

## 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:**

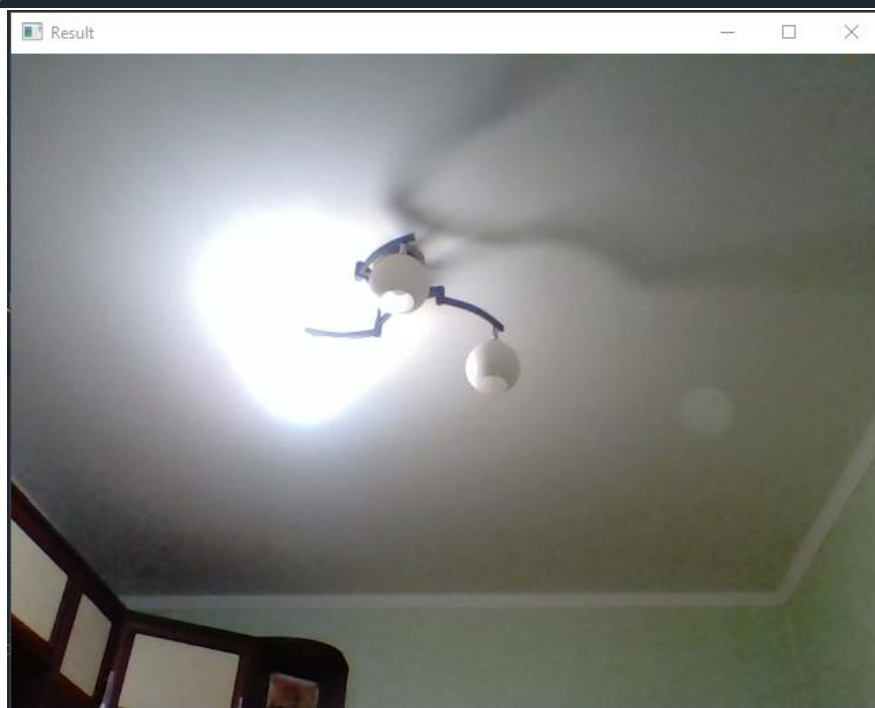
#### Завдання 1. Завантаження зображень та відео в *OpenCV*.

Лістинг коду файлу LR\_8\_task\_1.py:

```
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

img = cv2.imread("kravchenko.jpg")
cv2.imshow("kravchenko",img)
cv2.waitKey(0)
```



					ДУ «Житомирська політехніка».22.121.08.000 – Лр8			
Змн.	Арк.	№ докум.	Підпис	Дата	Звіт з лабораторної роботи	Літ.	Арк.	Аркушів
Розроб.		Кравченко О.І.						
Перевір.		Філіпов В.О.					1	9
Керівник						ФІКТ Гр. ІПЗ-19-2[2]		
Н. контр.								
Зав. каф.								

Рис.1 – Успішно підключена камера.

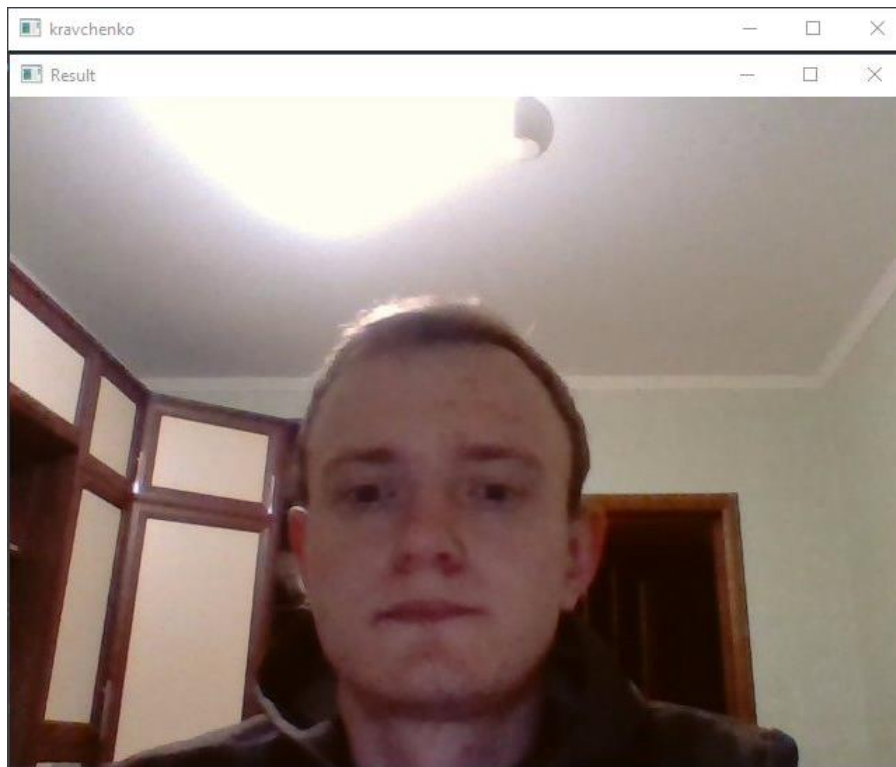


Рис.2 – Успішно завантажене фото.

