Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра «Системи штучного інтелекту»

**Лабораторна робота №2**

# З дисципліни «Обробка зображень методами штучного інтелекту»

Виконав студент групи КН-408 Горностай Б. Я.

Прийняв Пелешко Д. Д.

Львів 2022

Варіант 10

# Завдання

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 х і напрямку y).

10. BRIEF

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 дискрипторів. У цьому випадку  матчінг можна здійснювати стандартними засобами (якщо це можливо).

# Код програми

from google.colab.patches import cv2\_imshow

import urllib.request

import numpy as np

import cv2

def import\_url\_img(url):

  url\_response = urllib.request.urlopen(url)

  return cv2.imdecode(np.array(bytearray(url\_response.read()), dtype=np.uint8), -1)

url = "https://upload.wikimedia.org/wikipedia/commons/4/41/Sunflower\_from\_Silesia2.jpg"

img = import\_url\_img(url)

img = cv2.resize(img, (img.shape[1]//5, img.shape[0]//5))

# img = cv2.imread('./2.png')

training\_image = img

training\_gray = cv2.cvtColor(training\_image, cv2.COLOR\_RGB2GRAY)

test\_image = cv2.pyrDown(training\_image)

#test\_image = cv2.pyrDown(test\_image)

#test\_image = training\_image

# num\_rows, num\_cols = test\_image.shape[:2]

# rotation\_matrix = cv2.getRotationMatrix2D((num\_cols/2, num\_rows/2), 47, 1)

# test\_image = cv2.warpAffine(test\_image, rotation\_matrix, (num\_cols, num\_rows))

test\_gray = cv2.cvtColor(test\_image, cv2.COLOR\_RGB2GRAY)

def fast\_plus\_brief(img\_gray):

  fast = cv2.FastFeatureDetector\_create()

  brief = cv2.xfeatures2d.BriefDescriptorExtractor\_create()

  keypoints = fast.detect(img\_gray, None)

  keypoints, descriptor = brief.compute(img\_gray, keypoints)

  return keypoints,descriptor

def bf\_Matcher(train\_descriptor, test\_descriptor):

  bf = cv2.BFMatcher(cv2.NORM\_HAMMING, crossCheck = True)

  matches = bf.match(train\_descriptor, test\_descriptor)

  return sorted(matches, key = lambda x : x.distance)

train\_keypoints, train\_descriptor = fast\_plus\_brief(training\_gray)

test\_keypoints, test\_descriptor = fast\_plus\_brief(test\_gray)

keypoints\_img = cv2.drawKeypoints(training\_gray, train\_keypoints, training\_gray, flags = cv2.DRAW\_MATCHES\_FLAGS\_DRAW\_RICH\_KEYPOINTS)

cv2\_imshow(keypoints\_img)

matches\_1 = bf\_Matcher(train\_descriptor,test\_descriptor)

result\_1 = cv2.drawMatches(training\_gray, train\_keypoints, test\_gray, test\_keypoints, matches\_1[:10], None, flags = 2)

cv2\_imshow(result\_1)

matches\_2 = custom\_matcher(train\_descriptor, test\_descriptor)

result\_2 = cv2.drawMatches(training\_gray, train\_keypoints, test\_gray, test\_keypoints, matches\_2[:10], test\_gray,

                        flags=2)

cv2\_imshow(result\_2)

# Результати роботи

Імпортуємо оригінальне зображення

**Зображення, що містить рослина, соняшник, квітка

Автоматично згенерований опис**

Зменшуємо його розмірність для отримання видозміненого зображення

Зображення, що містить рослина, квітка, соняшник

Автоматично згенерований опис

Виводимо ключові точки користуючись алгоритмом детектора FAST, дескриптимо точки за допомогою алгоритму BRIEF.

Зображення, що містить квітка, рослина

Автоматично згенерований опис

Використовуючи вбудовану функцію opencv алгоритму Brute Force matcher знаходимо найбільш подібні пари.

Зображення, що містить рослина, квітка

Автоматично згенерований опис

Реалізуємо власний Brute Force matcher шляхом порівнення відстаней усіх точок між собою.

Зображення, що містить рослина, квітка

Автоматично згенерований опис

# Висновок

Під час виконання цієї лабораторної роботи було для кожної отриманої пари побудувано дескриптор з застосуванням алгоритму BRIEF. Для перевірки збігів написано власну функцію матчінгу, а результати її роботи  перевірено засобами OpenCV.