

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
Інститут комп'ютерних наук та інформаційних технологій
Кафедра систем штучного інтелекту



Звіт до лабораторної №2
з дисципліни
“Обробка зображень методами штучного інтелекту”

Виконав:
ст. гр. КН-410
Жигайло Ярослав

Викладач:
Пелешко Д. Д.

Лабораторна робота №2

Суміщення зображень на основі використання дескрипторів

Варіант 8

Мета – навчитись вирішувати задачу суміщення зображень засобом видобування особливих точок і використання їх в процедурах метчінгу.

Завдання – вибрати з інтернету набори зображень з різною контрастністю і різними флуктуаціями освітленості. Для кожного зображення побудувати варіант спотвореного (видозміненого зображення)ю Для кожної отриманої пари побудувати дескриптор і проаналізувати можливість суміщення цих зображень і з визначенням параметрів геометричних перетворень. Використати BRISK. Для перевірки збігів необхідно написати власну функцію матчінгу, а результати її роботи перевірити засобами OpenCV. Якщо повної реалізації дескриптора немає в OpenCV, то такий необхідно створити власною функцією побудови цих дескрипторів. У цьому випадку матчінг можна здійснювати стандартними засобами.

Хід роботи

Для початку, напишемо функцію для роботи з дескриптором BRISK:

```
def brisk(image1, image2):
    # Initiate BRISK descriptor
    BRISK = cv.BRISK_create()

    # Find the keypoints and compute the descriptors for input
    keypoints1, descriptors1 = BRISK.detectAndCompute(image1, None)
    keypoints2, descriptors2 = BRISK.detectAndCompute(image2, None)

    print("Num of keypoints for image #1: {}, descriptors shape: {}".format(len(keypoints1), descriptors1.shape))
    print("Num of keypoints for image #2: {}, descriptors shape: {}".format(len(keypoints2), descriptors2.shape))

    return keypoints1, descriptors1, keypoints2, descriptors2
```

Після цього, напишемо функції метчингу з використання OpenCV та власну функцію метчингу:

```
def brut_match(keypoints1, descriptors1, keypoints2, descriptors2, image1, image2):  
    # create BFMatcher object  
    BFMatcher = cv.BFMatcher(normType = cv.NORM_HAMMING)  
  
    # Matching descriptor vectors using Brute Force Matcher  
    matches = BFMatcher.match(queryDescriptors = descriptors1,  
                              trainDescriptors = descriptors2)  
  
    # Sort them in the order of their distance  
    matches = sorted(matches, key = lambda x: x.distance)  
  
    res = cv.drawMatches(image1, keypoints1, image2, keypoints2, matches[:15],  
None, flags=2)  
    plt.figure(figsize=(20,20))  
    plt.imshow(res)  
    plt.show()
```

```
def handwrite_matcher(keypoints1, descriptors1, keypoints2, descriptors2, image1, image2):  
  
    matches = []  
    for i, k1 in enumerate(descriptors1):  
        for j, k2 in enumerate(descriptors2):  
            matches.append(cv.DMatch(_distance=cv.norm((k1 - k2), cv.NORM_HAMMING), _imgIdx=0, _queryIdx=i, _trainIdx=j))  
  
    matches = sorted(matches, key = lambda x : x.distance)  
  
    res = cv.drawMatches(image1, keypoints1, image2, keypoints2, matches[:15], None,  
], None,  
    flags=2)  
    plt.figure(figsize=(20,20))  
    plt.imshow(res)  
    plt.show()
```

Завантажимо та оглянемо зображення №1 та його спотворену версію в чорно-білому форматі.



