

ЛАБОРАТОРНА РОБОТА № 8

ДОСЛІДЖЕННЯ МЕТОДІВ КОМП'ЮТЕРНОГО ЗОРУ

Мета: використовуючи спеціалізовані бібліотеки та мову програмування Python навчити-ся обробляти зображення за допомогою бібліотеки OpenCV.

Хід роботи:

Завдання 1. Завантаження зображень та відео в OpenCV

Лістинг коду файлу Task_1.py:

```
import cv2

def get_image():
    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

def load_image():
    img = cv2.imread("Chyzhmotria.png")
    cv2.imshow("Output", img)
    cv2.waitKey(0)

if __name__ == "__main__":
    # get_image()
    load_image()
```

					ДЧ «Житомирська політехніка».19.121.22.000 – Лр8			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Чижемотря М.О.			Звіт з лабораторної роботи		Літ.	Арк.
Перевір.		Пулеко І.В.						1
Керівник								12
Н. контр.							ФІКТ Гр. ІПЗ-19-1[2]	
Зав. каф.								



Рис.8.1 – Збережене та завантажене зображення з веб-камери

Завдання 2. Дослідження перетворень зображення

Лістинг коду файлу Task_2.py:

```
import cv2
import numpy as np

img = cv2.imread("Chyzhmotria.png")
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)
imgDilation = cv2.dilate(imgCanny, kernel, iterations=1)
imgEroded = cv2.erode(imgDilation, kernel, iterations=1)

cv2.imshow("Gray Image", imgGray)
cv2.imshow("Blur Image", imgBlur)
cv2.imshow("Canny Image", imgCanny)
cv2.imshow("Dilation Image", imgDilation)
cv2.imshow("Eroded Image", imgEroded)
cv2.waitKey(0)
```

		Чижмотря М.О.			ДУ «Житомирська політехніка».19.121.2.000 – Лр8	Арк.
		Пулеко І.В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

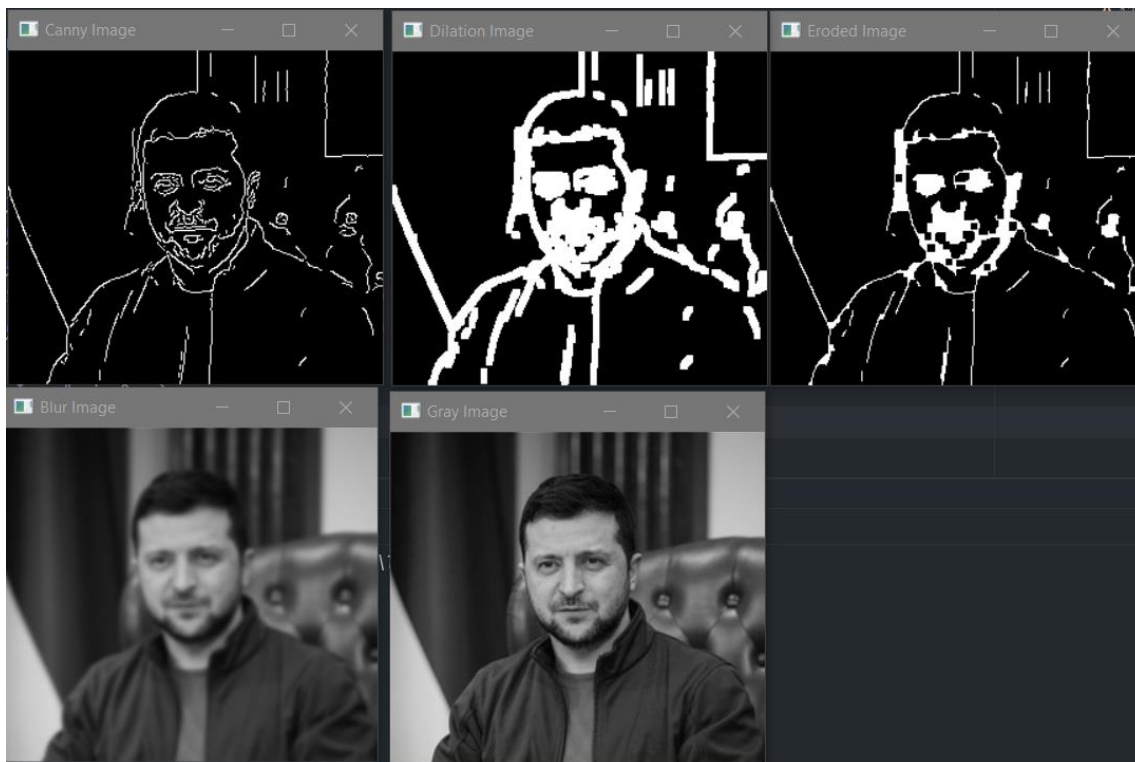


Рис.8.2 – Результати накладання фільтрів

Різні фільтри, надані OpenCV, надають можливість зміни кольорової гами зображення (зробити сірим), розмити, виявити кути, зробити більш різким тощо.

