МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ ЛЬВІВСЬКА ПОЛІТЕХНІКА ”
ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ

ЗВІТ

про виконання лабораторної роботи №2
з курсу “ Обробка зображень методами штучного інтелекту ”

Виконав:
ст. групи КН-410
Качмарик В. Р.
Перевірив:
Пелешко Д. Д.

Львів – 2022

Тема: Суміщення зображень на основі використання дескрипторів.

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

Варіант 11

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 геометричних перетворень (кут повороту, зміщень в напрямку x і напрямку y).

11. ORB

Для перевірки збігів необхідно написати власну функцію матчінгу, а результати її роботи перевірити засобами OpenCV. Якщо повної реалізації дескриптора не має в OpenCV, то такий необхідно створити власну функцію побудови цих дескрипторів. У цьому випадку матчінг можна здійснювати стандартними засобами (якщо це можливо).

Теоретичні відомості

1. Основи Brute-Force Matcher:

Brute-Force matcher (BF-matcher) є реалізовує простий матчінг метод. Він приймає дескриптор однієї ознаки в першій множині і зіставляє за деякою метрикою з усіма ознаками в другій множині. Як результат повертається найближчий дескриптор (ознака).

Для використання BF-matcher спочатку використовуючи функцію cv.BFMatcher() необхідно створити об'єкт BFMatcher. Функція має два необов'язкові параметри. Перший - normType. Він задає тип метрики для вимірювання відстані між дескрипторами. За замовчуванням

використовується метрика L2(cv.NORM_L2). Для SIFT та SURF і т.д. також рекомендується використовувати метрику L1(cv.NORM_L1).

Для дескрипторів, заснованих на бінарних рядках, таких як ORB, BRIEF, BRISK і т.д., треба використовувати метрику Хемінга (cv.NORM_ HAMMING). Якщо ORB використовує WTA_K == 3 або 4, то слід використовувати cv.NORM_HAMMING2.

Другий параметр функції створення матчера є булева змінна crossCheck, яка за замовчуванням рівна False. Якщо встановити її в True, то Matcher повертає тільки ті збіги, коли обидві ознаки в обох множинах збігаються одна з одною.

Після його створення матчера його двома важливими методами є BFMatcher.match() і BFMatcher.knnMatch(). Перший повертає кращий збіг. Другий метод - повертає k кращих збігів, де k задається параметром.

Як і ми використовували cv.drawKeypoints () для відтворення ключових точок, cv.drawMatches() допомагає нам малювати відповідності. Вона складає два зображення по горизонталі і малює лінії від першого зображення до другого, показуючи найкращі збіги. Також існує cv.drawMatchesKnn, який відображає всі k кращих збігів. Якщо k = 2, то він відобразить дві лінії збігів для кожної ключової точки. Тому ми повинні передати маску, якщо ми хочемо вибірково намалювати її.

2. ORB (Orientated FAST and Rotated BRIEF):

ORB справляється так само добре, як і SIFT, із завданням виявлення ознак (і краще, ніж SURF), будучи майже вдвічі швидшим. ORB базується на добре відомому детекторі ключових точок FAST і дескрипторі BRIEF. Обидві ці методики привабливі завдяки своїй хорошій продуктивності та невисокій “вартості”. Основний внесок ORB полягає в наступному:

- Додавання швидкого та точного компонента орієнтації до FAST;
- Ефективне обчислення орієнтованих BRIEF функцій;
- Аналіз дисперсій та кореляції орієнтованих BRIEF ознак;

- Метод навчання для декореляції характеристик BRIEF за ротаційної інваріантності, що призводить до кращої продуктивності в програмах найближчого сусіда.

Виконання

Зображення №1 з спотвореним:

CV2 matcher:

