第十周报告

1. **实验概览**

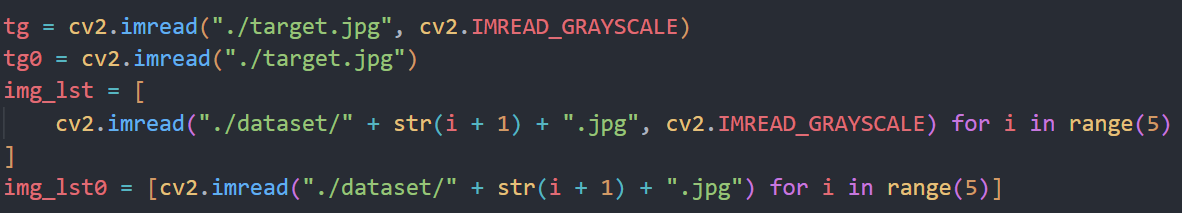
本实验旨在理解并实现SIFT算法，理解SIFT特征子具有不变性、独特性等诸多良好特征的原因，学习SIFT算法的主要步骤和具体原理。在实验中自行实现SIFT算法并和opencv提供的算法进行比较。

1. **实验环境**

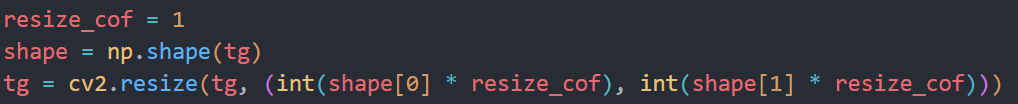
Docker: sjtumic/ee208

1. **解决思路**
   1. **图像预处理**

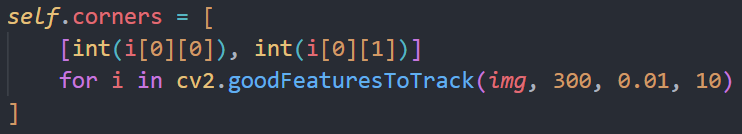
* 灰度图读取：利用opencv自带cv2.IMREAD\_GRAYSCALE



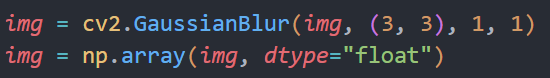
* Resize目标图像：基于给定的参数对目标图像进行缩放，缩放参数将进行实验获得



