МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Кафедра «Системи штучного інтелекту»



**Звіт**

**до лабораторної роботи №2**

**З дисципліни «Обробка зображень методами штучного інтелекту»**

**Виконав:**

студент групи КН-409

Слава Л.Л.

**Прийняв:**

Пелешко Д.Д.

*Львів-2022*

**Лабораторна робота 2**

Суміщення зображень на основі використання дескрипторів

Варіант 1 SIFT

Після вибору двох зображень спотворимо їх:



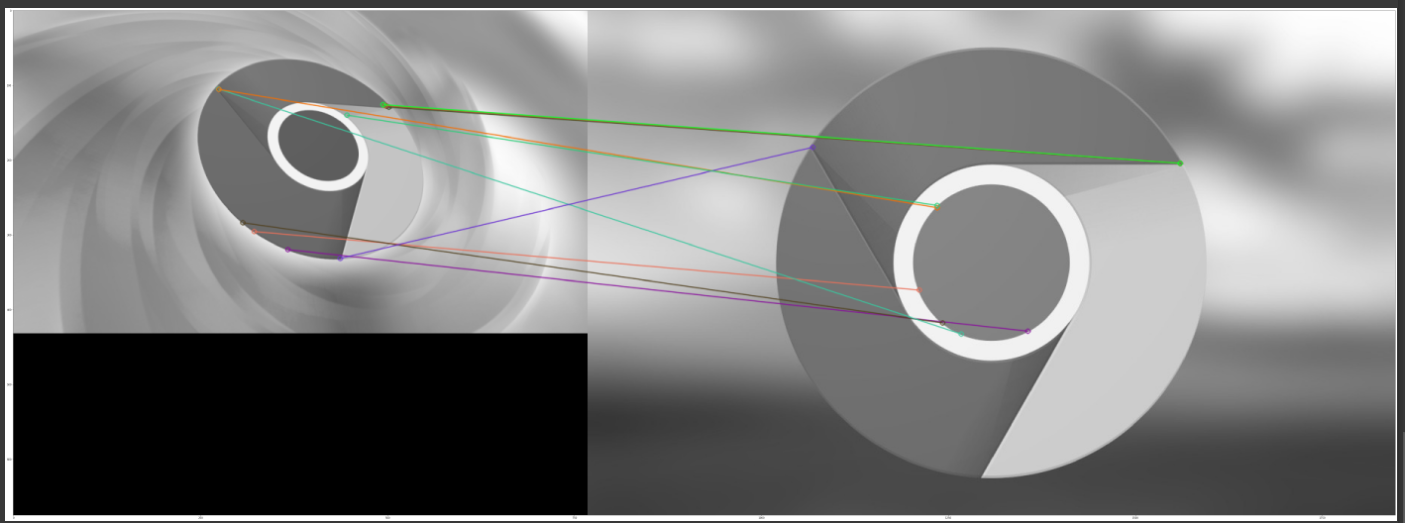
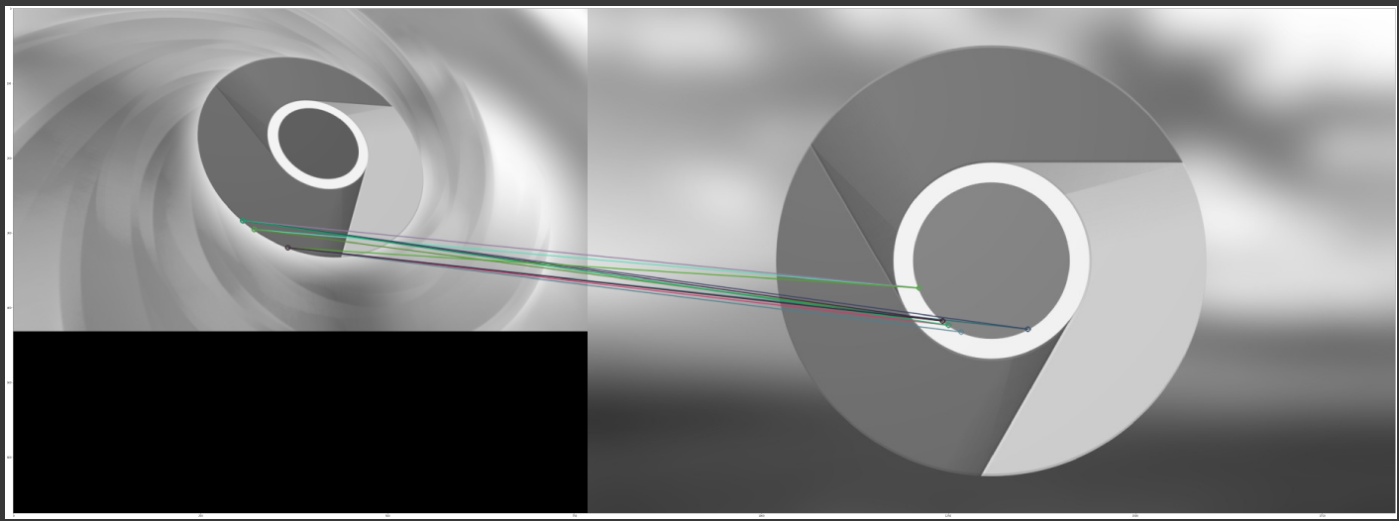
Після цього перевіримо результати:



Та власного метчера:



Тепер на іншій картинці:

Код:

!pip uninstall opencv-python

!pip uninstall opencv-contrib-python

!pip install opencv-python==4.4.0.44

!pip install opencv-contrib-python==4.4.0.44

def matcher(des1, des2, kp1, kp2, topn=11):

    matcher = cv.BFMatcher(normType=cv.NORM\_L1)

    matches = matcher.match(des1,des2)

    top\_matches = sorted(matches, key = lambda x : x.distance)[:topn]

    matched\_image = cv.drawMatches(img1,kp1,img2,kp2,top\_matches,None,flags=cv.DrawMatchesFlags\_NOT\_DRAW\_SINGLE\_POINTS)

    return matched\_image

def custom\_matcher(des1, des2, kp1, kp2, topn=11):

    matches = []

    for i, k1 in enumerate(des1):

        for j, k2 in enumerate(des2):

            matches.append(cv.DMatch(\_distance=np.linalg.norm((k1 - k2), ord=1), \_imgIdx=0, \_queryIdx=i,\_trainIdx=j))

    top\_matches = sorted(matches, key = lambda x : x.distance)[:topn]

    matched\_image = cv.drawMatches(img1,kp1,img2,kp2,top\_matches,None,flags=cv.DrawMatchesFlags\_NOT\_DRAW\_SINGLE\_POINTS)

    return matched\_image

import numpy as np

import cv2 as cv

import matplotlib.pyplot as plt

img1 = cv.imread('/content/Bird.jpg',cv.IMREAD\_GRAYSCALE)

img2 = cv.imread('/content/Birg-rotated.jpg',cv.IMREAD\_GRAYSCALE)

sift = cv.xfeatures2d.SIFT\_create()

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

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

cv\_bf\_img = matcher(des1, des2, kp1, kp2)

plt.figure(figsize=(100,100))

plt.imshow(cv\_bf\_img),plt.show()

custom\_bf\_img = custom\_matcher(des1, des2, kp1, kp2)

plt.figure(figsize=(100,100))

plt.imshow(custom\_bf\_img),plt.show()

img1 = cv.imread('/content/pic-2-new.jpg',cv.IMREAD\_GRAYSCALE)

img2 = cv.imread('/content/pic-2.jpg',cv.IMREAD\_GRAYSCALE)

sift = cv.xfeatures2d.SIFT\_create()

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

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

cv\_bf\_img = matcher(des1, des2, kp1, kp2)

plt.figure(figsize=(100,100))

plt.imshow(cv\_bf\_img),plt.show()

custom\_bf\_img = custom\_matcher(des1, des2, kp1, kp2)

plt.figure(figsize=(100,100))

plt.imshow(custom\_bf\_img),plt.show()