## Завдання 2. Дослідження перетворень зображення.

Лістинг коду файлу LR\_8\_task\_2.py:

```
import cv2
import numpy as np

img = cv2.imread("kravchenko.jpg")
kernel = np.ones((5,5),np.uint8)
# is used to convert an image from one color space to another
imgGray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
# Blurred image
imgBlur = cv2.GaussianBlur(imgGray,(7,7),0)
# used to detect the edges in an image.
imgCanny = cv2.Canny(img,150,200)
# used to apply the dilation operation on the given image with the specified kernel
imgDialation = cv2.dilate(imgCanny,kernel,iterations=1)
# method is used to perform erosion on the image.
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)
```

		Кравченко О.І.			Житомирська політехніка.22.121.08.000 – Лр8	Арк.
		Філіпов В.О.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

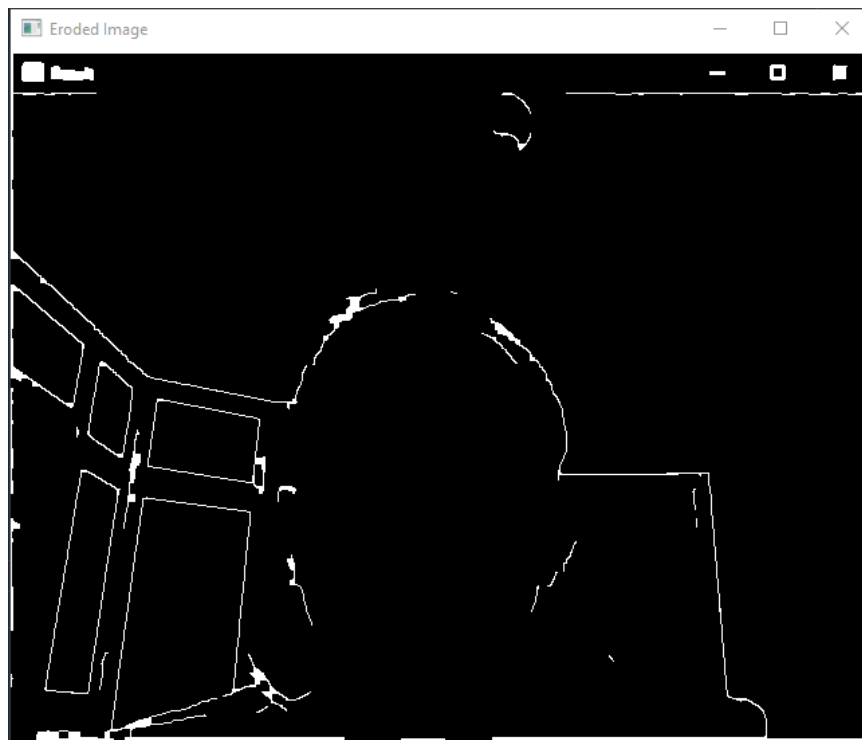


Рис.3 – Результат виконання програми.

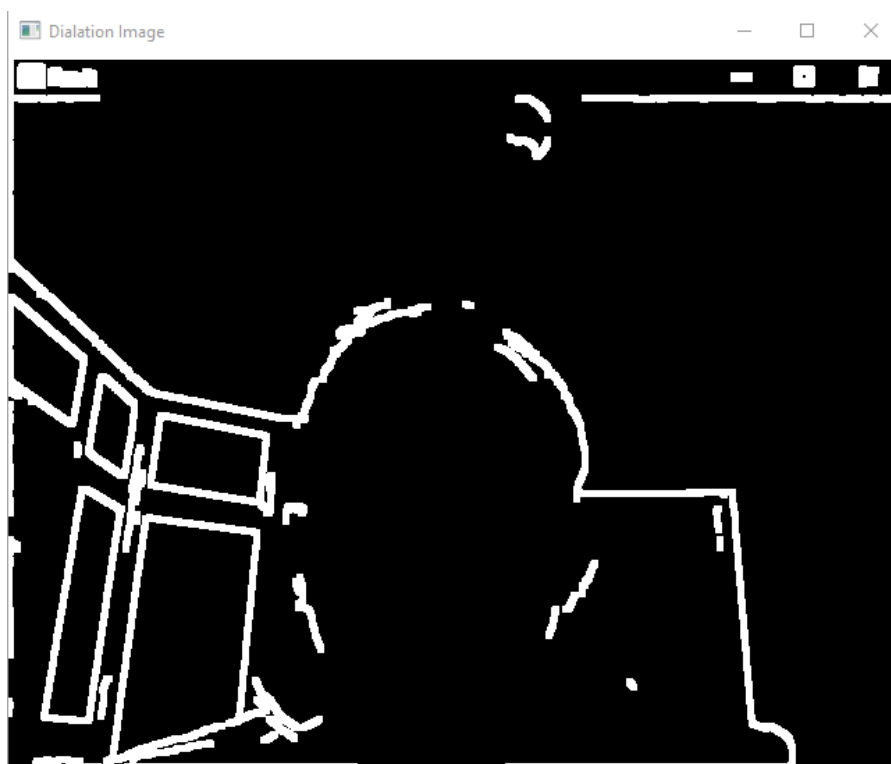


Рис.4 – Результат виконання програми.