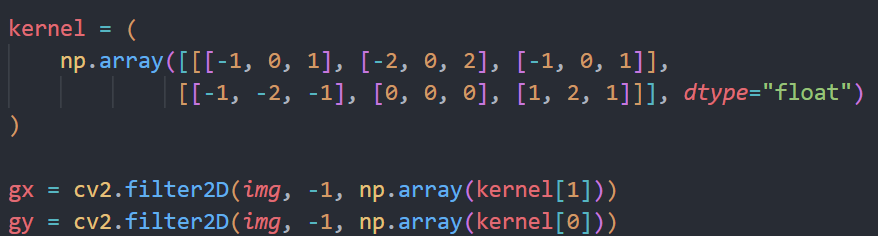
* 基于Harris角点提取获得特征点



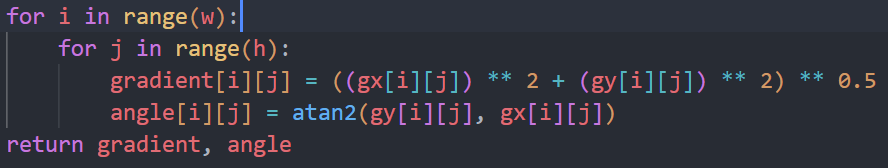
* 高斯滤波



* 1. **获取梯度和特征点方向**
* 基于索贝尔核对图像进行卷积梯度提取

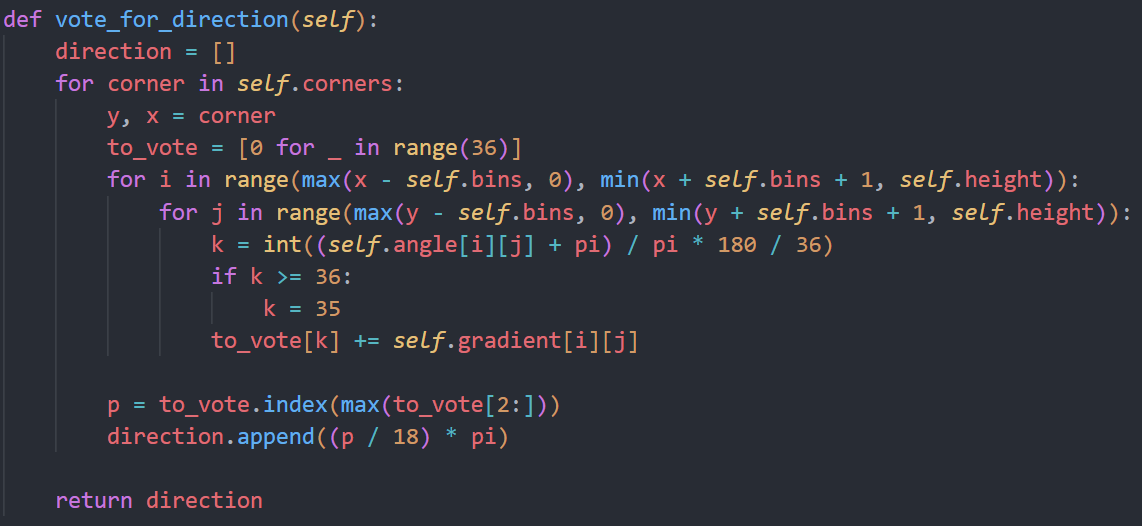


* 获得各点的具体梯度值并计算得到该点的梯度方向



* 1. **投票得到关键点主方向**

对图像内角点周围各个像素点进行遍历，计算所属bin值，然后gradient 为权重为其所在的bin投票；将票数最高的bin来作为该关键点的主方向（代码中总共分为36个方向）

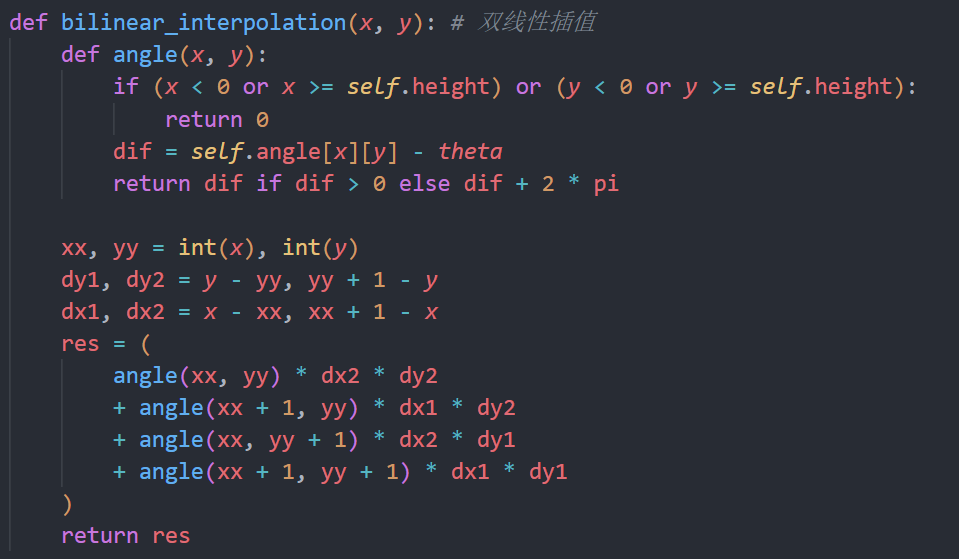
****

* 1. **计算获得特征值**
* 分块计算特征

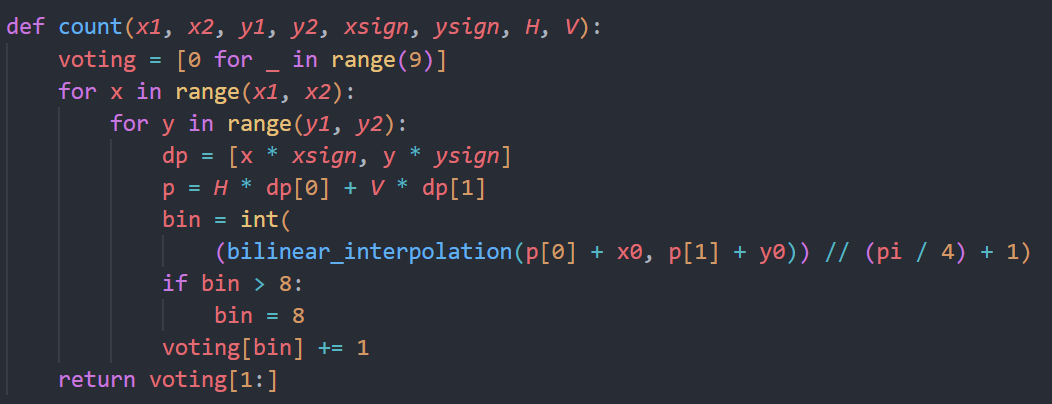
物体坐标系16\*16的邻域分成4\*4个块，每个块4\*4个像素。

在每个块内按照求主方向的方式把360度分成8个bins，统计梯度方向直方图，最终每个块可生成8维的直方图向量，每个关键点可生成4\*4\*8=128维的SIFT描述子。

