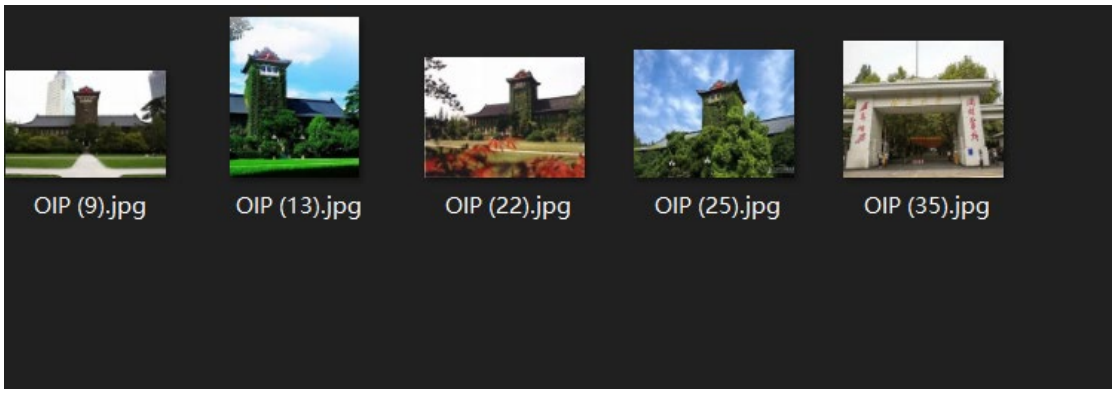
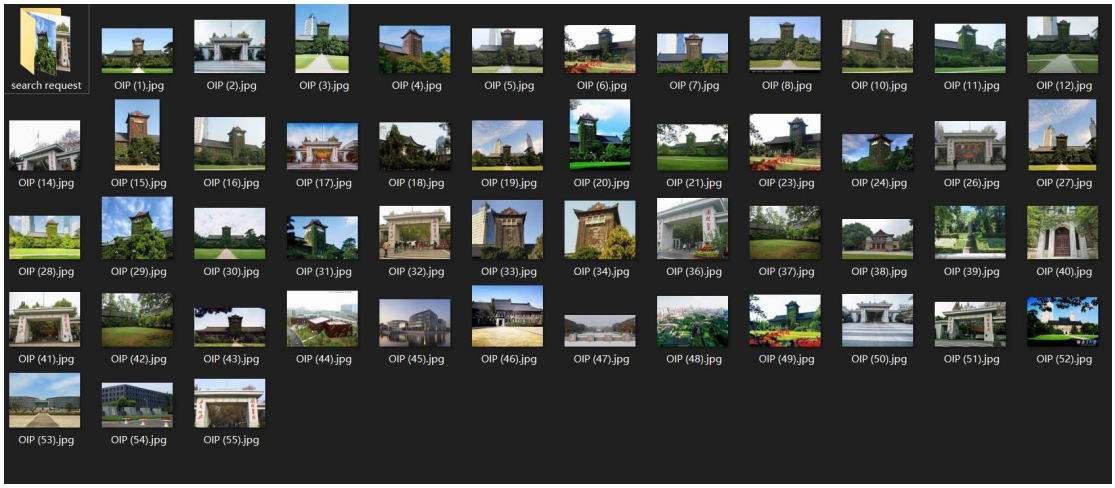
《计算机视觉》（本科，2023）作业 3