		Кравченко О.І.			Житомирська політехніка.22.121.08.000 – Лр8	Арк.
		Філіпов В.О.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

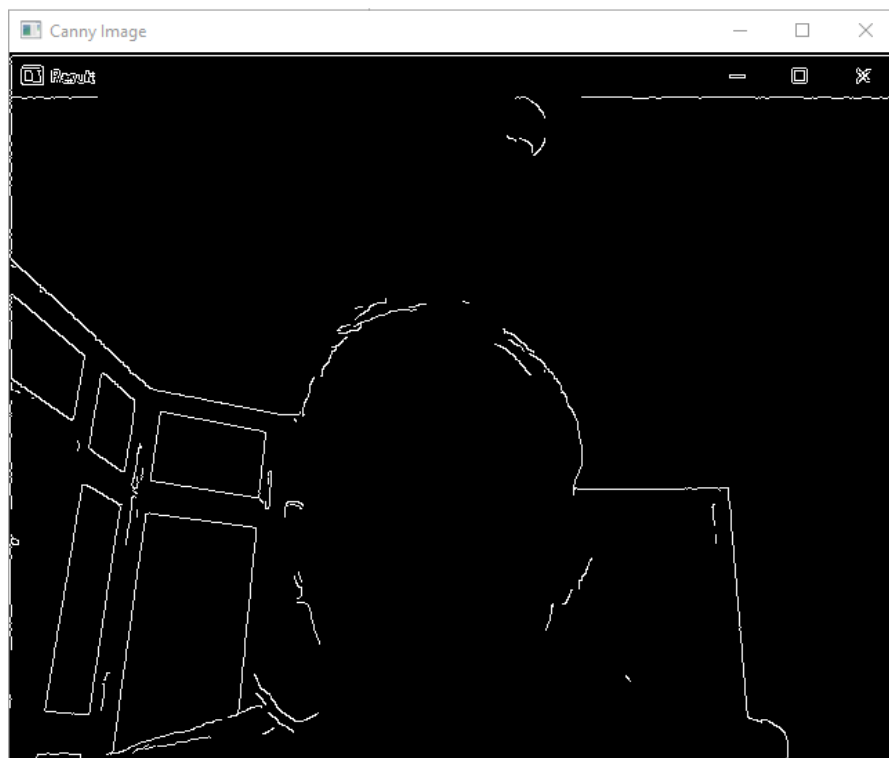


Рис.5 – Результат виконання програми.

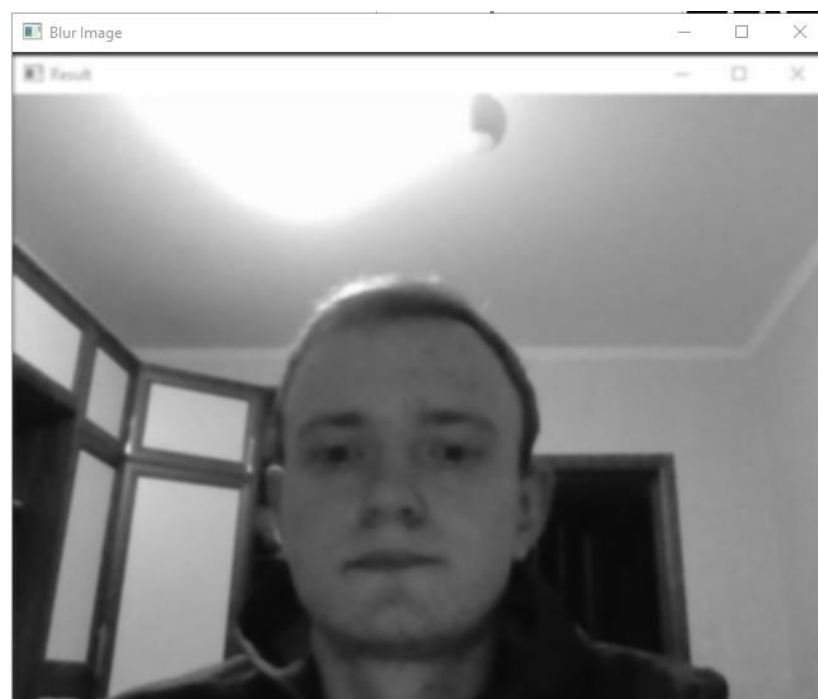


Рис.6 – Результат виконання програми.

		Кравченко О.І.			Житомирська політехніка.22.121.08.000 – Лр8	Арк.
		Філіпов В.О.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