Завдання 3. Вирізання частини зображення

Лістинг коду файлу Task_3.py:

```
import cv2
import numpy as np

img = cv2.imread("Chyzhmotria.png")
print(img.shape)

imgResize = cv2.resize(img, (300, 200))
print(imgResize.shape)

imgCropped = img[0:300, 100:500]

cv2.imshow("Image", img)
cv2.imshow("Image Resize", imgResize)
cv2.imshow("Image Cropped", imgCropped)
cv2.waitKey(0)
```

		Чижмотря М.О.			ДУ «Житомирська політехніка».19.121.2.000 – Лр8	Арк.
		Пулеко І.В.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

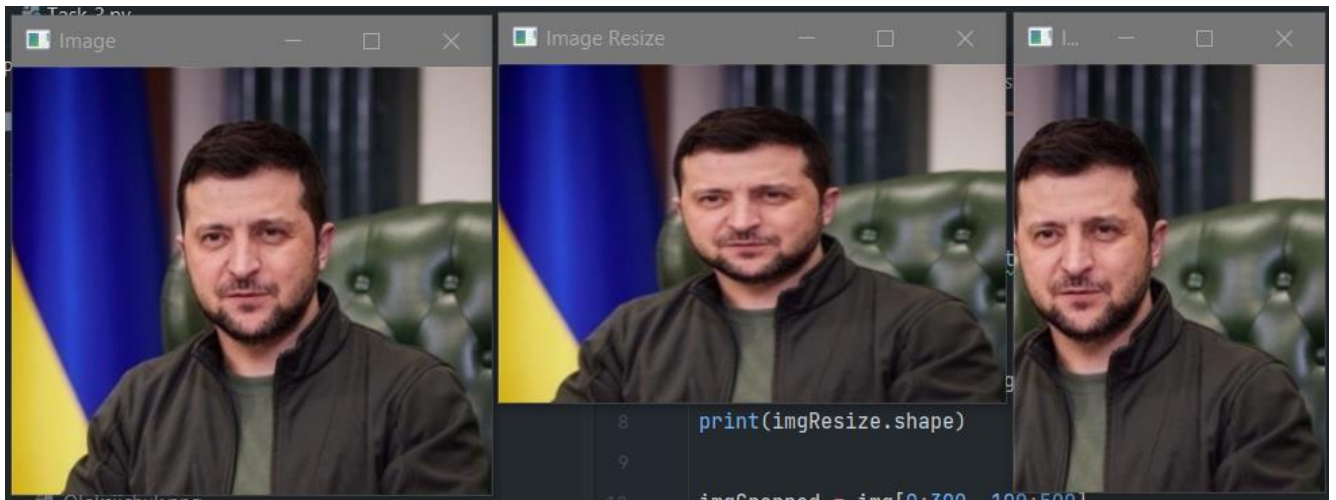


Рис.8.3 – Результат деформації зображень

Завдання 4. Розпізнавання обличчя на зображенні

Лістинг коду файлу Task_4.py:

```
import cv2

faceCascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
img = cv2.imread("Chyzhmotria.png")
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)
```

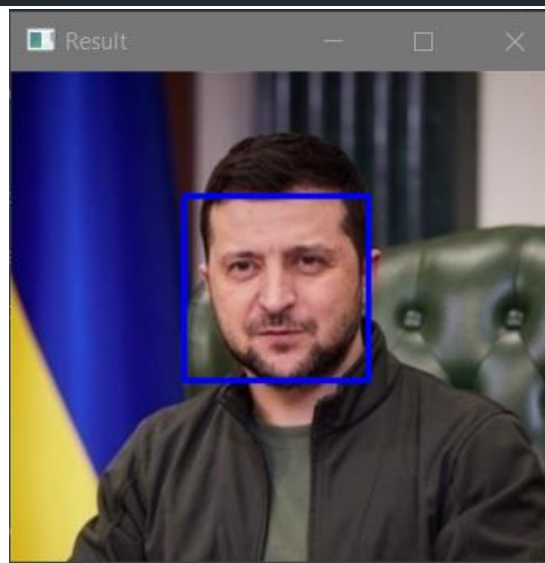


Рис.8.4 – Знайдене обличчя на зображенні

		Чижморя М.О.			ДУ «Житомирська політехніка».19.121.2.000 – Лр8	Арк.
		Пулеко І.В.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

Використання каскадного класифікатора із наданою бібліотекою OpenCV надає правильний результат зі знаходження обличчя. Область обличчя виділена синім прямокутником.

Завдання 5. Розпізнавання об'єктів на зображенні за допомогою методів зіставлення шаблонів

Лістинг коду файлу Task_5.py:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

def main(full_img="messi_full.jpg", face_img="messi_face.jpg"):
    img = cv2.imread(full_img, 0)
    img2 = img.copy()
    template = cv2.imread(face_img, 0)

    w, h = template.shape[::-1]
    methods = ["cv2.TM_CCOEFF", "cv2.TM_CCOEFF_NORMED", "cv2.TM_CCORR",
               "cv2.TM_CCORR_NORMED", "cv2.TM_SQDIFF", "cv2.TM_SQDIFF_NORMED"]

    for meth in methods:
        img = img2.copy()
        method = eval(meth)

        res = cv2.matchTemplate(img, template, method)
        min_val, max_val, min_loc, max_loc = cv2.minMaxLoc(res)

        if method in [cv2.TM_SQDIFF, cv2.TM_SQDIFF_NORMED]:
            top_left = min_loc
        else:
            top_left = max_loc

        bottom_right = (top_left[0] + w, top_left[1] + h)

