第二次作业

一、主要功能

指纹验证识别

二、代码实现

def sift\_match(img\_A, img\_B):

    sift = cv.SIFT\_create()

    kp1, des1 = sift.detectAndCompute(img\_A, None)

    kp2, des2 = sift.detectAndCompute(img\_B, None)

    bf = cv.BFMatcher()

    matches = bf.knnMatch(des1, des2, k=2)

    good\_matches = []

    for m, n in matches:

        if m.distance < 0.7 \* n.distance:

            good\_matches.append(m)

    if len(good\_matches) > 500:

        return True

    else:

        return False

def main\_1():

    img\_template = cv.imread('./0417dataset/DB1\_B/101\_7.tif')

    img\_A = cv.imread('./0417dataset/DB1\_B/101\_6.tif')

    img\_B = cv.imread('./0417dataset/DB1\_B/103\_7.tif')

    if sift\_match(img\_template, img\_A):

        print("source\_A Successful authentication!")

    else:

        print("source\_A Failed authentication!")

    if sift\_match(img\_template, img\_B):

        print("source\_B Successful authentication!")

    else:

        print("source\_B Failed authentication!")

    plt.subplot(131)

    plt.imshow(img\_template)

    plt.title('Template')

    plt.axis('off')

    plt.subplot(132)

    plt.imshow(img\_A)

    plt.title('Source\_A')

    plt.axis('off')

    plt.subplot(133)

    plt.imshow(img\_B)

    plt.title('Source\_B')

    plt.axis('off')

main\_1()

三、运行结果



source\_A Failed authentication!

source\_B Failed authentication!