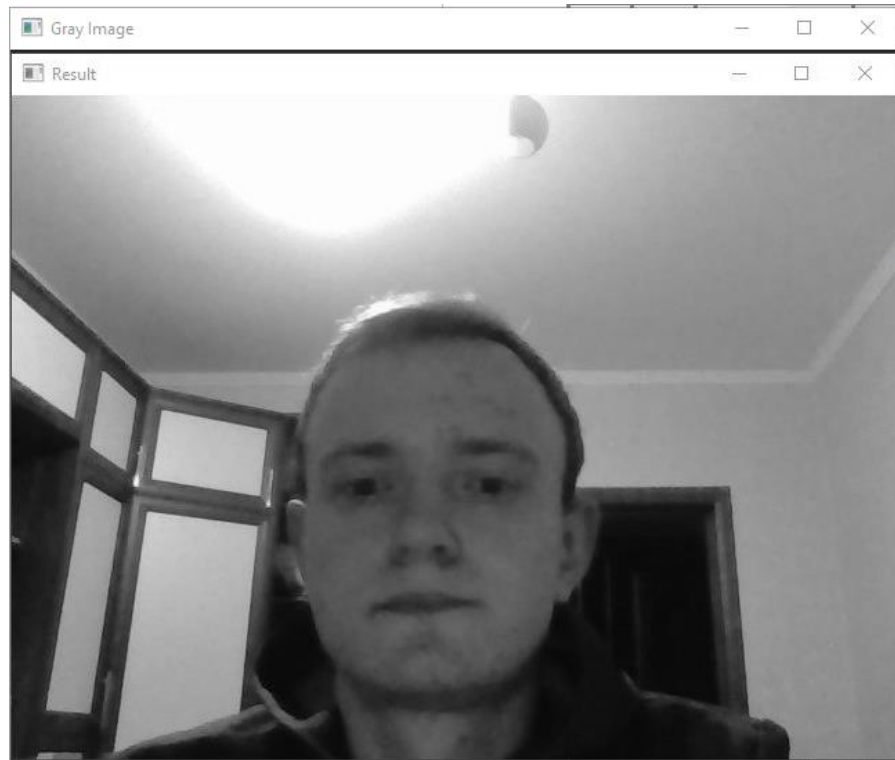


Рис.7 – Результат виконання програми.

Метод `cvtColor` - використовується для перетворення зображення з одного кольорового простору на інший.

Метод `dilate()` приймає два вхідні параметри, один із яких є нашим вхідним зображенням; другий називається структуруючим елементом чи ядром, яке визначає характер операції. Розширення зображення збільшує об'єкт.

Метод `erode` - метод використовується для виконання розмивання зображення.

Метод `GaussianBlur` - розмите зображення.

Метод `Canny` - використовується для визначення країв зображення.

### Завдання 3. Вирізання частини зображення.

Лістинг коду файлу `LR_8_task_3.py`:

```
import cv2

img = cv2.imread("kravchenko.jpg")
print(img.shape)

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

imgCropped = img[46:119, 352:495]

cv2.imshow("Image", img)
cv2.imshow("Image Resize",imgResize)
```

		Кравченко О.І.			Житомирська політехніка.22.121.08.000 – Лр8	Арк.
		Філіпов В.О.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

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



Рис.8 – Результат виконання програми.

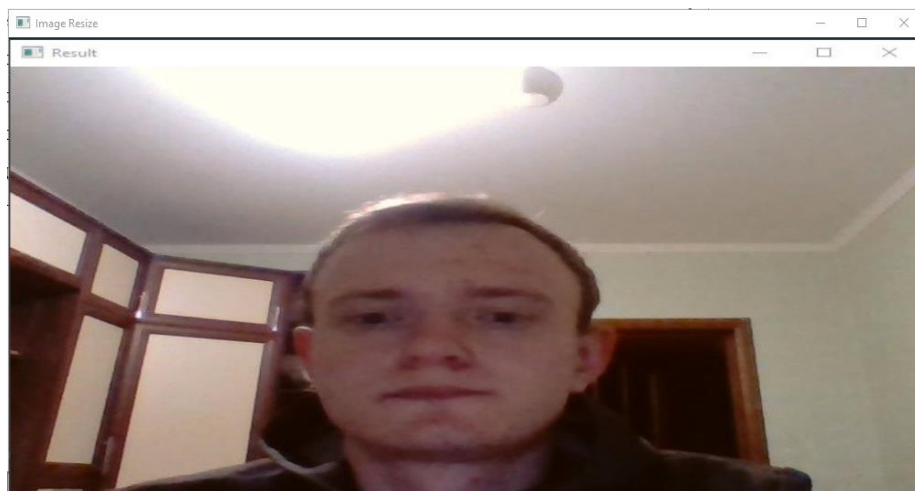


Рис.9 – Результат виконання програми.