Рис. 1. Оригінал першого зображення



Рис. 2. Спотворене перше зображення

Виконаємо те саме і для пари другого зображення:

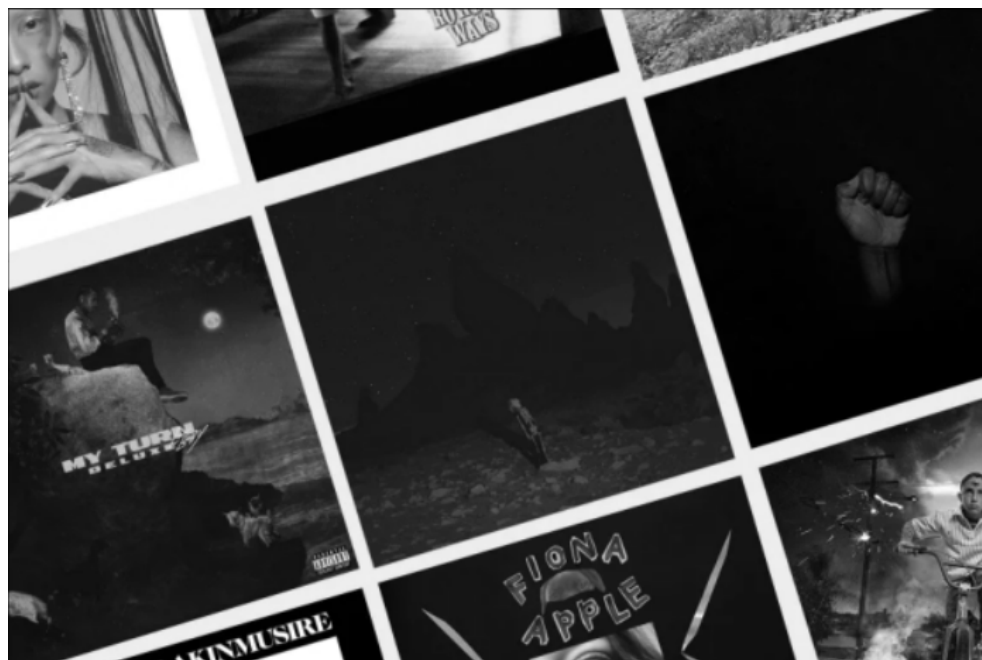


Рис. 3. Оригінал другого зображення



Рис. 4. Спотворене друге зображення

Тепер, виконаємо матчінг оригінального та спотвореного першого зображення:

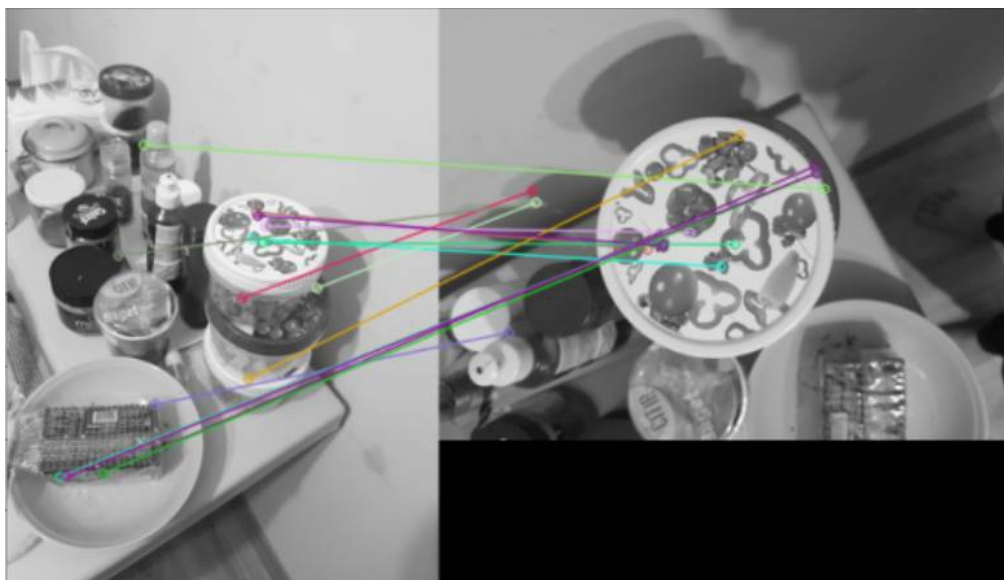


Рис. 5. Матчінг першого зображення з використанням матчеру OpenCV

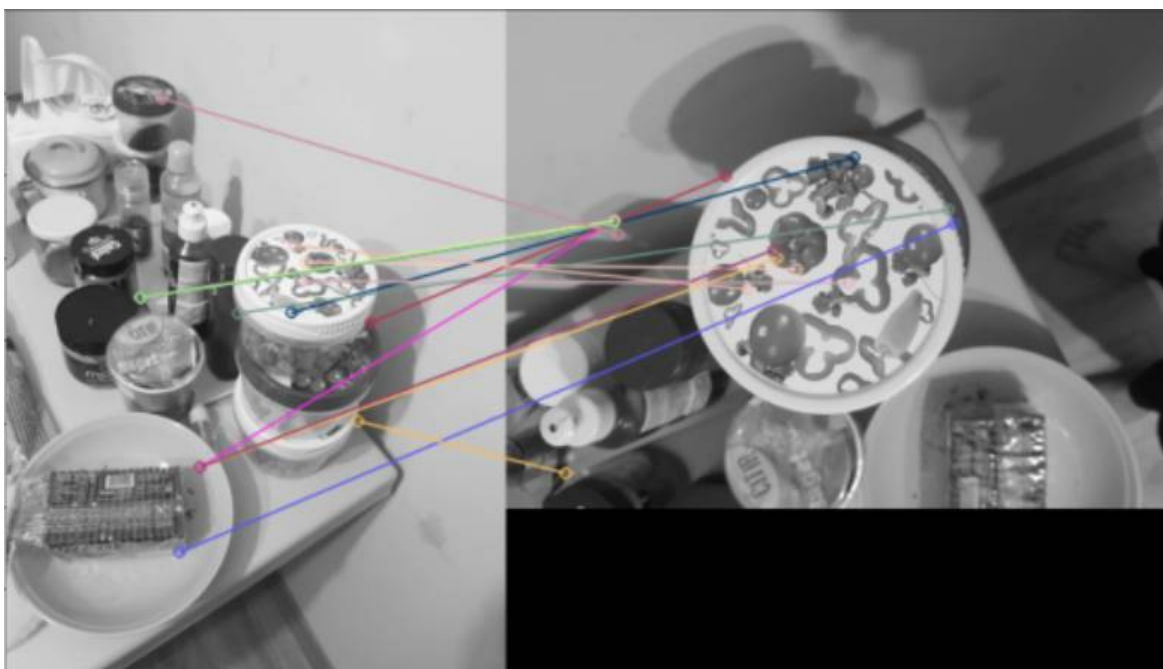


Рис. 6. Матчінг першого зображення з використанням власного матчеру

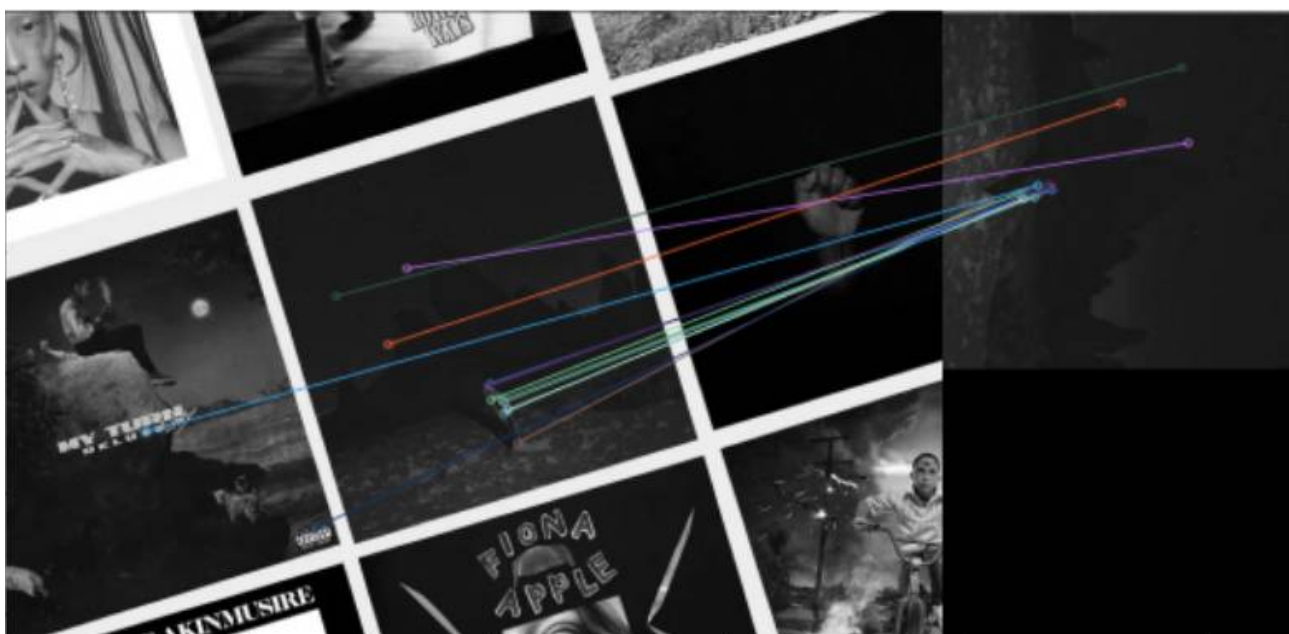


Рис. 7. Матчінг другого зображення з використанням матчеру OpenCV

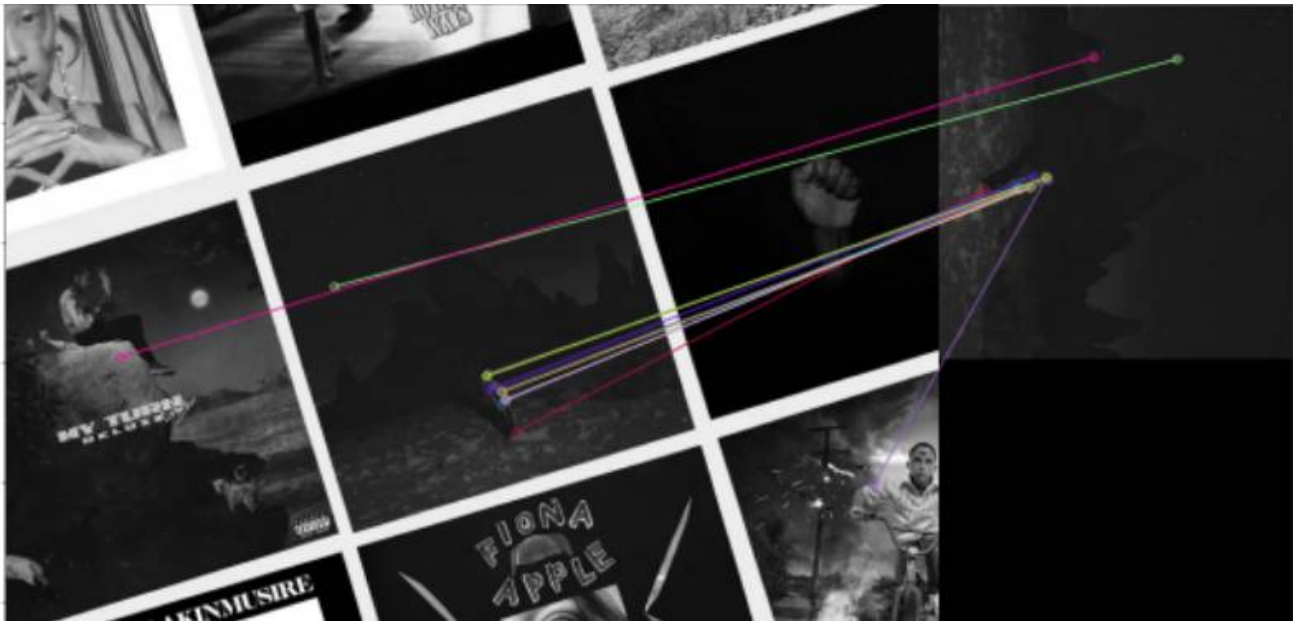


Рис. 8. Матчінг другого зображення з використанням власного матчеру

Як бачимо, обидва матчери доволі погано впорались з першим зображенням, оскільки його трансформація є дуже складною.

Щодо другого зображення, обидва матчери впорались добре і знайшли доволі схожі збіги.

Висновок: у ході виконання даної лабораторної роботи я навчився вирішувати задачу суміщення зображень засобом видобування особливих точок і використання їх в процедурах метчінгу.