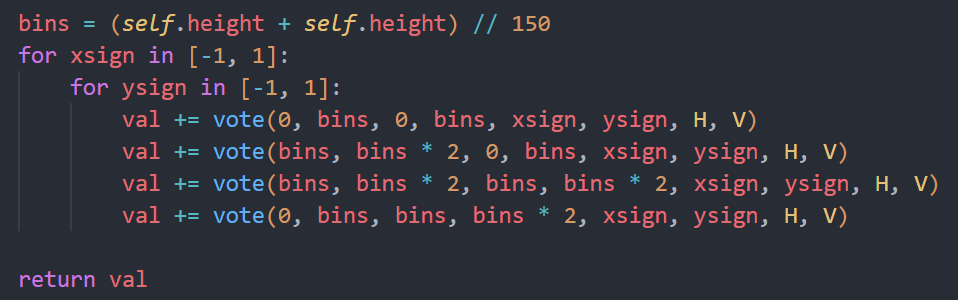
* + 双线性插值



* + 对每个bin投票方式



* 组合获得特征



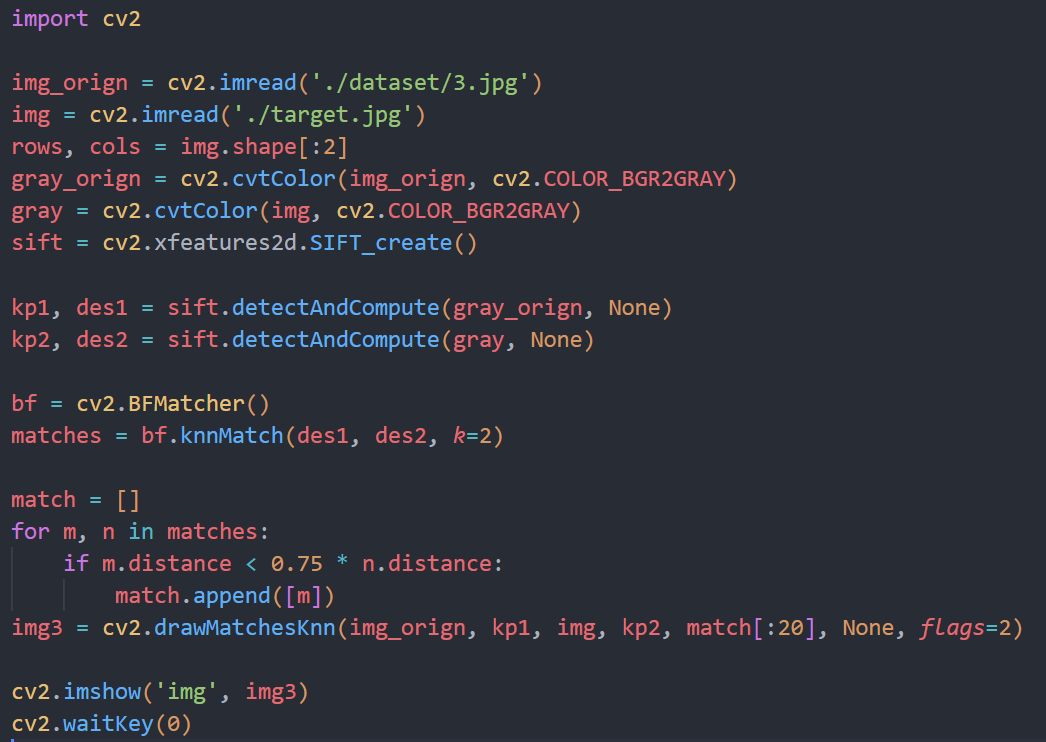
* 1. **图像匹配**

对两张图片的每个关键点的特征值进行匹配，若匹配值大于threshold则记录，若匹配的关键点大于10个则认为匹配成功



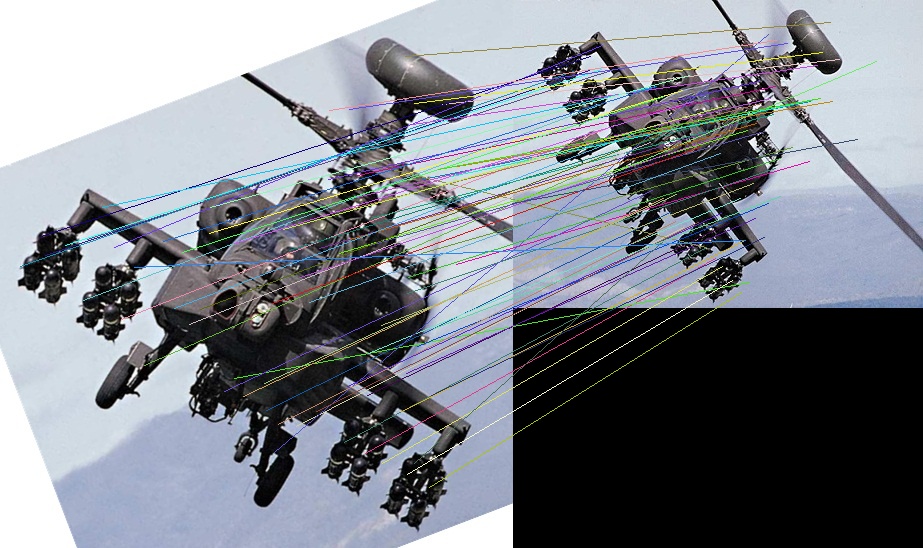
* 1. **对比原版库**

运行原版库代码

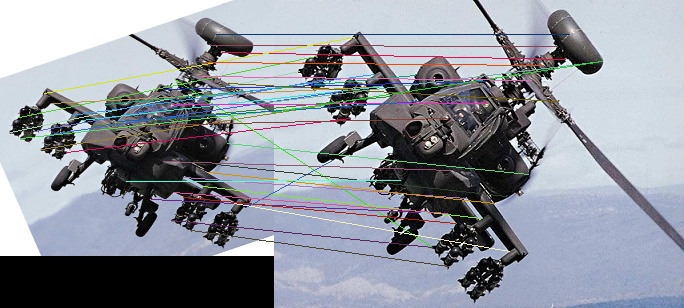


1. **代码运行结果**

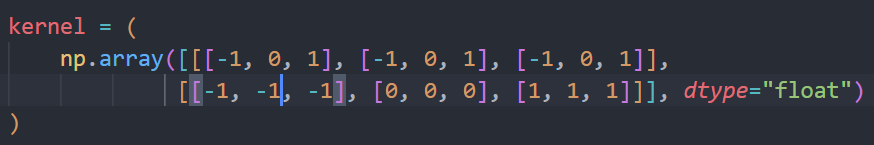
resize\_cof=1, 当threshold设置为0.8时可以得到正确匹配结果：