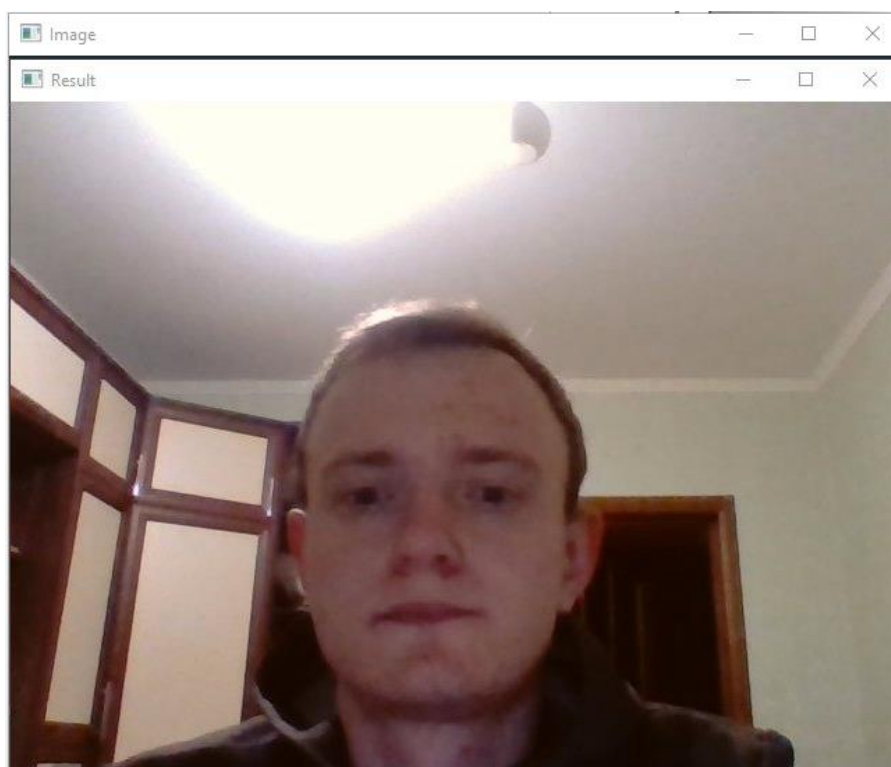


Рис.10 – Результат виконання програми.

#### Завдання 4. Розпізнавання обличчя на зображенні.

		Кравченко О.І.			Житомирська політехніка.22.121.08.000 – Лр8	Арк.
		Філіпов В.О.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

Лістинг коду файлу LR\_8\_task\_4.py:

```
import cv2

faceCascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
img = cv2.imread('kravchenko.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)
```

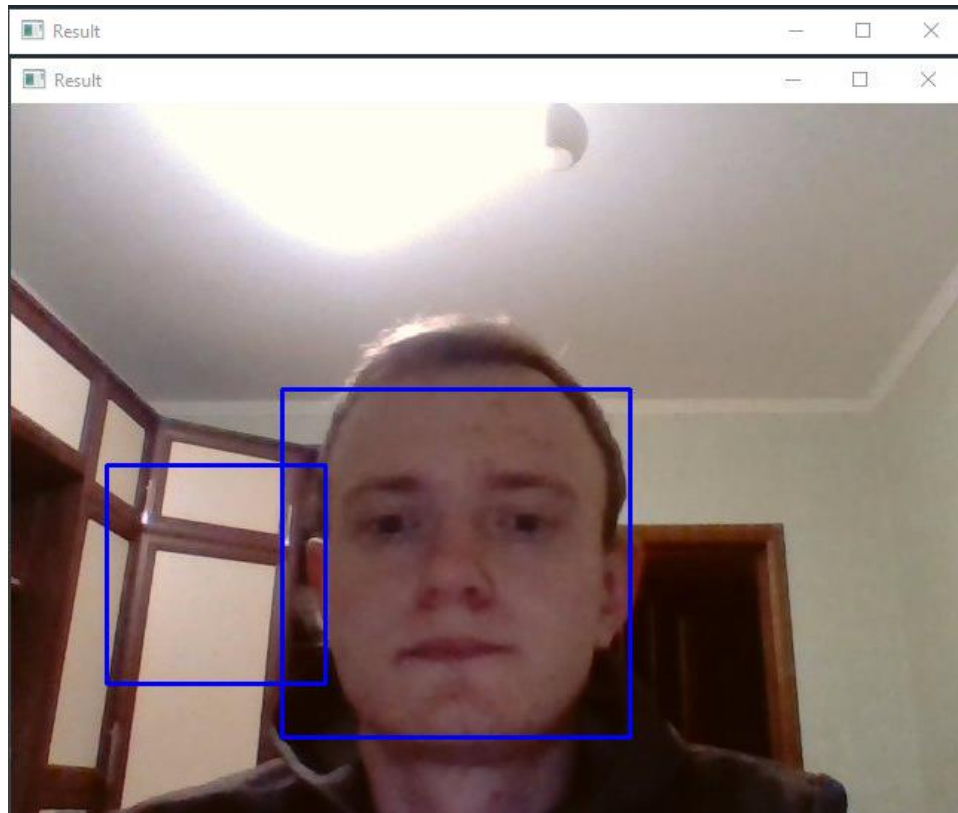


Рис.11 – Результат виконання програми.

### Завдання 5. Розпізнавання об'єктів на зображенні за допомогою методів зіставлення шаблонів (Template Matching).

Лістинг коду файлу LR\_8\_task\_5.py:

```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt

img = cv.imread('kravchenko.JPG', 0)
img2 = img.copy()
template = cv.imread('kravchenko_face.png', 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',
```

