import cv2

import matplotlib.pyplot as plt

# Load images

img1 = cv2.imread('image1.jpg', cv2.IMREAD\_GRAYSCALE) # Query image

img2 = cv2.imread('image2.jpg', cv2.IMREAD\_GRAYSCALE) # Train image

# Check if images are loaded

if img1 is None:

print("Error: Unable to load image1.jpg")

exit()

if img2 is None:

print("Error: Unable to load image2.jpg")

exit()

# Initialize ORB detector

orb = cv2.ORB\_create()

# Find the keypoints and descriptors with ORB

kp1, des1 = orb.detectAndCompute(img1, None)

kp2, des2 = orb.detectAndCompute(img2, None)

# Create a BFMatcher object with distance measurement cv2.NORM\_HAMMING

bf = cv2.BFMatcher(cv2.NORM\_HAMMING, crossCheck=True)

# Match descriptors

matches = bf.match(des1, des2)

# Sort matches by distance (best matches first)

matches = sorted(matches, key=lambda x: x.distance)

# Draw top 10 matches

img\_matches = cv2.drawMatches(img1, kp1, img2, kp2, matches[:10], None, flags=cv2.DrawMatchesFlags\_NOT\_DRAW\_SINGLE\_POINTS)

# Display the matches

plt.figure(figsize=(10, 5))

plt.imshow(img\_matches)

plt.title('Feature Matching using ORB')

plt.axis('off')

plt.show()