1. 在搜索引擎上按照某一关键词，搜索 55 张不同的图像，从中选出 5 张作为检索请求，另 50 张作为被检索图像。

搜索引擎：bing

关键词：南京大学





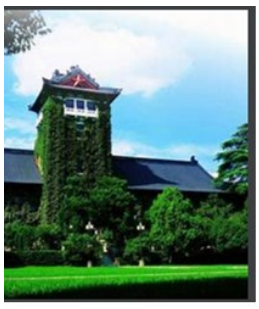
2. 以全局 RGB 颜色直方图（每通道 bin 的数量为 8）作为特征，进行图像检索。展示每个检索请求及对应的前 3 个结果。

```python
def calculate_histogram(image, bins):
    # 将图像转换为RGB颜色空间
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    # 计算颜色直方图
    hist = cv2.calcHist([image], [0, 1, 2], None, bins, [0, 256, 0, 256, 0, 256])
    # 归一化直方图
    hist = cv2.normalize(hist, hist).flatten()
    return hist
```

```python
def image_search(query_image, dataset_folder, num_results):
    # 加载检索请求图像
    query = cv2.imread(query_image)
    # 计算检索请求图像的颜色直方图
    query_hist = calculate_histogram(query, [8, 8, 8])

    results = []
    # 遍历数据集中的图像
    for filename in os.listdir(dataset_folder):
        # 加载图像
        image_path = os.path.join(dataset_folder, filename)
        image = cv2.imread(image_path)
        # 计算当前图像的颜色直方图
        image_hist = calculate_histogram(image, [8, 8, 8])
        # 计算与检索请求图像的直方图的巴氏距离
        distance = cv2.compareHist(query_hist, image_hist, cv2.HISTCMP_BHATTACHARYYA)
        # 将图像路径和对应的距离添加到结果列表中
        results.append((image_path, distance))

    # 根据距离进行排序
    results = sorted(results, key=lambda x: x[1])

    # 返回前num_results个结果
    return results[:num_results]
```

检索请求 1：

结果：



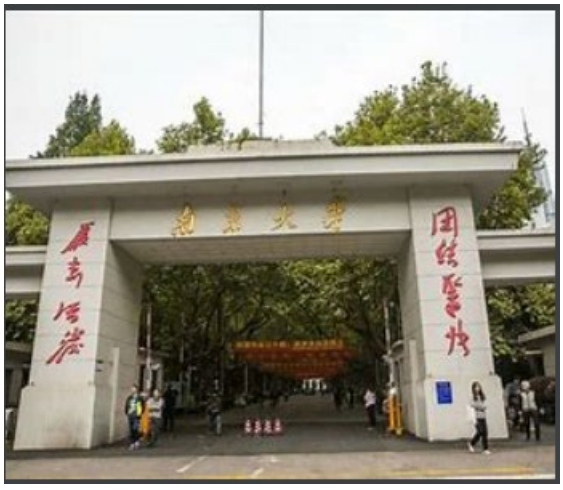

检索请求 2：

结果：

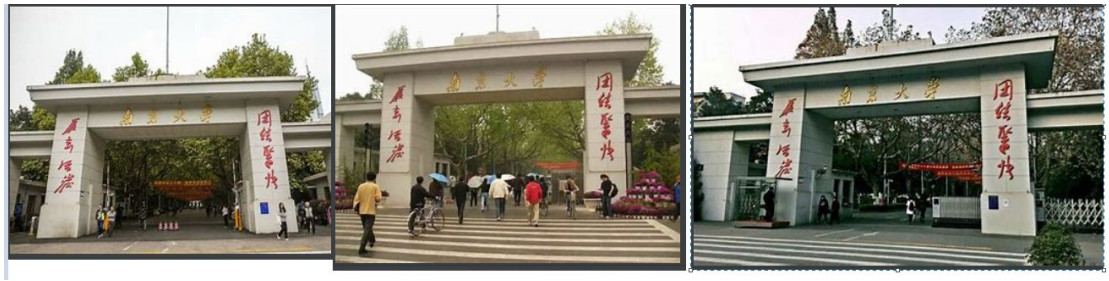
检索请求 3：

结果：

检索请求 4：



结果：



检索请求 5：

结果：

3. 选择 1-2 种其它特征，重复问题 2。

```python
def calculate_shape_features(image):
    # 将图像转换为灰度图像
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    # 对图像进行阈值处理
    _, thresh = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)
    # 查找轮廓
    contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    # 计算图像的形状特征（轮廓面积和周长）
    area = cv2.contourArea(contours[0])
    perimeter = cv2.arcLength(contours[0], True)
    # 将形状特征组合为特征向量
    features = np.array([area, perimeter])
    return features
```

```python
def image_search(query_image, dataset_folder, num_results):
    # 加载检索请求图像
    query = cv2.imread(query_image)
    # 计算检索请求图像的形状特征
    query_features = calculate_shape_features(query)

    results = []
    # 遍历数据集中的图像
    for filename in os.listdir(dataset_folder):
        # 加载图像
        image_path = os.path.join(dataset_folder, filename)
        image = cv2.imread(image_path)
        # 计算当前图像的形状特征
        image_features = calculate_shape_features(image)
        # 计算与检索请求图像的特征之间的欧氏距离
        distance = np.linalg.norm(query_features - image_features)
        # 将图像路径和对应的距离添加到结果列表中
        results.append((image_path, distance))

    # 根据距离进行排序
    results = sorted(results, key=lambda x: x[1])

    # 返回前num_results个结果
    return results[:num_results]
```