		Кравченко О.І.			Житомирська політехніка.22.121.08.000 – Лр8	Арк.
		Філіпов В.О.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        '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 method in [cv.TM_CCOEFF, cv.TM_CCOEFF_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()

```

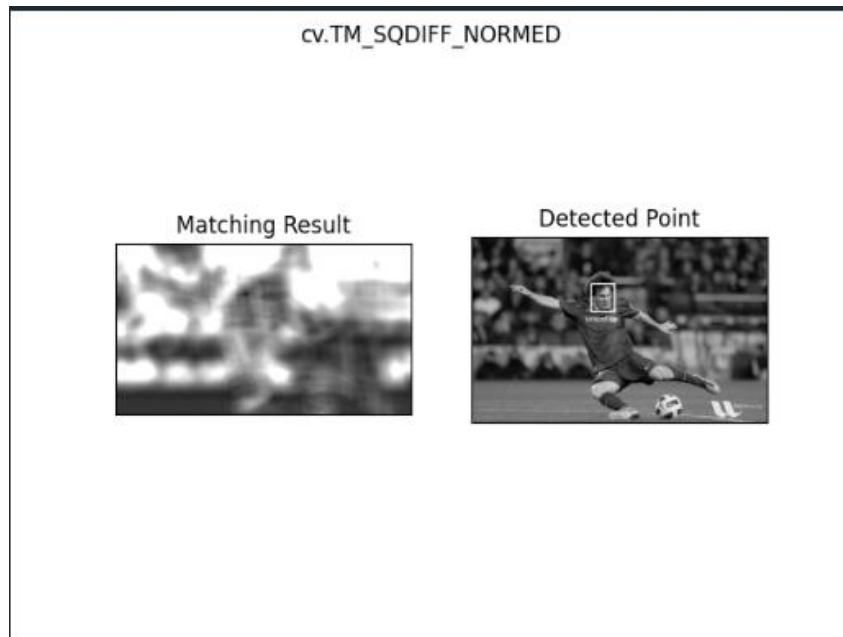


Рис.12 – Результат виконання програми.

		Кравченко О.І.			Житомирська політехніка.22.121.08.000 – Лр8	Арк.
		Філіпов В.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		8



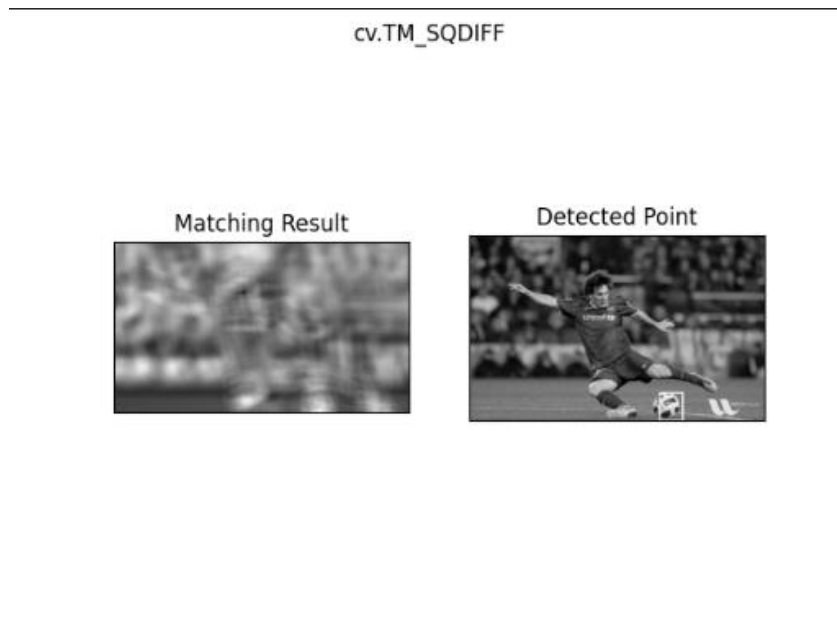


