|^p - |d_i|^p) \\ &= 2 + \sum_{i|q_i \neq 0, d_i \neq 0} (|q_i - d_i|^p - |q_i|^p - |d_i|^p),\end{aligned}$$

对于一个图像, 用其图像特征, 计算和所有图像特征的距离, 找到距离最小的图像, 作为检索的结果.

3 实验实现

3.1 sift 特征提取

因为不清楚生成数据的数据结构, 并没有使用老师提供的程序以提取 sift 特征. 作为替代, 使用了 sklearn 库中的实现.

```
import cv2
```

```
sift = cv2.SIFT_create()
kp, des = sift.detectAndCompute(img, None)
```

实验中对特征进行标准化, 调整为均值为 0, 方差为 1.

3.2 层次聚类

尝试两种, 聚类个数 width = 10, 聚类深度 depth = 4; width = 10, 聚类深度 depth = 5. 两者分别有 $10^{**4}, 10^{**5}$ 个视觉单词.

每次聚类时, 使用分到此分支下所有的数据. 如果特征的个数较多, 从中采样出 10^{**4} 个数据进行聚类. 其他情况下, 不对数据进行采样以增加或减少数据.

3.3 检索指标

我们检索除去本身外, 与其最为相似的 3 张图像. 因为每个图像有 3 张不同视角的图像, 此时准确率和召回率相等, 进而得到 F 值和它们相等 (实验文档要求计算图像和包括自己前 4 副图像的准确率, 我们认为图像和自身的距离为 0, 必然被检索为最相似的图像, 因此计算除去本身外的 3 张图像).

我们汇报 F@3 和 mAP 值.

4 实验结果

4.1 实验结果

实验得到的最好结果为 F@3: 0.8150, mAP: 0.8542. 在深度为 5, 使用 L1 范数归一化, 使用 L1 范数计算距离时得到.

使用不同的归一化方式, 距离范数计算方式, 码本大小, 得到的准确率如下所示:

Table 1: 准确率

特征归一化-距离范数-深度	F-score@3	mAP
L1-L1-4	0.7467	0.7926
L1-L1-5	0.8150	0.8542
L2-L1-4	0.4653	0.5115
L2-L1-5	0.7403	0.7797
L2-L2-4	0.6050	0.6584
L2-L2-5	0.7317	0.7786
L2-L1-4	-	-
L2-L1-5	0.7403	0.7797

我们可以得到几个结论: 使用 L1 范数进行归一化效果更好; 如论文 Scalable Recognition with a Vocabulary Tree 中所说, 使用 L1 范数计算距离效果更好; 本实验中, 使用 10^{**5} 个视觉单词可以得到更高的准确率 (平均每个视觉单词在所有数据中出现 12 次).

随机检索 5 副图像, 其结果如下所示, 其中第一幅为检索图像, 剩下依次为与其相似的 3 张图像.







4.2 sift 特征

实验发现 sift 特征分布非常均匀. 共有 1204237 个 sift 特征, 将所有特征划分为 $10^{**}5$ 个特征 (视觉单词), 最后每个视觉单词出现的次数都不为 0, 出现最频繁的视觉单词出现了 199 次, 出现次数超过 40 的视觉单词有 669 个, 平均每个视觉单词出现了 12.04237 次. 可以看到 sift 特征分布十分均匀, 十分适合搭配聚类算法.

4.3 检索时间

测试深度为 4, 5(视觉单词为 $10^{**}4$, $10^{**}5$) 时的运行时间.

Table 2: 运行时间

内容-深度	时间 (单位 s)
图像特征-4	113.56 / 1000
图像特征-5	133.22 / 1000
检索-4	197.48 / 1000
检索-5	64.28 / 1000

检索分为 2 步:

根据图像的 sift 特征计算图像特征, 此处时间应与深度成正比. 可以看到随着深度增加, 时间增长较为缓慢.

根据图像特征, 使用倒排表计算相似度 (距离). 可以看到随着深度增加, 检索时间反而减少. 这是因为每个图像的 sift 特征数量不变, 随着深度增加, 图像特征的非 0 分量数量变化不大. 而整体来看, 倒排表每个视觉特征的链表 (列表) 长度缩小 10 倍, 即遍历此链表的时间缩小 10 倍.

5 实验总结

本文的内容可以看作论文 Scalable Recognition with a Vocabulary Tree 的子集 (例如并没有实现论文中赋予不同视觉单词不同的权重). 具体来说, 提取 sift 特征, 使用层次 k 均值聚类得到视觉单词, 进而使用倒排表算法检索相似图像. 最好结果为 F@3: 0.8150, mAP: 0.8542.