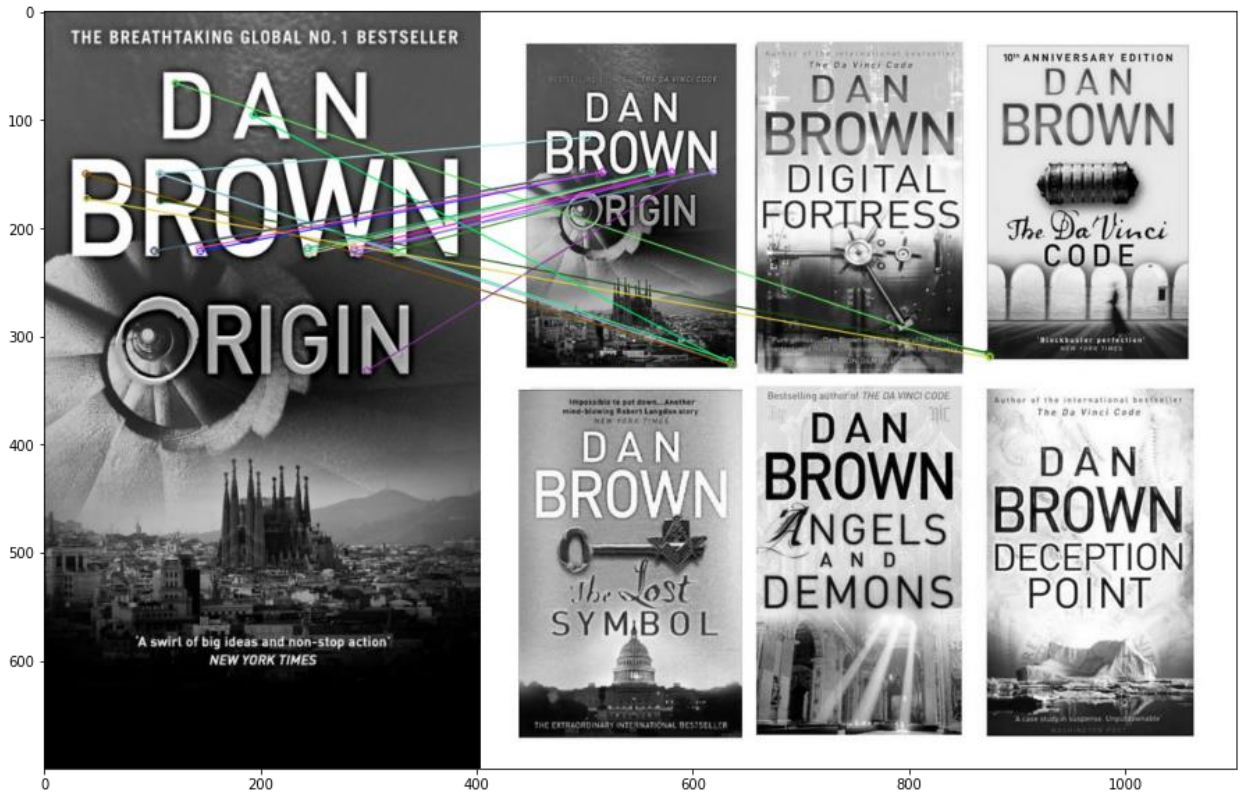
Own matcher (hamming normalization):

Own matcher (L1 normalization):

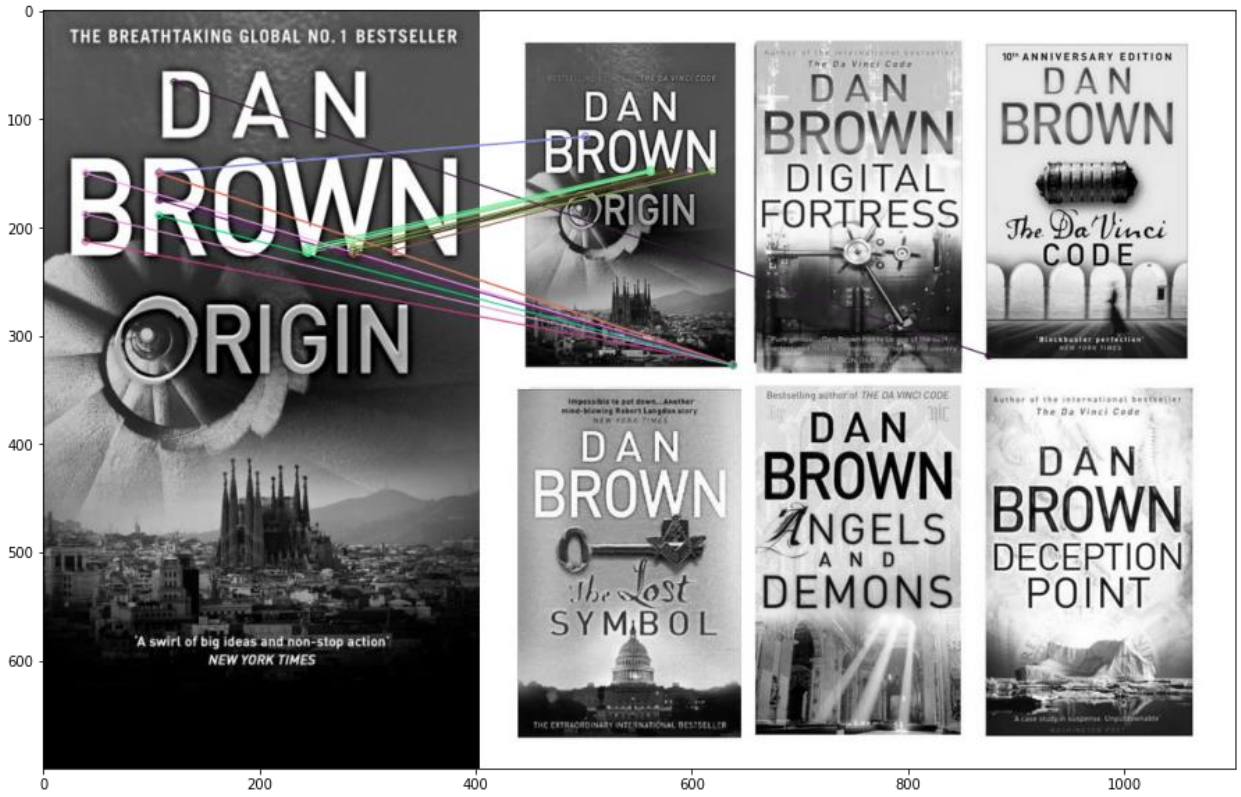
Own matcher (L2 normalization):