Рис.13 – Результат виконання програми.

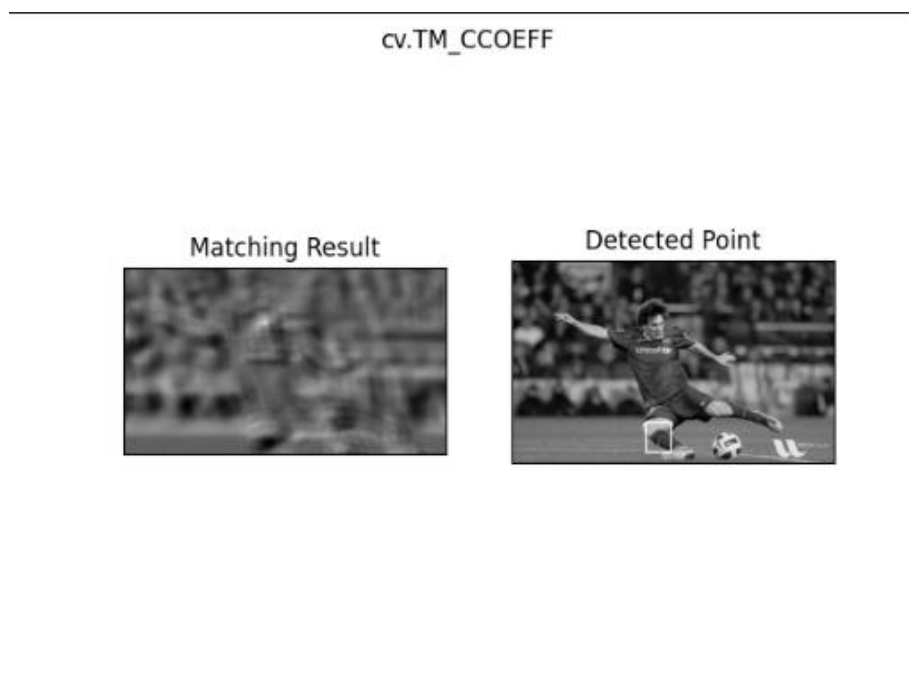


Рис.14 – Результат виконання програми.

		Кравченко О.І.			Житомирська політехніка.22.121.08.000 – Лр8	Арк.
		Філіпов В.О.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

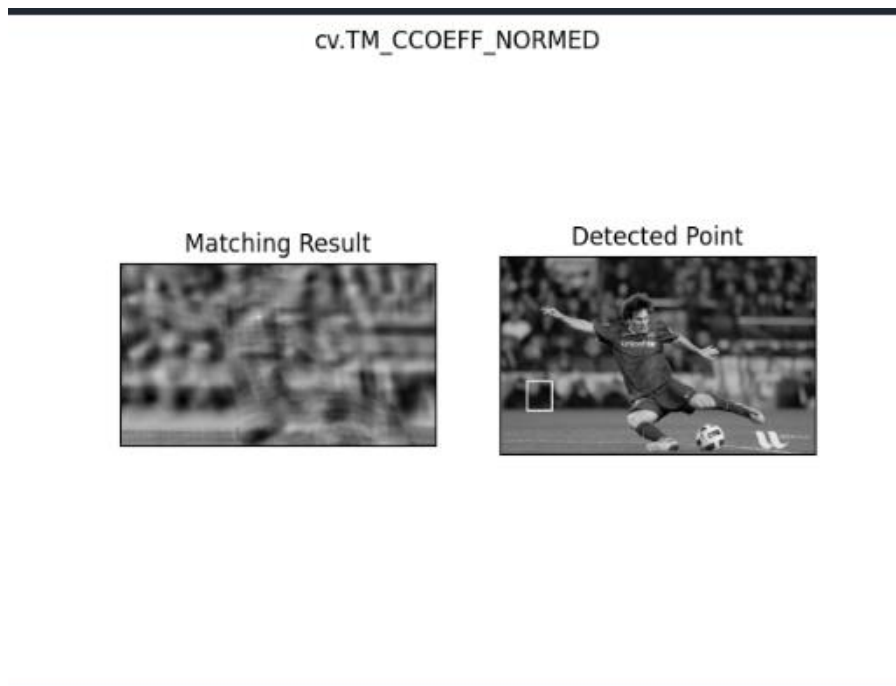


Рис.15 – Результат виконання програми.

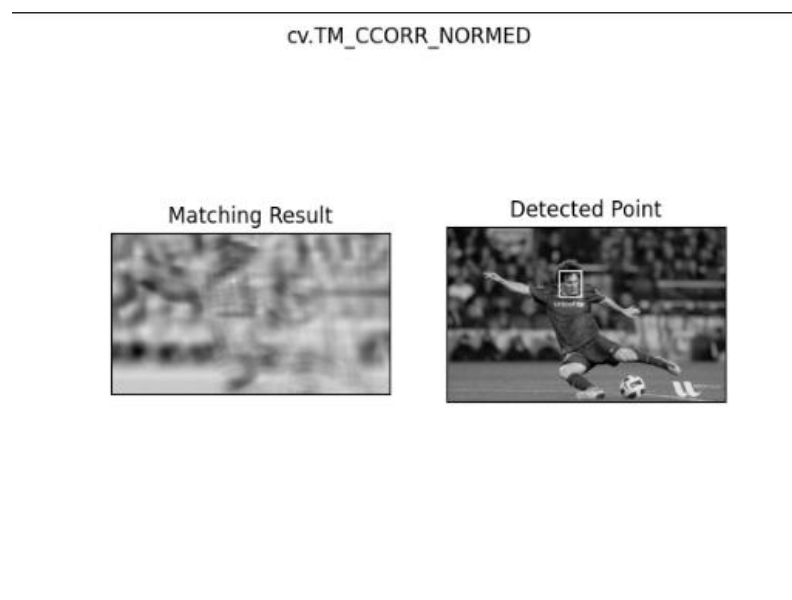


Рис.16 – Результат виконання програми.

		Кравченко О.І.			Житомирська політехніка.22.121.08.000 – Лр8	Арк.
		Філіпов В.О.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