        cv2.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()

if __name__ == "__main__":
    main('Chyzhmotria.png', 'Chyzhmotria_face.png')

plt.show()
```

		Чижмотря М.О.			ДУ «Житомирська політехніка».19.121.2.000 – Лр8	Арк.
		Пулеко І.В.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

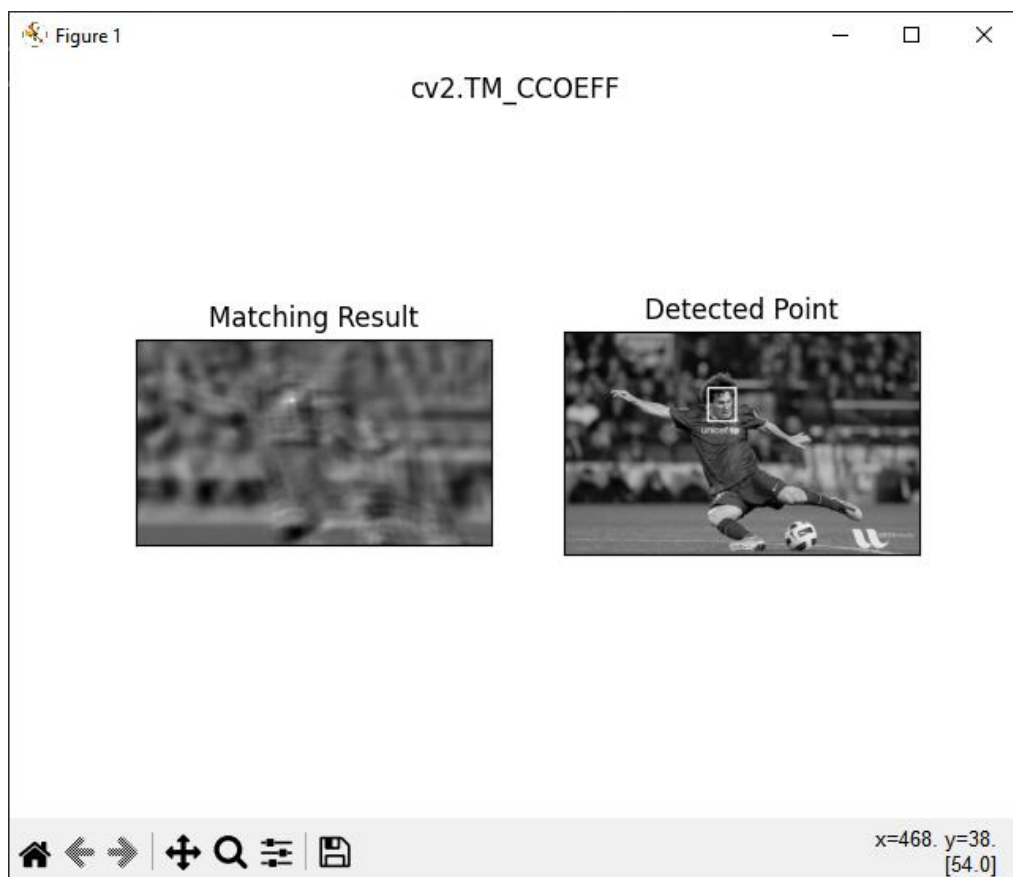


Рис.8.5 – Робота методу TM_CCOEFF

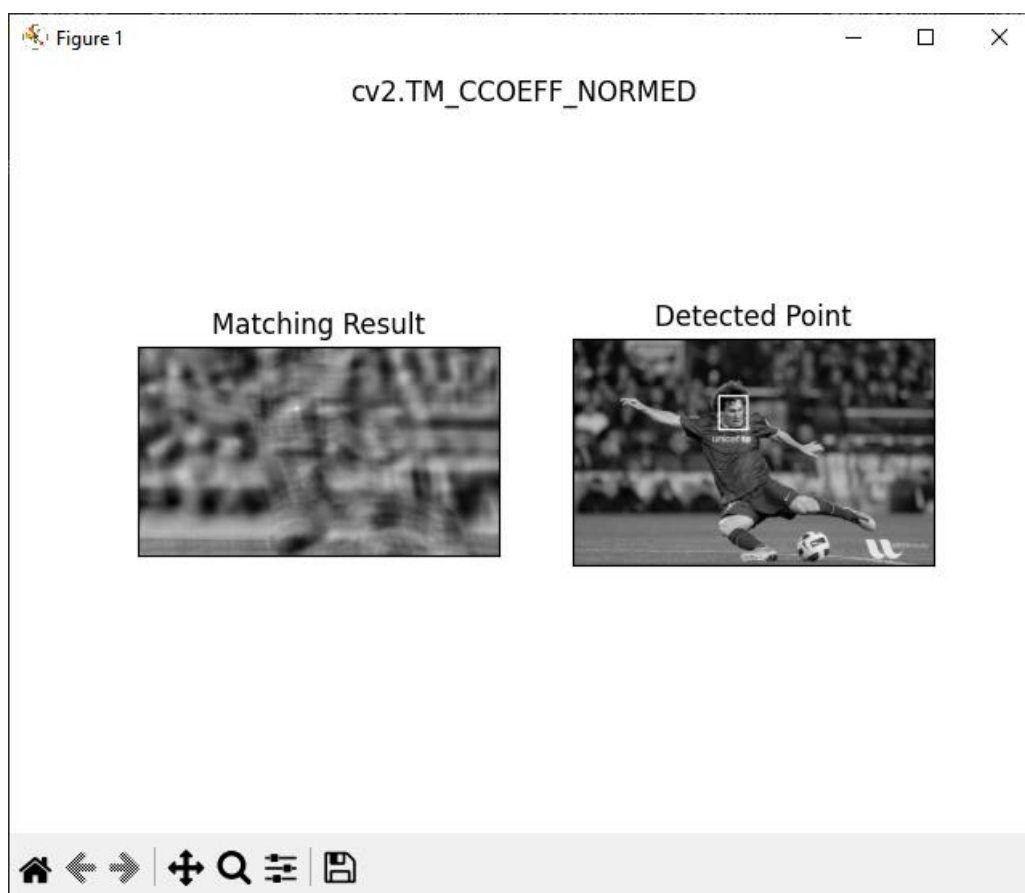


Рис.8.6 – Робота методу TM_CCOEFF_NORMED

		Чижмотря М.О.			ДУ «Житомирська політехніка».19.121.2.000 – Лр8	Арк.
