Nguyễn Thảo Nhi -2151010266

SIFT MATCHING

```
import cv2
import numpy as np

# Load images
img1 = cv2.imread('small.jpg')
img2 = cv2.imread('big.jpg')

# Convert images to grayscale
gray1 = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
gray2 = cv2.cvtColor(img2, cv2.COLOR_BGR2GRAY)

# Initialize SIFT detector
sift = cv2.SIFT_create()
sift.setContrastThreshold(0.03)
sift.setEdgeThreshold(5)

# Detect keypoints and compute descriptors
keypoints1, descriptors1 = sift.detectAndCompute(gray1, None)
keypoints2, descriptors2 = sift.detectAndCompute(gray2, None)
```

```
# Initialize Brute-Force Matcher
bf = cv2.BFMatcher(cv2.NORM_L1, crossCheck=False)

# Match keypoints
matches = bf.match(descriptors1, descriptors2)
matches = sorted(matches, key=lambda x: x.distance)

# Draw matches before ratio test
img3 = cv2.drawMatches(img1, keypoints1, img2, keypoints2,
matches[:50], None, flags=2)

# Apply ratio test
matches = bf.knnMatch(descriptors1, descriptors2, k=2)
good_matches = []
for m, n in matches:
    if m.distance < 0.7 * n.distance:
        good_matches.append(m)

# Draw matches after ratio test
img4 = cv2.drawMatchesKnn(img1, keypoints1, img2, keypoints2,
[good_matches], None,
                          matchColor=(255, 255, 0) ,
matchesMask=None,
```

```
                        singlePointColor=(255, 255, 0) ,
flags=0)

# Combine images horizontally
combined_img = np.hstack((img3, img4))

# Resize combined image to fit window
max_width = 3000  # Set the maximum width for display
scale = max_width / combined_img.shape[1]
combined_img_resized = cv2.resize(combined_img, None, fx=scale
fy=scale)

# Display the result
cv2.imshow('SIFT', combined_img_resized)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

RESULT :