

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
Кафедра систем штучного інтелекту



Лабораторна робота №2
З курсу «Обробка зображень методами штучного інтелекту»

Виконав:
ст.гр. КН-408
Петров Кирило

Перевірив:
Пелешко Д.

Львів – 2022

Суміщення зображень на основі використання дескрипторів.

Мета: навчитись вирішувати задачу суміщення зображень засобом видобування особливих точок і викорисання їх в процедурах матчіngu.

Завдання

12. Вибрати з інтернету набори зображень з різною контрастністю і різним флуктуаціями освітленості. Для кожного зображення побудувати варіант спотвореного (видозміненого зображення). Для кожної отриманої пари побудувати дескриптор і проаналізувати можливість суміщення цих зображень і з визначення параметрів геметричних перетворень (кут повороту, зміщень в напрямку x і напрямку y). SIFT.

Виконання

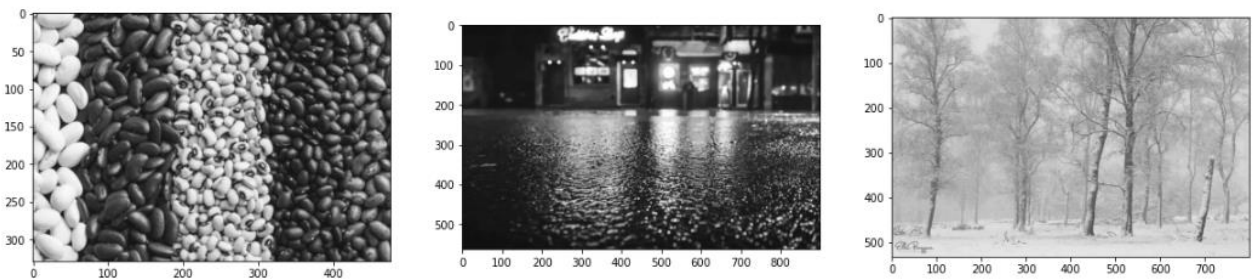


Рис. 1. Оригінальні зображення

Вирізавши певну частину з зображень і прогнавши через даний метод:

```
def blur(image):  
    kernel_size = (5,5)  
    blur = cv2.blur(image, kernel_size)  
    return blur
```

отримали наступні спотворені зображення:

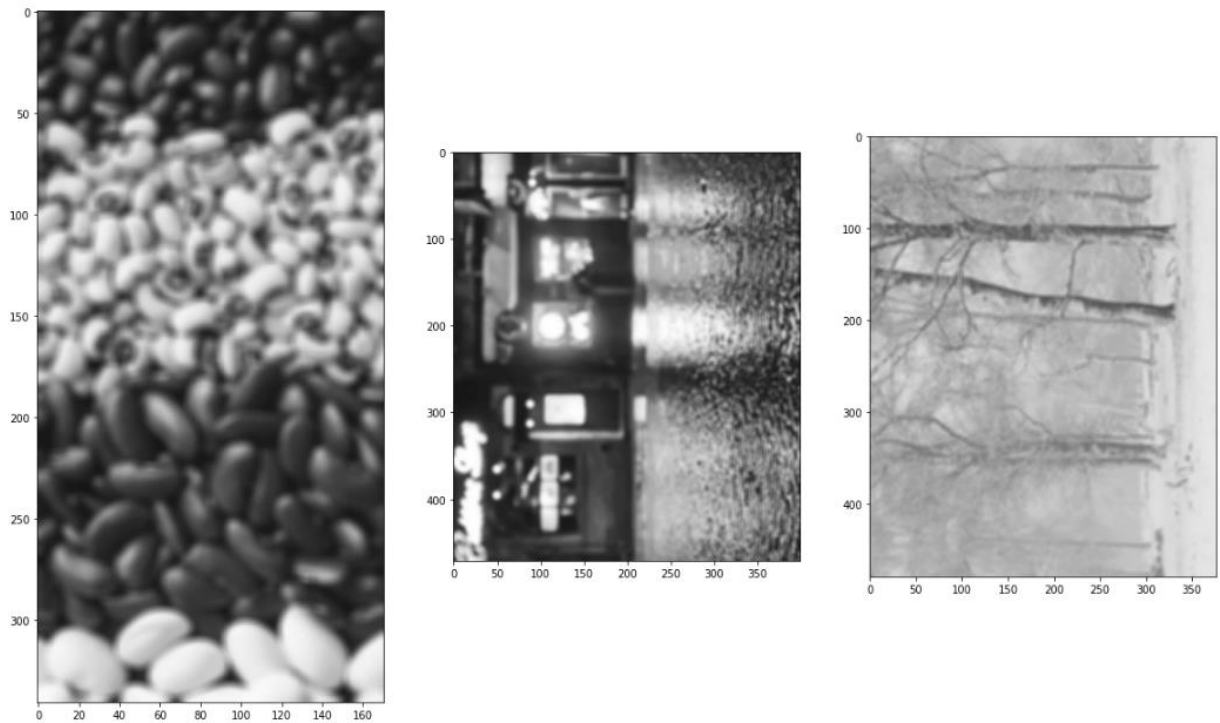


Рис. 2. Спотворені зображення.

