**实现说明：**

该实验的实现主要为几个步骤：

1.读取图像：使用cv2.imread函数读取两张图像，分别是原始图像lena.png和旋转45度后的图像lena\_rot45.png。

2.定义SIFT算子：通过cv2.SIFT\_create()创建一个SIFT算子对象。

3.特征点获取：对读取的两张图像使用SIFT算子进行特征点检测和描述，得到关键点(kp1, kp2)和描述子(des1, des2)。

4.可视化特征点：使用cv2.drawKeypoints函数在原图上绘制出特征点，并将两幅图像的特征点可视化后横向拼接，保存并显示。

5.图像匹配：使用暴力匹配器cv2.BFMatcher和K近邻算法(k=2)进行特征点匹配，得到匹配点集合。然后通过设定一个比值ratio来筛选出最佳匹配点，形成good1列表。

6.特征点匹配结果可视化：使用cv2.drawMatchesKnn函数将匹配的特征点对绘制出来，保存并显示。

7.图像对齐：如果匹配的特征点对数量大于4，则使用RANSAC算法计算单应性矩阵H，并通过cv2.warpPerspective函数将旋转图像变换到与原图对齐的状态，保存并显示变换后的图像

**结果截图：**

输入图片：



拼接完成：



**运行说明：**

以前为了用sift，都是用的旧版本：opencv-contib-python=3.4.2.17，现在sift专利过期了，新版的opencv直接可以使用sift算法，使用从阿里云镜像下载的opencv-python==4.5.1.48版本测试可以使用。与旧版本相比没有cv2.xfeatures2d.SIFT\_create()了，新定义是cv2.SIFT\_create()函数