Зображення №2 з спотвореним:

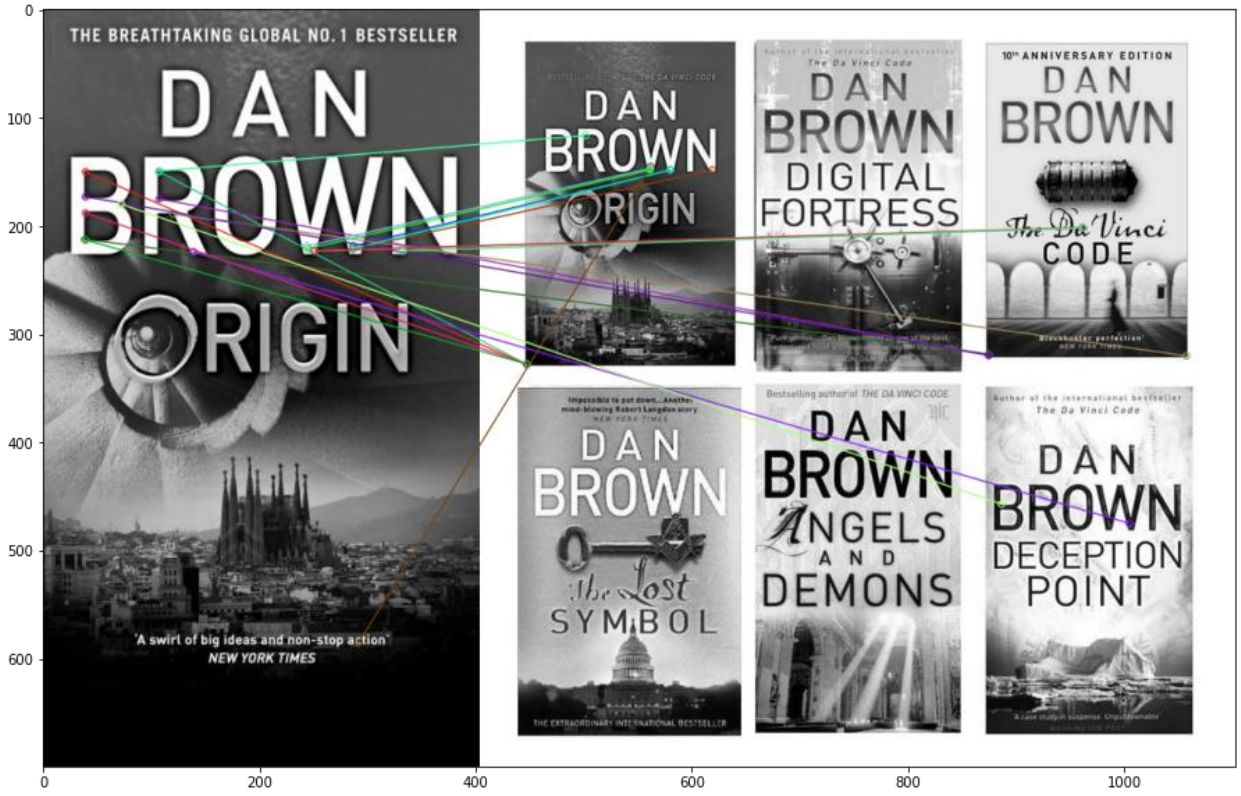
CV2 matcher:



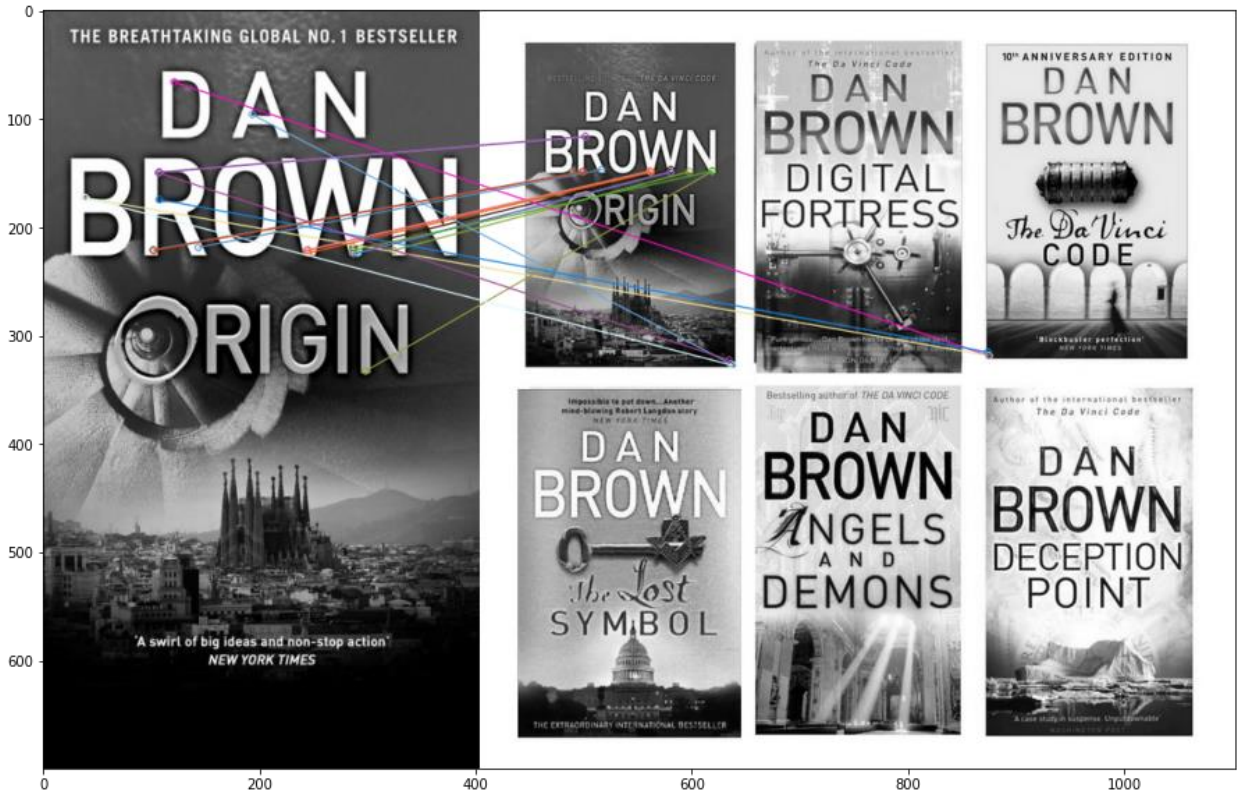
Own matcher (hamming normalization):



Own matcher (L1 normalization):

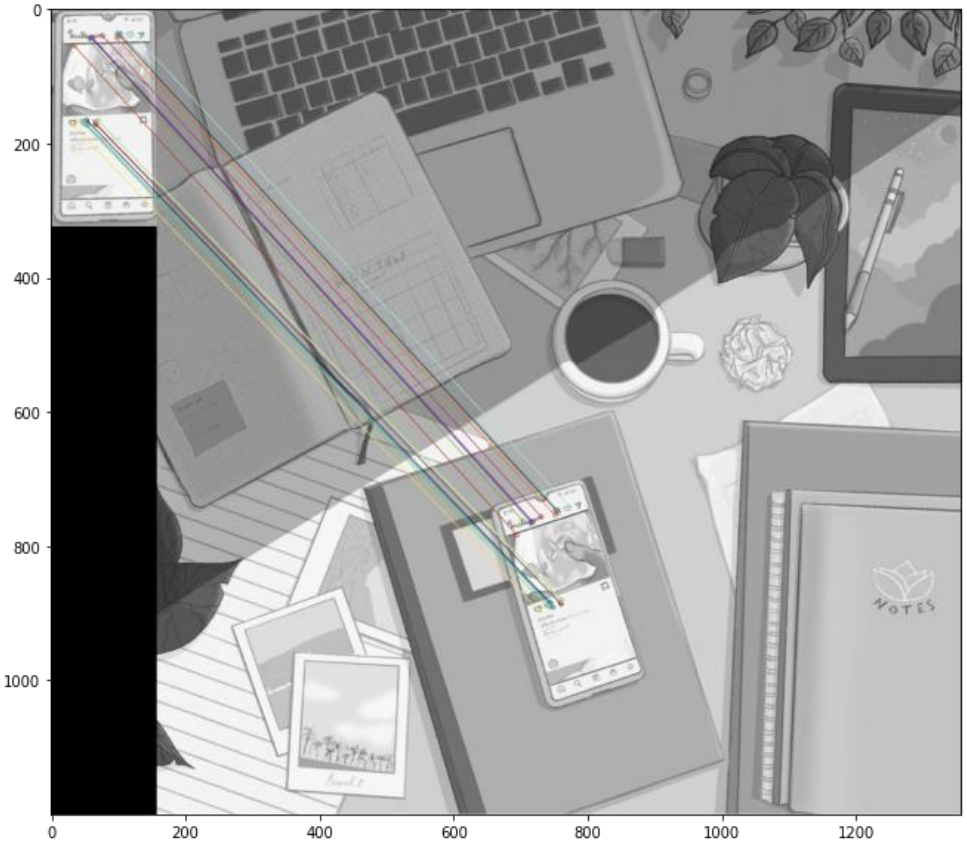


Own matcher (L2 normalization):