		Пулеко І.В.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

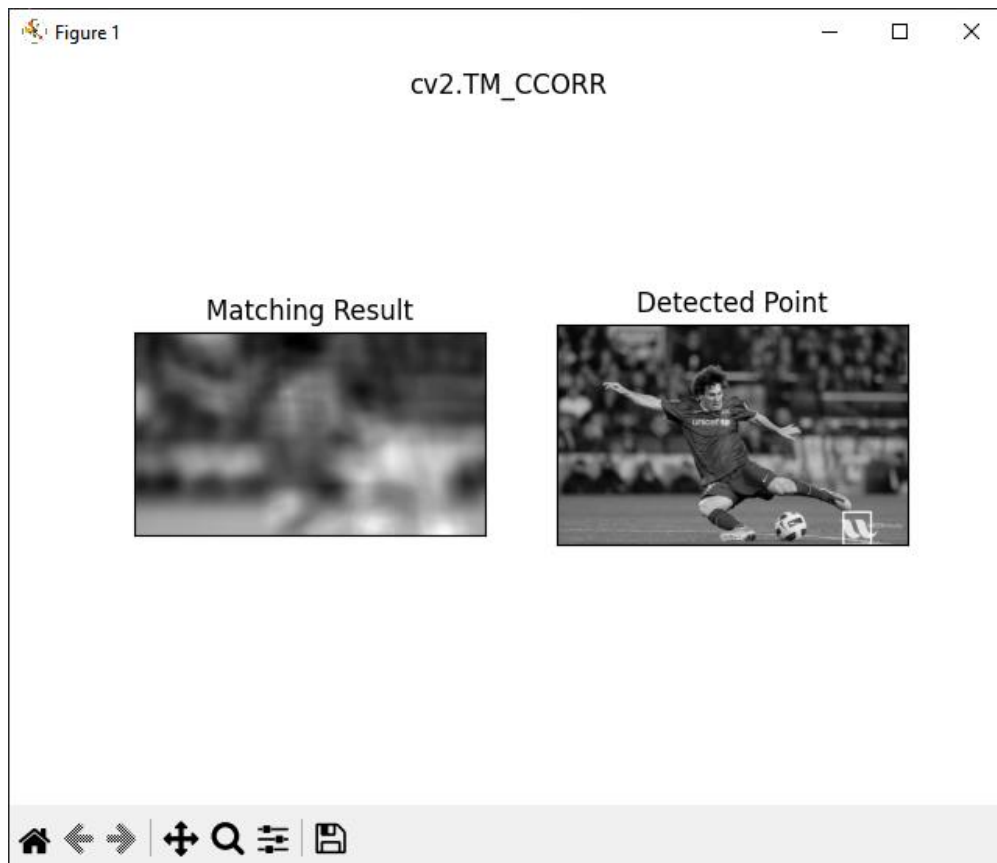


Рис.8.7 – Робота методу TM_CCORR

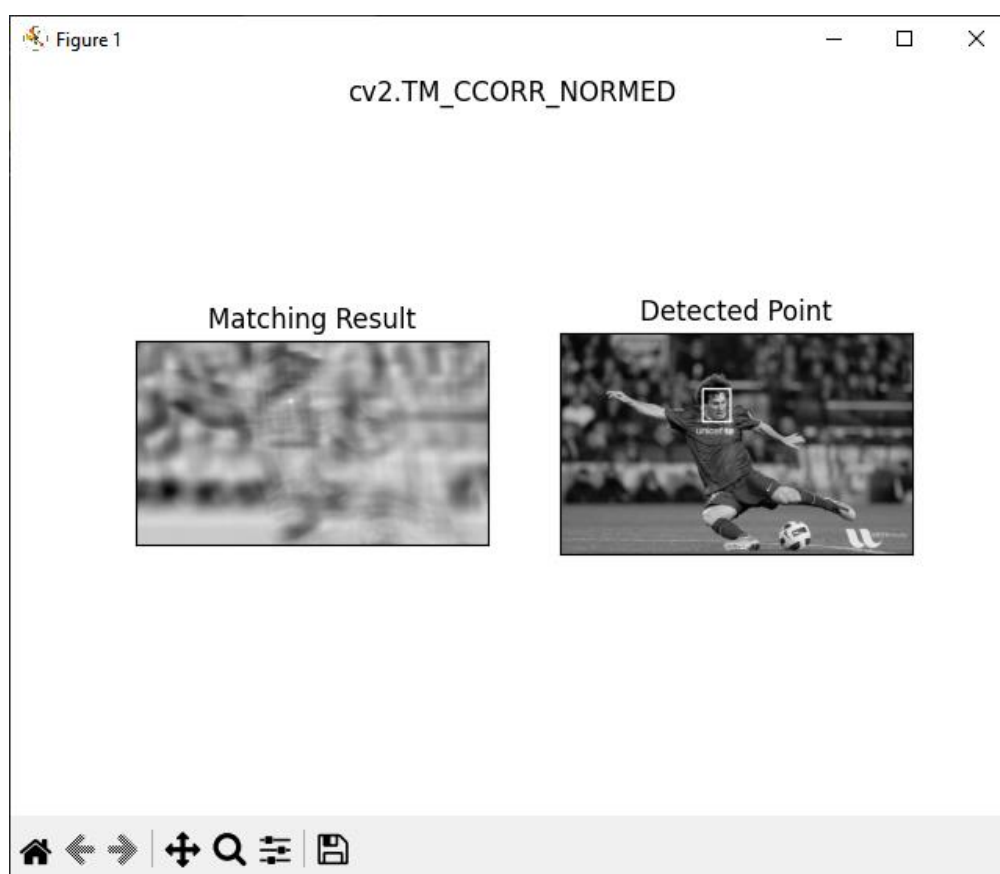


Рис.8.8 – Робота методу TM_CCORR_NORMED

		Чижморя М.О.			ДУ «Житомирська політехніка».19.121.2.000 – Лр8	Арк.
		Пулеко І.В.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

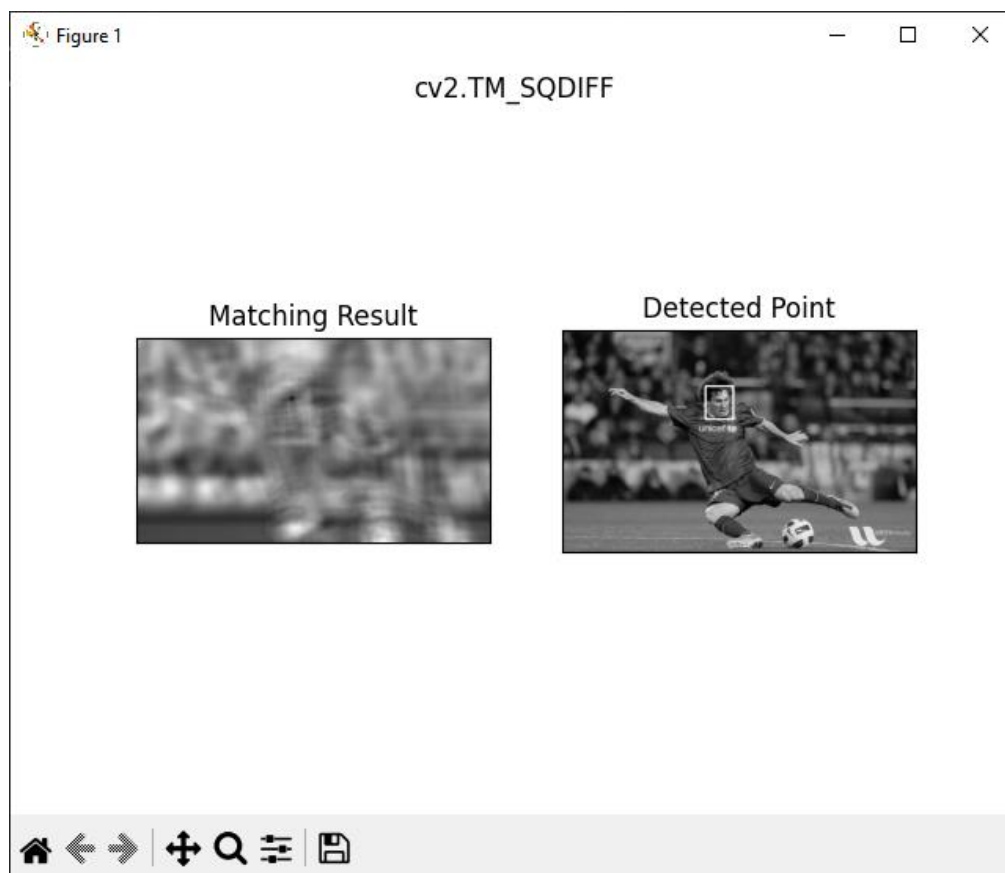


Рис.8.9 – Робота методу TM_SQDIFF

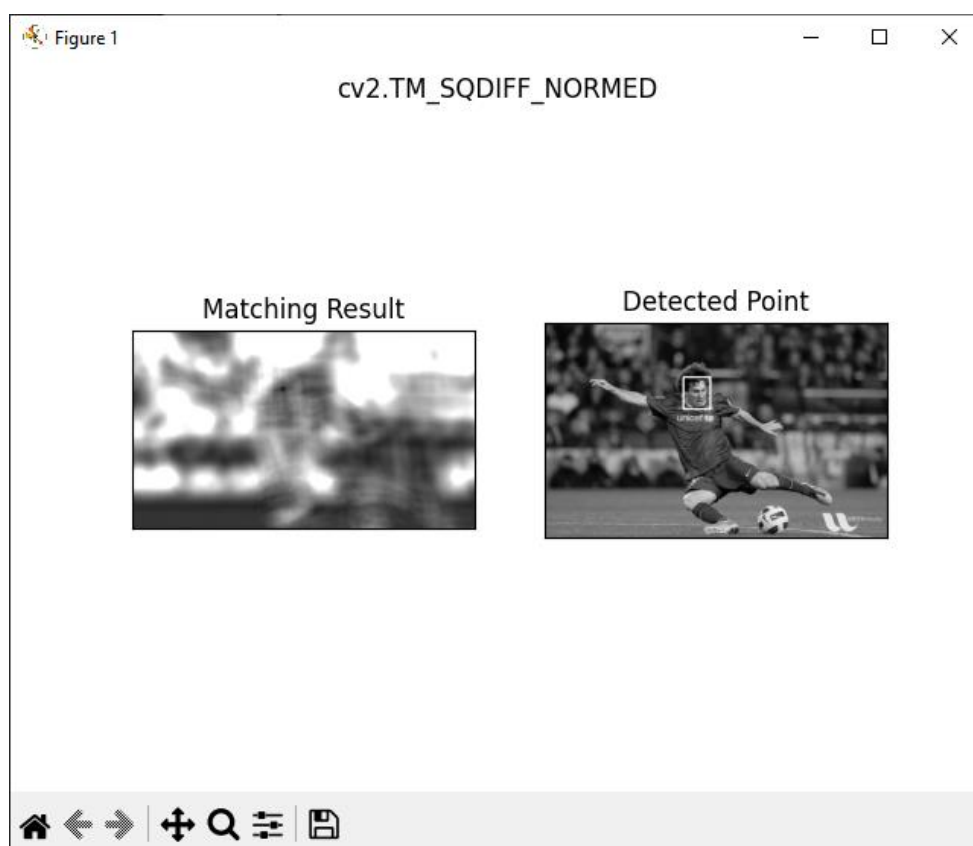


Рис.8.10 – Робота методу TM_SQDIFF_NORMED