resize\_cof=0.5, 当threshold设置为0.75时可以得到正确匹配结果：

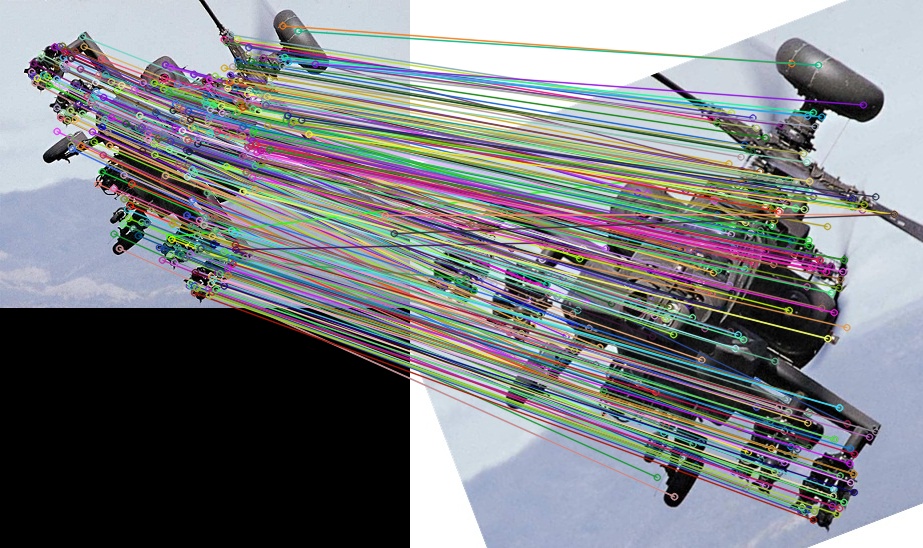


resize\_cof=1, 使用其他卷积核结果：





原版匹配结果：



1. **分析与思考**
   1. **Resize\_cof:** 图片缩小时，匹配成功的点会相应增加，但对于关键点的匹配容易增加误匹配的可能性
   2. **算子比较：**Sobel算子和Prewitt算子匹配效果均可，整体而言Sobel算子的匹配点较多，两者误判水平近似
2. **代码结构解释**

sift.py：自行搭建Sift算法

cv\_sift.py：调用opencv的Sift算法

match.jpg：调用sift.py得到的匹配结果

cv\_match.jpg：调用cv\_sift.py得到的匹配结果