《计算机视觉》实验报告

姓名：孔馨怡 学号：22122128

实验 8 图像检索

一. 采用 SIFT 特征实现图像检索功能，即输入一张图片，在数据集中检索出相似的图片，数据集自选。

(1) 基于词袋模型实现

(2) 检索结果按照相似度进行排序

a) 核心代码：

```
filelist = get_imlist('data/')

# 输入的图片
im1f = '488.jpg'
im1 = array(Image.open(im1f))
sift.process_image(im1f, 'out_sift_1.txt')
l1, d1 = sift.read_features_from_file('out_sift_1.txt')

i = 0
num = [0] * len(filelist)  # 存放匹配值
for infile in filelist:  # 对文件夹下的每张图片进行如下操作
    im2 = array(Image.open(infile))
    sift.process_image(infile, 'out_sift_2.txt')
    l2, d2 = sift.read_features_from_file('out_sift_2.txt')
    matches = sift.match_twosided(d1, d2)
    num[i] = len(matches.nonzero()[0])
    i = i + 1
    print('{} matches'.format(num[i - 1]))  # 输出匹配值

i = 1
figure()
while i < 4:  # 循环三次，输出匹配最多的三张图片
    index = num.index(max(num))
    print(index, filelist[index])
    lena = mpimg.imread(filelist[index])  # 读取当前匹配最大值的图片
```

```
    # 此时 lena 就已经是一个 np.array 了，可以对它进行任意处理
    # lena.shape  # (512, 512, 3)
    subplot(1, 3, i)
    plt.imshow(lena)  # 显示图片
    plt.axis('off')  # 不显示坐标轴
    num[index] = 0  # 将当前最大值清零
    i = i + 1
show()
```

词袋口优化：

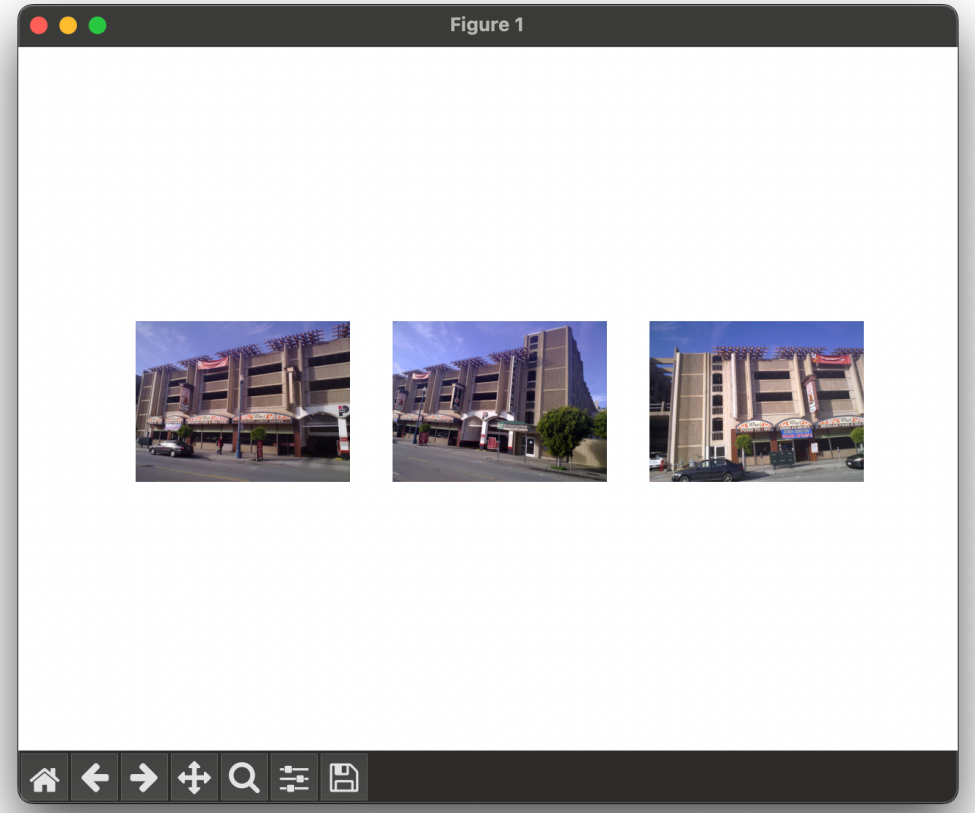
a) 核心代码：

```
# 构建视觉词典
def build_visual_vocabulary(features, num_clusters):
    kmeans = KMeans(n_clusters=num_clusters)
    kmeans.fit(features)
    visual_vocabulary = kmeans.cluster_centers_
    return visual_vocabulary

# 计算图像的词袋表示
def compute_bovw_representation(image_descriptors, visual_vocabulary):
    bovw_representation = np.zeros(len(visual_vocabulary))
    nearest_neighbor = NearestNeighbors(n_neighbors=1)
    nearest_neighbor.fit(visual_vocabulary)
    distances, indices = nearest_neighbor.kneighbors(image_descriptors)
    for index in indices:
        bovw_representation[index] += 1
    return bovw_representation

# 计算两个词袋表示之间的相似度
def compute_similarity(bovw1, bovw2):
    similarity = np.dot(bovw1, bovw2) / (np.linalg.norm(bovw1) *
np.linalg.norm(bovw2))
    return similarity
```

TIPS：

测试图片集/训练的图片集：地标建筑图片集，约 500 张选取 https://purl.stanford.edu/rb470rw0983 网站的 lansmarks 中的 Query 为数据集。

b) 实验结果截图



检索结果按照相似度进行排序

```
processed tmp.pgm to out_sift_2.txt
0 matches
processed tmp.pgm to out_sift_2.txt
3 matches
processed tmp.pgm to out_sift_2.txt
0 matches
processed tmp.pgm to out_sift_2.txt
2 matches
processed tmp.pgm to out_sift_2.txt
2 matches
processed tmp.pgm to out_sift_2.txt
0 matches
processed tmp.pgm to out_sift_2.txt
1 matches
```

终端输出结果

c) 实验小结

经历了前几次的实验，基本上实验 8 图像检索也是同样一个道理。在计算机视觉的实验过程中，往往是先确认实验的一个背景目的，选取相应的一个模型，找合适的数据集，进行划分之后开始训练，最后得出结果。无论是上一次的人脸识别还是人脸检测还是这个图像检索，需要在整个实验的过程中去感受解决问题的一个过程，并在不同的实验中逐渐熟悉各个库和算法的实际使用。

在这之中不仅收获了对于计算机视觉各个方面的应用的经验，还提升了自己对于信息的检索能力。

下面是一些实验中踩的坑和碎碎念？：

在使用 PCV 库的时候，先是经历了下载后 py2 要适应 py3 等的操作，后来各种配置弄好以后 sift.py 文件里一直识别不了，不会报错，但是就是不认识这个文件。弄半天才知道在 sift 文件里的可执行文件要自己去下载和改路径。总之弄了很久才开始正式跑（）

```
cmmd = str(r"/Users/kongxinyi/Desktop/计算机/专业课
/CV/SIFT/vlfeat-0.9.20/bin/maci64/sift "+imagename+" --output="+resultname+
          " "+params)
os.system(cmmd)
```