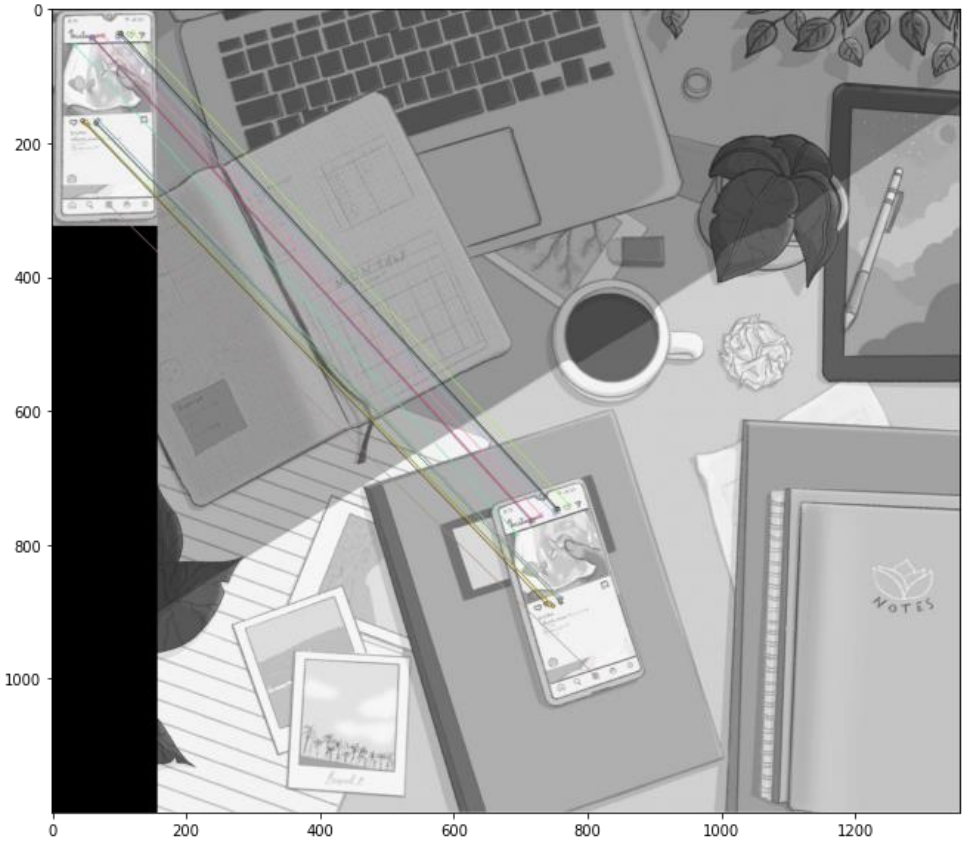


Зображення №3 з спотвореним:

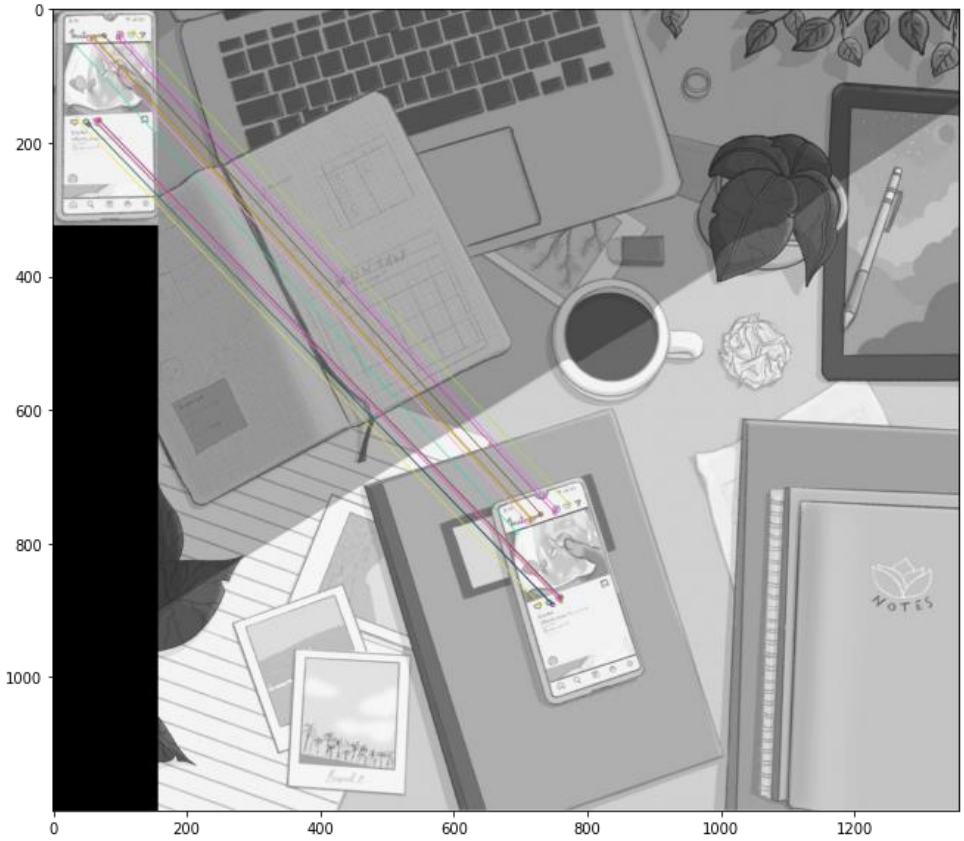
CV2 matcher:



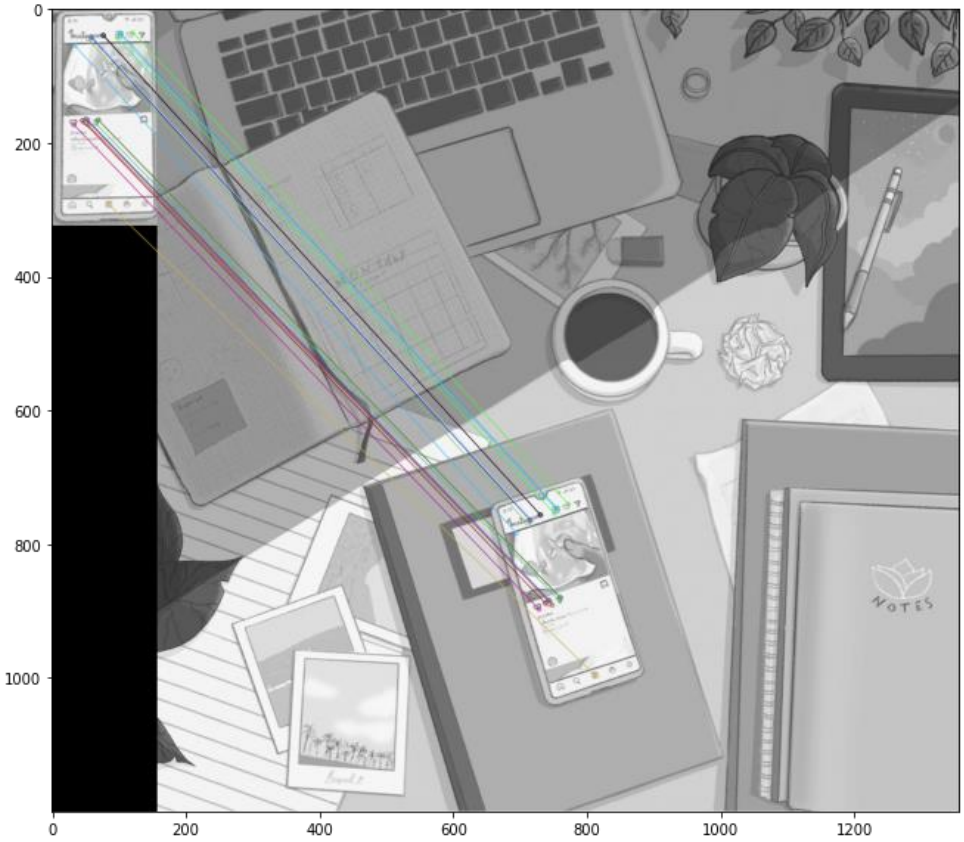
Own matcher (hamming normalization):



Own matcher (L1 normalization):