		Чижморя М.О.			ДУ «Житомирська політехніка».19.121.2.000 – Лр8	Арк.
		Пулеко І.В.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

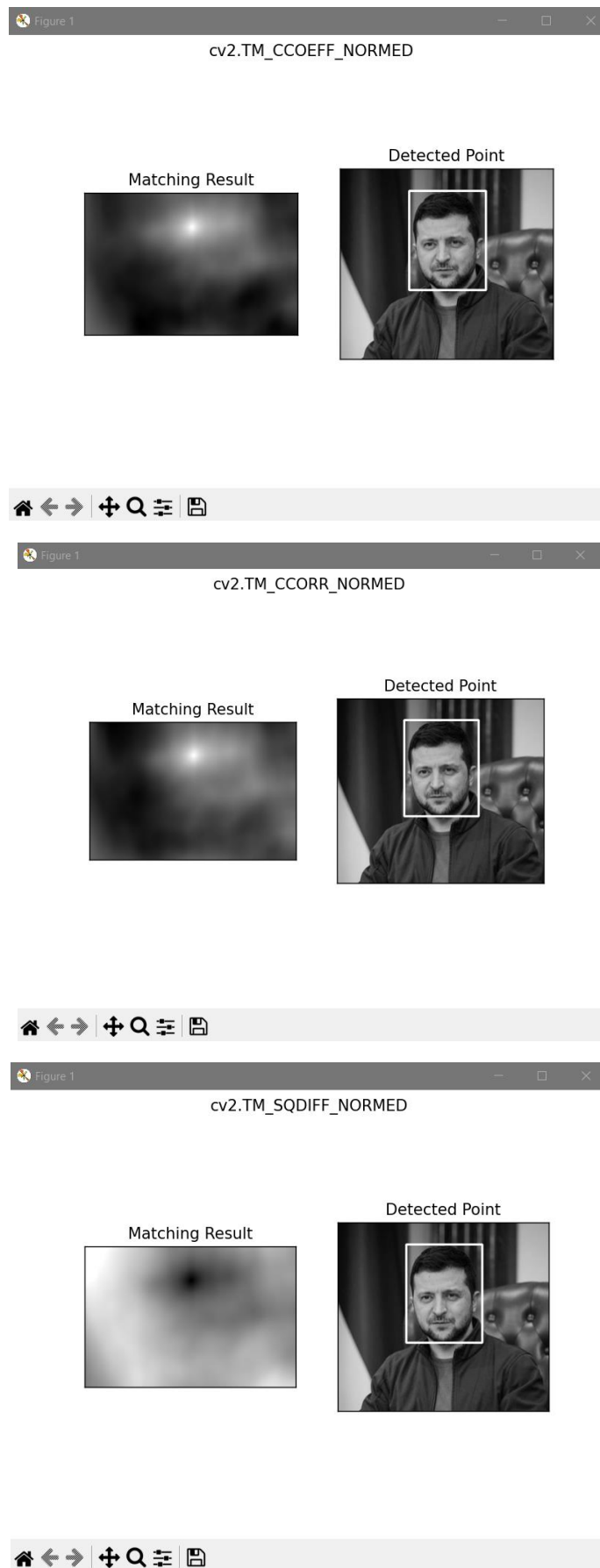


Рис.8.11 – Робота методів над власним зображенням

		Чижмотря М.О.			ДУ «Житомирська політехніка».19.121.2.000 – Лр8	Арк.
		Пулеко І.В.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

Робота більшості методів пошуку обличчя за шаблоном є ефективною, окрім методу TM_CCORR. Роботу решти методів, які не вказано, на власному зображенні, мають ідентичний результат зі вказаними.