Знайшовши необхідні ключові точки та їх дескриптори для зображень, співставляємо дескриптори за допомогою методу:

```
def match(des1, des2, ratio=0.85):
    matches1 = []
    matches2 = []
    distances = {}

    for i in range(len(des1)):
        des2_des1 = des2-des1[i]
        d = np.linalg.norm(des2_des1, ord=2, axis=1)
        sorted_args = np.argsort(d).tolist()

        if d[sorted_args[0]]/d[sorted_args[1]] <= ratio:
            matches1.append((i,sorted_args[0]))
            distances[f'{i}-{sorted_args[0]}'] = d[sorted_args[0]]

    # for cross-check
    for i in range(len(des2)):
        des1_des2 = des1-des2[i]
        d = np.linalg.norm(des1_des2, ord=2, axis=1)
        sorted_args = np.argsort(d).tolist()
```

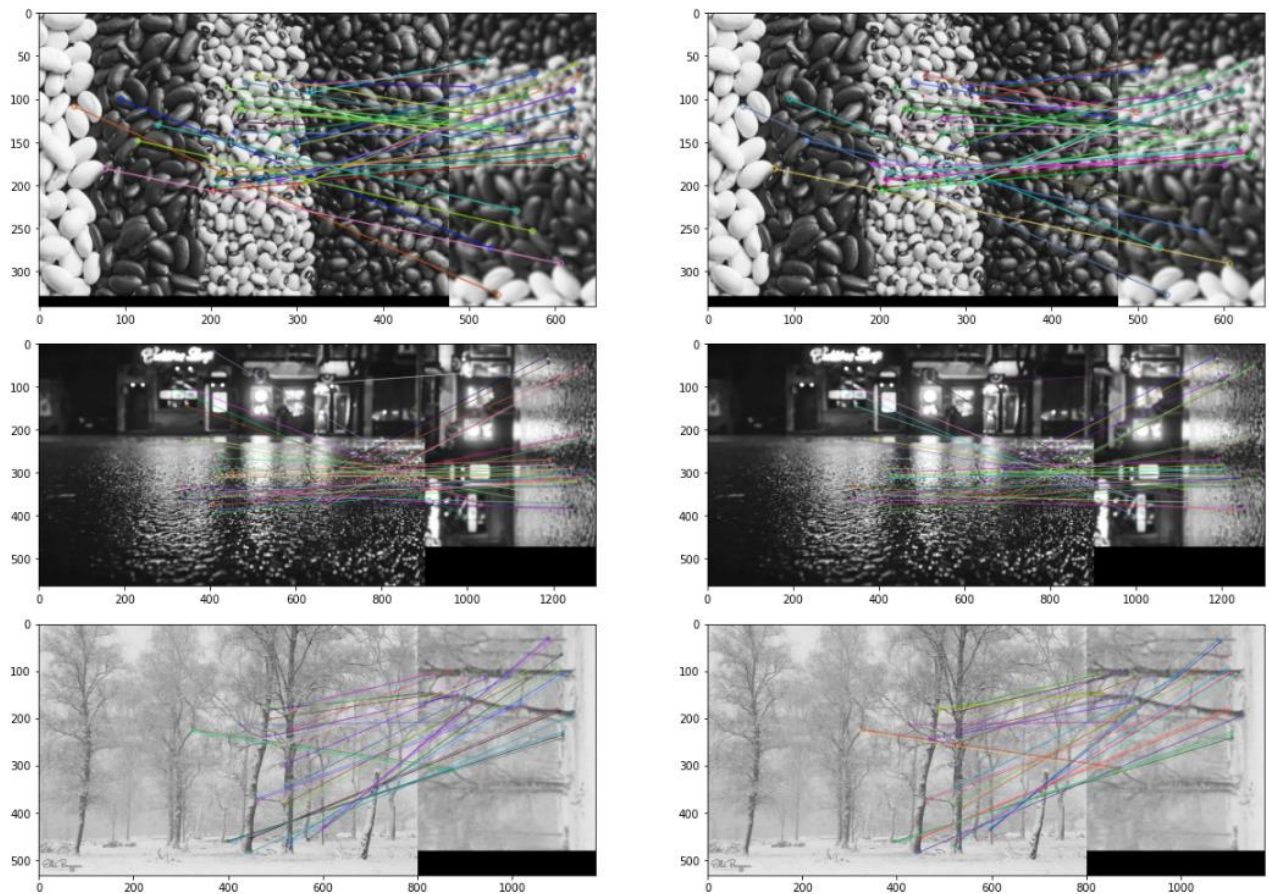
```

if d[sorted_args[0]]/d[sorted_args[1]] <= ratio:
    matches2.append((sorted_args[0],i))
    distances[f'{sorted_args[0]}-{i}'] = d[sorted_args[0]]

match = list(set(matches1).intersection(set(matches2)))
return [cv2.DMatch(args_pair[0], args_pair[1], distances[f'{args_pair[0]}-
{args_pair[1]}']) for args_pair in match]

```

Знаходимо найбільш близькі дескриптори за допомогою власного методу та flann-методу і сполучаємо їх точки на малюнках:



Висновок: в ході роботи я створив власну функцію для знаходження близьких дескрипторів, навчитися вирішувати задачу суміщення зображень засобом виобування особливих точок і викорисання їх в процедурах матчінугу.