

《计算机视觉》实验报告

姓名：王子棠 学号：21122897

实验 8

一. 任务 1

a) 核心代码：

i. 单张图片 SIFT 特征提取

读取图像

```
image_path = "data/000122.jpg"
```

```
image = cv2.imread(image_path)
```

```
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

创建 SIFT 对象并提取特征

```
sift = cv2.xfeatures2d.SIFT_create()
```

```
keypoints, descriptors = sift.detectAndCompute(gray_image, None)
```

在图像上绘制特征点

```
image_with_keypoints = cv2.drawKeypoints(gray_image, keypoints, image,  
color=(255,0,255))
```

显示图像和特征点

```
plt.figure(figsize=(10, 8))
```

```
plt.imshow(cv2.cvtColor(image_with_keypoints, cv2.COLOR_BGR2RGB))
```

```
plt.title('Image with SIFT Keypoints')
```

```
plt.axis('off')
```

```
plt.show()
```

ii. 数据集中检索相似图片

提取 SIFT 特征函数

```

def extract_sift_features(image_path):
    img = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
    sift = cv2.xfeatures2d.SIFT_create()
    keypoints, descriptors = sift.detectAndCompute(img, None)
    return keypoints, descriptors

# 数据集中检索图片，从相似度列表中选择前 top5 个图像作为相似图像结果
def image_retrieval(input_image, dataset_folder, top_n=5):
    input_keypoints, input_features = extract_sift_features(input_image)
    image_paths = os.listdir(dataset_folder)
    similarity_scores = []

    # 对于数据集中每张图片都提取 SIFT 特征
    for image_path in image_paths:
        full_path = os.path.join(dataset_folder, image_path)
        image_keypoints, image_features = extract_sift_features(full_path)
        if image_features is not None and input_features is not None:
            # 计算输入图像特征与数据集中所有图像特征的余弦相似度
            similarity = cosine_similarity(input_features, image_features)
            if not np.isnan(similarity[0][0]):
                similarity_scores.append((full_path, similarity[0][0]))

    # 根据相似度进行降序排序
    similarity_scores.sort(key=lambda x: x[1], reverse=True)
    similar_images = [path for path, _ in similarity_scores[:top_n]]
    return similar_images

# 显示相似图像
def plot_images(input_image, similar_images):
    plt.figure(figsize=(10, 5))
    plt.subplot(1, len(similar_images) + 1, 1)
    plt.imshow(cv2.imread(input_image))
    plt.title("Input Image")
    plt.axis('off')

```

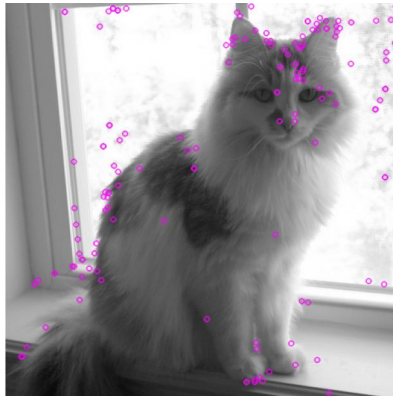
```

for i, image_path in enumerate(similar_images):
    plt.subplot(1, len(similar_images) + 1, i + 2)
    plt.imshow(cv2.cvtColor(cv2.imread(image_path),
cv2.COLOR_BGR2RGB))
    plt.title(f"Similar Image {i + 1}")
    plt.axis('off')
plt.tight_layout()
plt.show()
# 检索图像实际例子
input_image = 'data/13.png'
dataset_folder = 'data/' # 数据集路径
similar_images = image_retrieval(input_image, dataset_folder)
print(similar_images)
plot_images(input_image, similar_images)

```

b) 实验结果截图

- 单张图像 SIFT 特征提取



- 图像检索



- 检索到的相似图片的路径（检索结果按照相似度降序排序）：['data/000122.jpg', 'data/12.png', 'data/1.png', 'data/000209.jpg', 'data/000039.jpg']

c) 实验小结

本次实验的数据集共有 30 张图像，包含游戏截图、人物图像、动物图像和车辆图像等多种类型。图像检索的功能是通过计算输入图像和数据集中所有图像的特征的相似度来实现的，并把所有的相似度存储在列表中，并排序，从而实现了在数据集中检索相似图片且检索结果按照相似度排序。