Завдання 6. Сегментація зображення алгоритмом водорозподілу

Лістинг коду файлу Task_6.py:

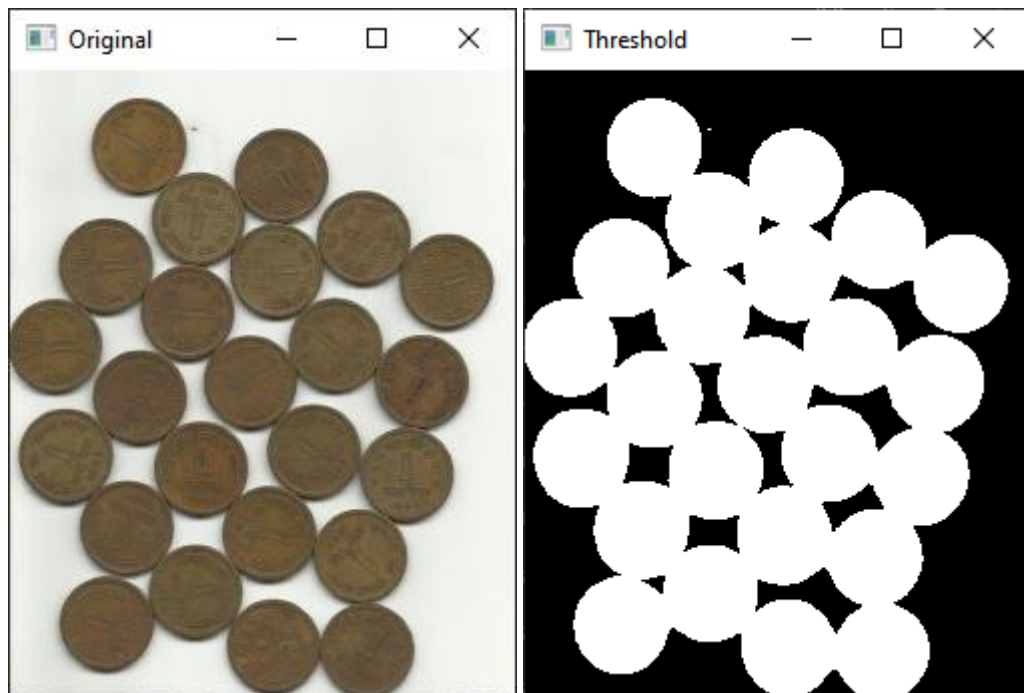
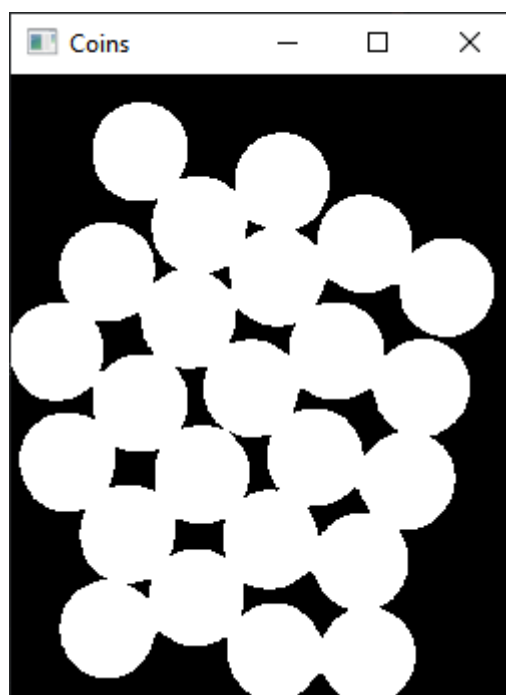


Рис.8.12 – Оригінальне та чорно-біле зображення монет



		Чижмотря М.О.			ДУ «Житомирська політехніка».19.121.2.000 – Лр8	Арк.
		Пулеко І.В.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

Рис.8.13 – Знешумлене зображення монет



Рис.8.14 – Помічені монети

Завдання 7. Дослідження перетворень зображення

Лістинг коду файлу Task_7.py:

```
import cv2
import numpy as np
from scipy import ndimage as ndi
from skimage.feature import peak_local_max
from skimage.segmentation import watershed
import matplotlib.pyplot as plt

img = cv2.imread('coins_2.jpg')
filtered = cv2.pyrMeanShiftFiltering(img, 20, 40)

gray = cv2.cvtColor(filtered, cv2.COLOR_BGR2GRAY)
_, thresh_img = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY_INV |
cv2.THRESH_OTSU)

contours, _ = cv2.findContours(thresh_img, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
contours_filtered = []
for con in contours:
    area = cv2.contourArea(con)
    if area < 1000:
        contours_filtered.append(con)

cv2.drawContours(thresh_img, contours_filtered, -1, 255, -1)
dist = ndi.distance_transform_edt(thresh_img)
dist_copy = dist.copy()

local_max = peak_local_max(dist, indices=False, min_distance=20,
labels=thresh_img)
markers = ndi.label(local_max, structure=np.ones((3, 3)))[0]

watershed_img = watershed(-dist, markers, mask=thresh_img)
titles = ['Original image', 'Binary Image', 'Watershed']
```

		Чижмотря М.О.			ДУ «Житомирська політехніка».19.121.2.000 – Лр8	Арк.
		Пулеко І.В.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