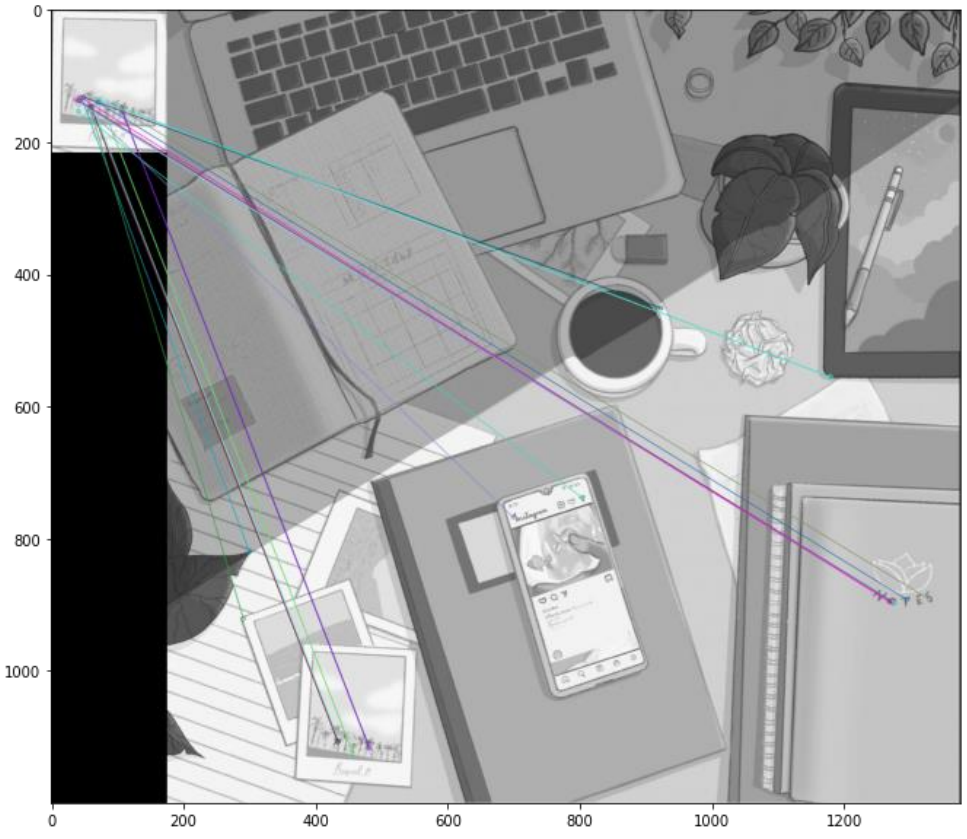


Own matcher (L2 normalization):

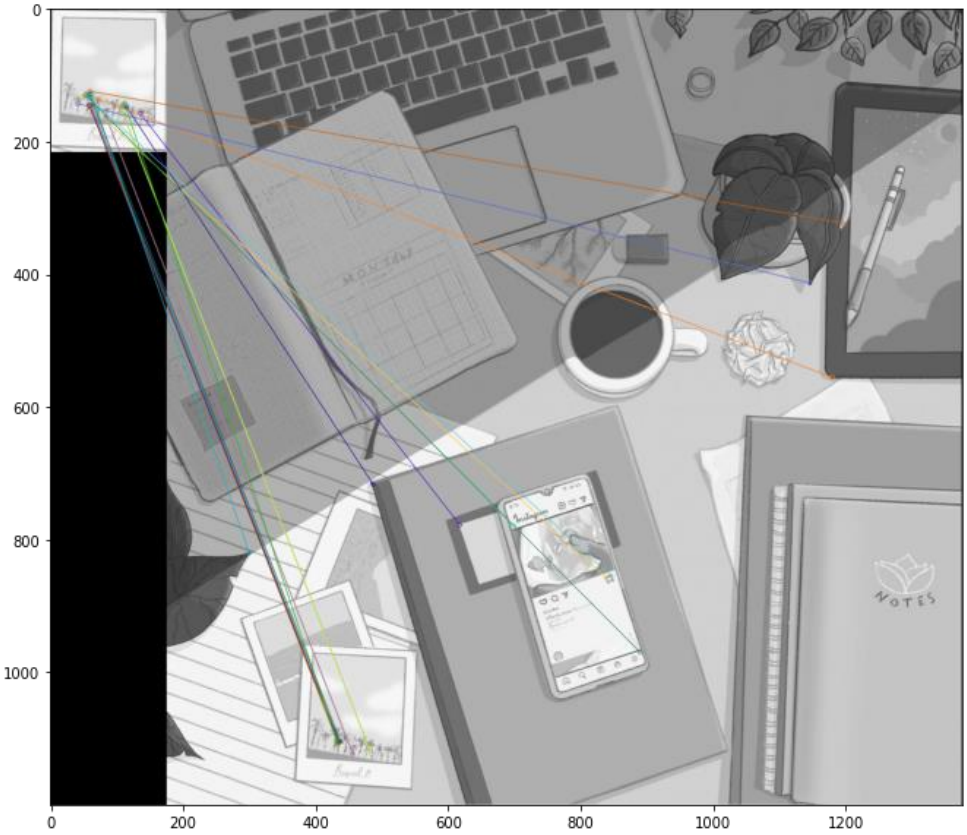


Зображення №4 з спотвореним:

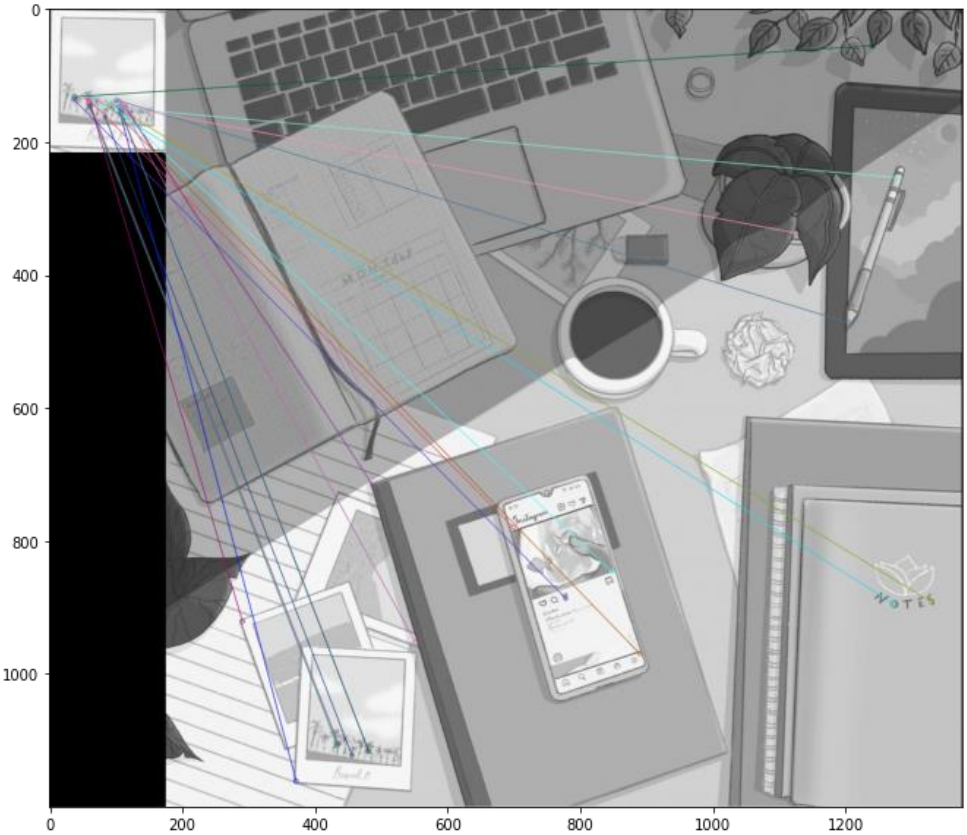
CV2 matcher:



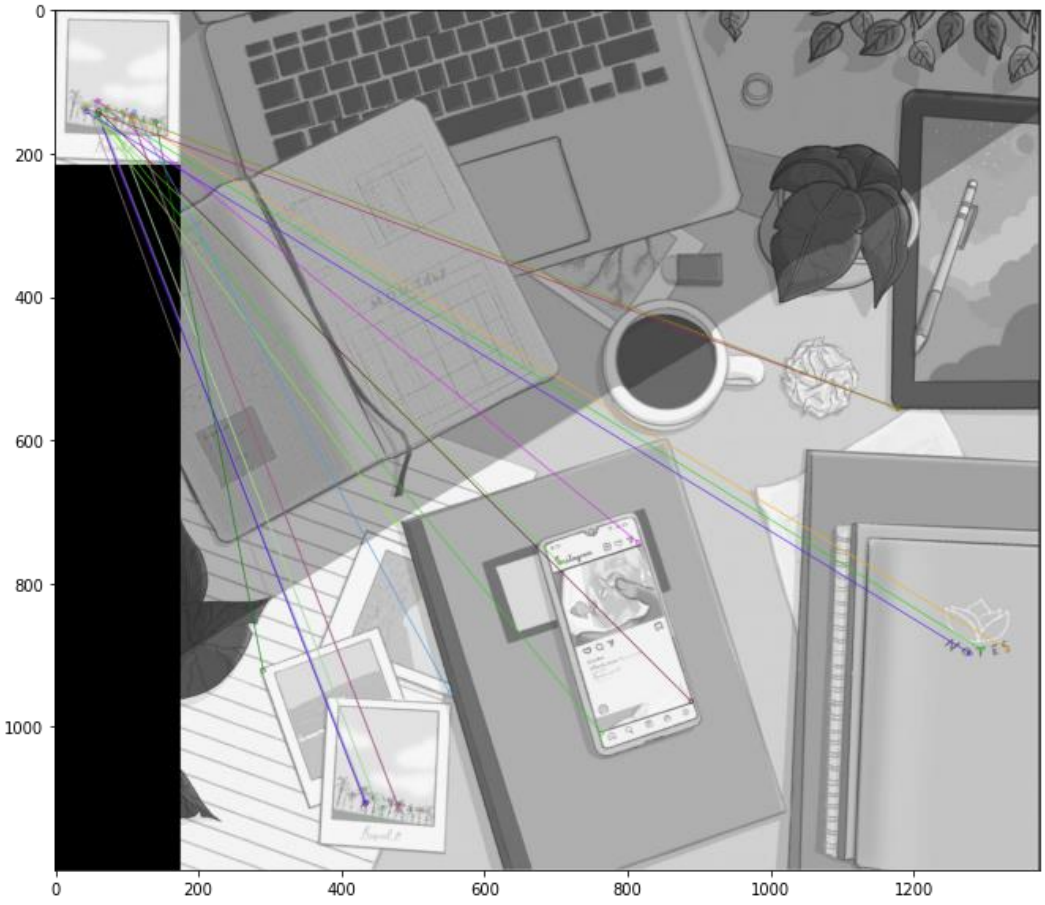
Own matcher (hamming normalization):



Own matcher (L1 normalization):



Own matcher (L2 normalization):



Висновок

Під час виконання цієї лабораторної роботи я навчився вирішувати задачу суміщення зображень засобом видобування особливих точок і використання їх в процедурах матчінгу.

З отриманих результатів (матчінгу за допомогою різних нормалізацій), було досліджено, що з ORB дескриптором найкраще справилась відстань Геммінга при застосуванні Brute-Force Matcher.