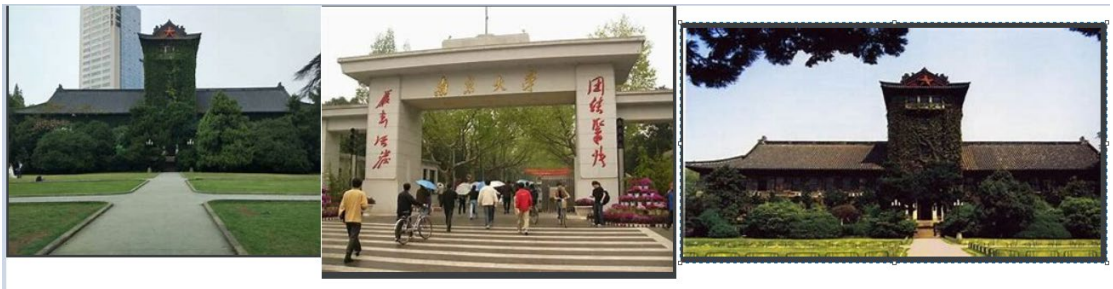
选取特征：形状

检索请求 1：

结果：

检索请求 2：

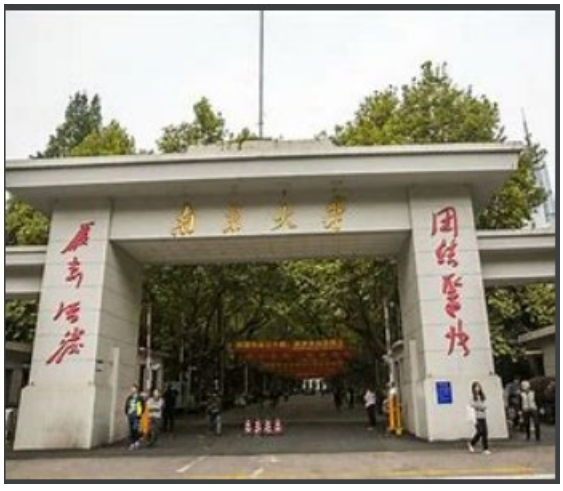
结果：



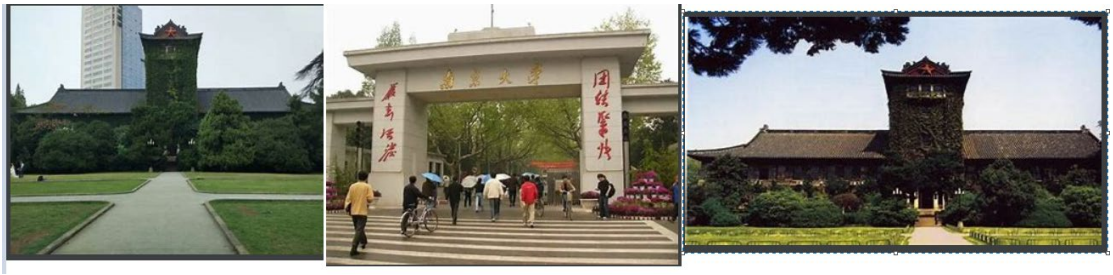
检索请求 3：

结果：

检索请求 4：



结果：



检索请求 5：

结果：

4. 将问题 2 和问题 3 的结果进行比较和分析。

对于问题 2 而言，对于颜色直方图的提取较为合理，搜索到的结果与原图的颜色情况大多相似。我在被检索图像中加入了一些与对应检索请求十分相似的图片，也都被检索出来了。但是对于问题 3，结果就不太合理。我使用的是形状特征作为搜索特征。具体来说，是针对图像中物体的外观和轮廓，通过提取物体的边缘、轮廓或使用形状描述符（如尺度不变特征变换）来表示图像的形状特征。但是这个效果似乎不好，可以看到对于检索请求 4，检索的结果与原图似乎无法正确对上，与检索请求 4 极为相似的图片（例如检索请求 3 的结果 3）并没有被搜索。这应该与我的特侦提取代码有关。

同时，上述结果让我想到课上老师将 SITF 的内容，即在神经网络出现之前，图像检索十分困难，因为单一的特征例如颜色直方图无法直接代表图片。