```

images = [img, thresh_img, watershed_img]

fig = plt.gcf()
fig.set_size_inches(8, 6)

for i, img in enumerate(images):
    plt.subplot(3, 1, i + 1)
    plt.imshow(img, "jet" if i == 2 else "gray")
    plt.title(titles[i])
    plt.xticks([], plt.yticks([]))

plt.show()

```

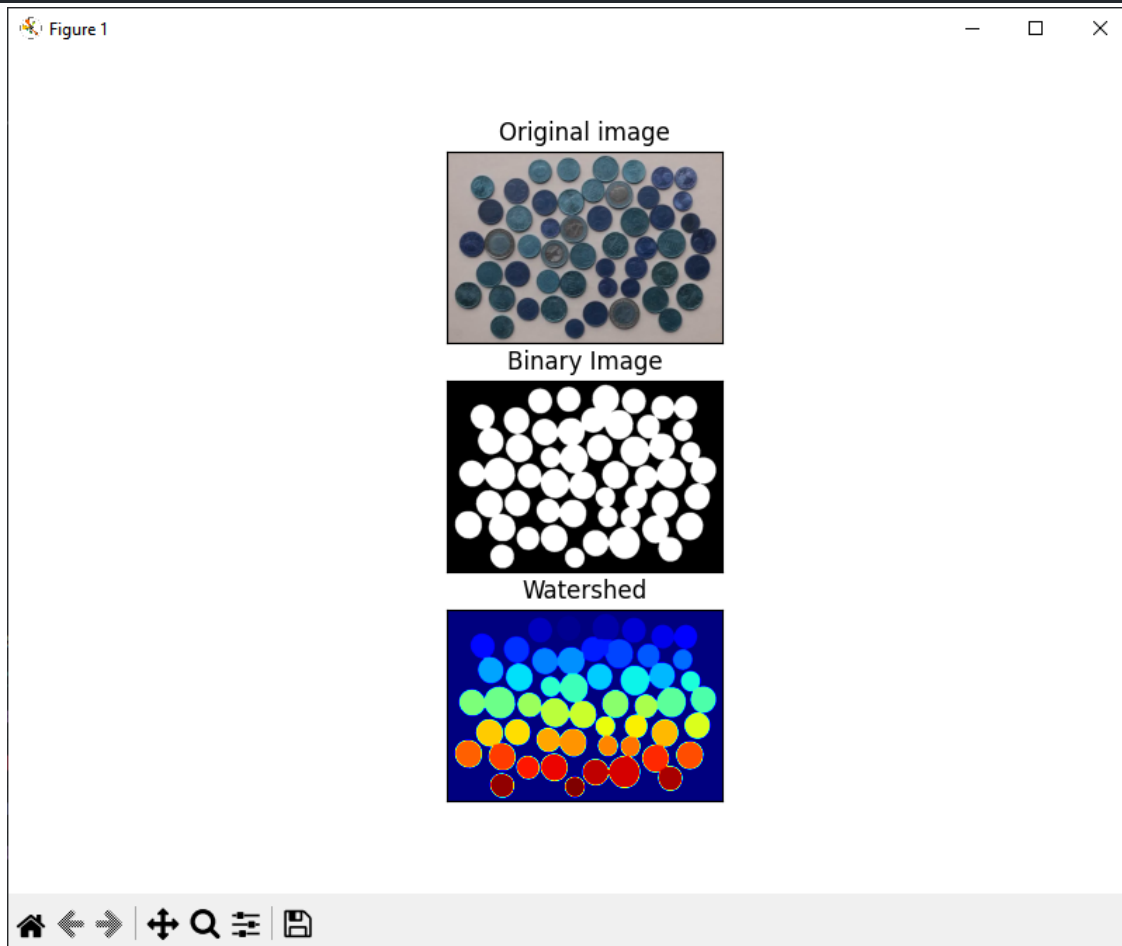


Рис.8.15 – Результат пошуку монет на іншій фотографії

Розрізнення монет по їх ціні виявилось на даний момент неможливим з урахуванням часу, що є в наявності для виконання завдання, та зображення кутів, яке не дає можливості знайти точні співпадіння для класифікації монет.

Висновок: протягом виконання завдань лабораторної роботи було отримано навички з обробки зображення використовуючи бібліотеку OpenCV та мову програмування Python.

Github: https://github.com/mikrorobot/Python_AI

		Чижмотря М.О.			ДУ «Житомирська політехніка».19.121.2.000 – Лр8	Арк.
		Пулеко І.В.				12
Змн.	Арк.	№ докум.	Підпис	Дата		