**ЛАБОРАТОРНА РОБОТА №8**

**ДОСЛІДЖЕННЯ МЕТОДІВ КОМП’ЮТЕРНОГО ЗОРУ**

**Мета:** використовуючи спеціалізовані бібліотеки та мову програмування Python навчитися обробляти зображення за допомогою бібліотеки OpenCV.

**Хід роботи:**

**GitHub:** https://github.com/ipz201svo/AI

**Завдання 1:** Завантаження зображень та відео в OpenCV.

import cv2

frameWidth = 640

frameHeight = 480

cap = cv2.VideoCapture(0)

cap.set(3, frameWidth)

cap.set(4, frameHeight)

cap.set(10,150)

while True:

    success, img = cap.read()

    cv2.imshow("Result", img)

    if cv2.waitKey(1) & 0xFF == ord('q'):

        break

import cv2

img = cv2.imread("skakovskyi.jpg")

# DISPLAY

cv2.imshow("camera", img)

cv2.waitKey(0)

Зображення, що містить особа, Обличчя людини, знімок екрана, просто неба

Автоматично згенерований опис

Рис. 1.1 – Результат виконання програми

**Завдання 2:** Дослідження перетворень зображення.

import cv2

import numpy as np

img = cv2.imread("skakovskyi.jpg")

kernel = np.ones((5, 5), np.uint8)

imgGray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

imgBlur = cv2.GaussianBlur(imgGray, (7, 7), 0)

imgCanny = cv2.Canny(img, 150, 200)

imgDialation = cv2.dilate(imgCanny, kernel, iterations=1)

imgEroded = cv2.erode(imgDialation, kernel, iterations=1)

cv2.imshow("Gray Image", imgGray)

cv2.imshow("Blur Image", imgBlur)

cv2.imshow("Canny Image", imgCanny)

cv2.imshow("Dialation Image", imgDialation)

cv2.imshow("Eroded Image", imgEroded)

cv2.waitKey(0)

Зображення, що містить текст, графічний дизайн, знімок екрана, мистецтво

Автоматично згенерований опис

Рис. 2.1 – Результат виконання програми

Зображення, що містить текст, знімок екрана, Мультимедійне програмне забезпечення, Редагування

Автоматично згенерований опис

Рис. 2.2 – Результат виконання програми

Зображення, що містить текст, знімок екрана, графічний дизайн, мистецтво

Автоматично згенерований опис

Рис. 2.3 – Результат виконання програми

Зображення, що містить Обличчя людини, особа, знімок екрана, одежа

Автоматично згенерований опис

Рис. 2.4 – Результат виконання програми

Зображення, що містить Обличчя людини, особа, знімок екрана, просто неба

Автоматично згенерований опис

Рис. 2.5 – Результат виконання програми

**Завдання 3:** Вирізання частини зображення.

import cv2

import numpy as np

img = cv2.imread("skakovskyi.jpg")

print(img.shape)

imgResize = cv2.resize(img, (500, 1000))

print(imgResize.shape)

imgCropped = img[350:900, 250:650]

cv2.imshow("Image", img)

# cv2.imshow("Image Resize",imgResize)

cv2.imshow("Image Cropped", imgCropped)

cv2.waitKey(0)

Зображення, що містить Обличчя людини, особа, брови, Селфі

Автоматично згенерований опис

Рис. 3.1 – Результат виконання програми

**Завдання 4:** Розпізнавання обличчя на зображенні.

import cv2

faceCascade = cv2.CascadeClassifier("./haarcascade\_frontalface\_default.xml")

img = cv2.imread("skakovskyi.jpg")

imgGray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

faces = faceCascade.detectMultiScale(imgGray, 1.1, 4)

for x, y, w, h in faces:

    cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)

cv2.imshow("Result", img)

cv2.waitKey(0)

Зображення, що містить особа, Обличчя людини, просто неба, знімок екрана

Автоматично згенерований опис

Рис. 4.1 – Результат виконання програми

**Завдання 5:** Розпізнавання об’єктів на зображенні за допомогою методів зіставлення шаблонів (Template Matching).

import cv2 as cv

import numpy as np

from matplotlib import pyplot as plt

img = cv.imread("skakovskyi.jpg", 0)

img2 = img.copy()

template = cv.imread("skakovskyi\_face.jpg", 0)

w, h = template.shape[::-1]

# All the 6 methods for comparison in a list

methods = [

    "cv.TM\_CCOEFF",

    "cv.TM\_CCOEFF\_NORMED",

    "cv.TM\_CCORR",

    "cv.TM\_CCORR\_NORMED",

    "cv.TM\_SQDIFF",

    "cv.TM\_SQDIFF\_NORMED",

]

for meth in methods:

    img = img2.copy()

    method = eval(meth)

    # Apply template Matching

    res = cv.matchTemplate(img, template, method)

    min\_val, max\_val, min\_loc, max\_loc = cv.minMaxLoc(res)

    # If the method is TM\_SQDIFF or TM\_SQDIFF\_NORMED, take minimum

    if method in [cv.TM\_SQDIFF, cv.TM\_SQDIFF\_NORMED]:

        top\_left = min\_loc

    else:

        top\_left = max\_loc

    bottom\_right = (top\_left[0] + w, top\_left[1] + h)

    cv.rectangle(img, top\_left, bottom\_right, 255, 2)

    plt.subplot(121), plt.imshow(res, cmap="gray")

    plt.title("Matching Result"), plt.xticks([]), plt.yticks([])

    plt.subplot(122), plt.imshow(img, cmap="gray")

    plt.title("Detected Point"), plt.xticks([]), plt.yticks([])

    plt.suptitle(meth)

    plt.show()

Зображення, що містить Обличчя людини, одежа, знімок екрана, чоловік

Автоматично згенерований опис

Рис. 5.1 – Результат виконання програми

Зображення, що містить Обличчя людини, текст, одежа, знімок екрана

Автоматично згенерований опис

Рис. 5.2 – Результат виконання програми

Зображення, що містить Обличчя людини, усмішка, одежа, знімок екрана

Автоматично згенерований опис

Рис. 5.3 – Результат виконання програми

Зображення, що містить Обличчя людини, текст, одежа, чоловік

Автоматично згенерований опис

Рис. 5.4 – Результат виконання програми

Зображення, що містить Обличчя людини, текст, одежа, особа

Автоматично згенерований опис

Рис. 5.5 – Результат виконання програми

Зображення, що містить Обличчя людини, текст, одежа, знімок екрана

Автоматично згенерований опис

Рис. 5.6 – Результат виконання програми

**Завдання 6:** Сегментація зображення алгоритмом водорозподілу.

import numpy as np

import cv2

from matplotlib import pyplot as plt

img = cv2.imread('coins.jpg')

cv2.imshow("coins",img)

cv2.waitKey(0)

gray = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)

ret, thresh = cv2.threshold(gray,0,255,cv2.THRESH\_BINARY\_INV+cv2.THRESH\_OTSU)

cv2.imshow("coins bin ",thresh)

cv2.waitKey(0)

# видалення шуму

kernel = np.ones((3,3),np.uint8)

opening = cv2.morphologyEx(thresh,cv2.MORPH\_OPEN,kernel, iterations = 2)

# певна фонова область

sure\_bg = cv2.dilate(opening,kernel,iterations=3)

# Пошук впевненої області переднього плану

dist\_transform = cv2.distanceTransform(opening,cv2.DIST\_L2,5)

ret, sure\_fg = cv2.threshold(dist\_transform,0.7\*dist\_transform.max(),255,0)

# Пошук невідомого регіону

sure\_fg = np.uint8(sure\_fg)

unknown = cv2.subtract(sure\_bg,sure\_fg)

cv2.imshow("coins ",opening)

cv2.waitKey(0)

# Маркування міток

ret, markers = cv2.connectedComponents(sure\_fg)

# Додайте один до всіх міток, щоб впевнений фон був не 0, а 1

markers = markers+1

# Тепер позначте область невідомого нулем

markers[unknown==255] = 0

markers = cv2.watershed(img,markers)

img[markers == -1] = [255,0,0]

cv2.imshow("coins\_markers",img)

cv2.waitKey(0)

Зображення, що містить знімок екрана, монета, порівняння

Автоматично згенерований опис

Рис. 6.1 – Результат виконання програми

Зображення, що містить знімок екрана

Автоматично згенерований опис

Рис. 6.2 – Результат виконання програми

Зображення, що містить знімок екрана

Автоматично згенерований опис

Рис. 6.3 – Результат виконання програми

Зображення, що містить знімок екрана, коло, порівняння

Автоматично згенерований опис

Рис. 6.4 – Результат виконання програми

**Висновок:** під час виконання лабораторної роботи, використовуючи спеціалізовані бібліотеки та мову програмування Python навчилися обробляти зображення за допомогою бібліотеки OpenCV.