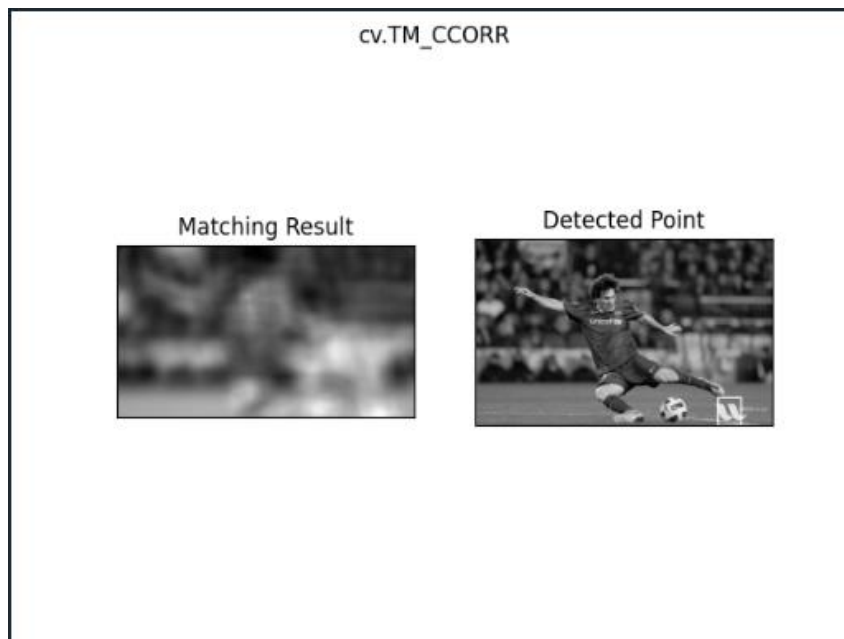


Рис.17 – Результат виконання програми.

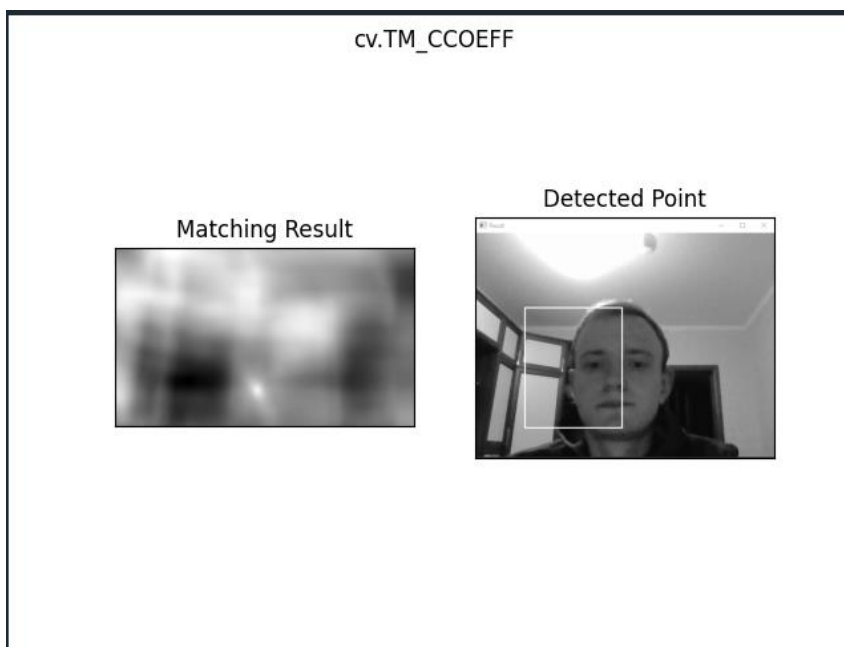


Рис.18 – Результат виконання програми.

cv.TM\_CCOEFF\_NORMED

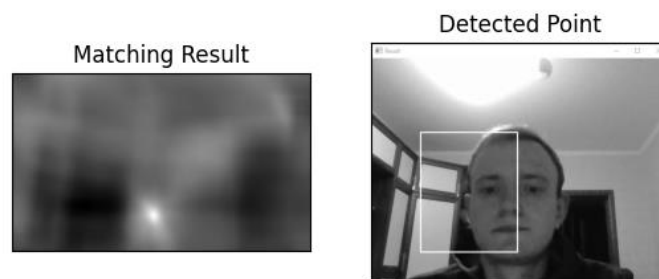


Рис.19 – Результат виконання програми.

cv.TM\_CCORR

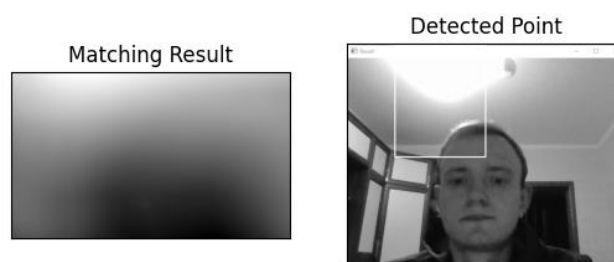


Рис.20 – Результат виконання програми.

		Кравченко О.І.			Житомирська політехніка.22.121.08.000 – Лр8	Арк.
		Філіпов В.О.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

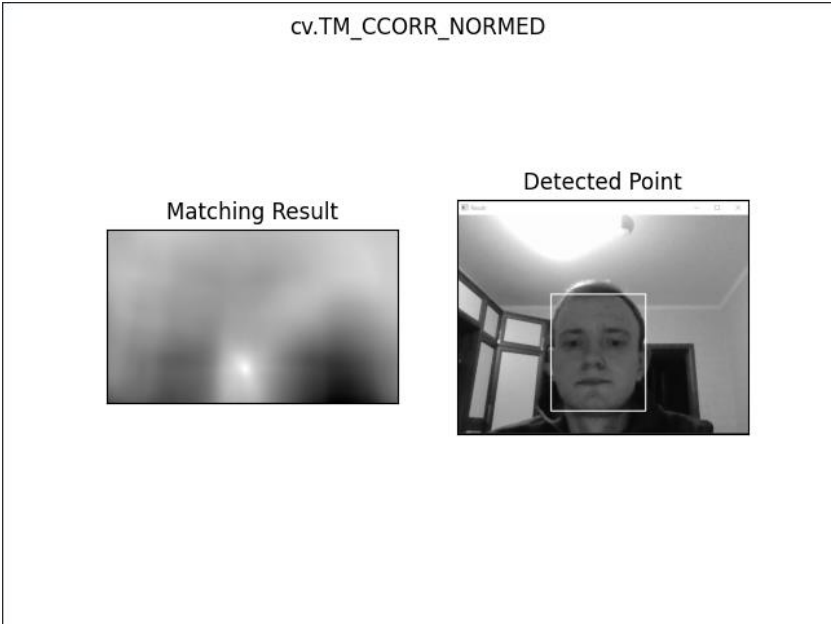


Рис.21 – Результат виконання програми.

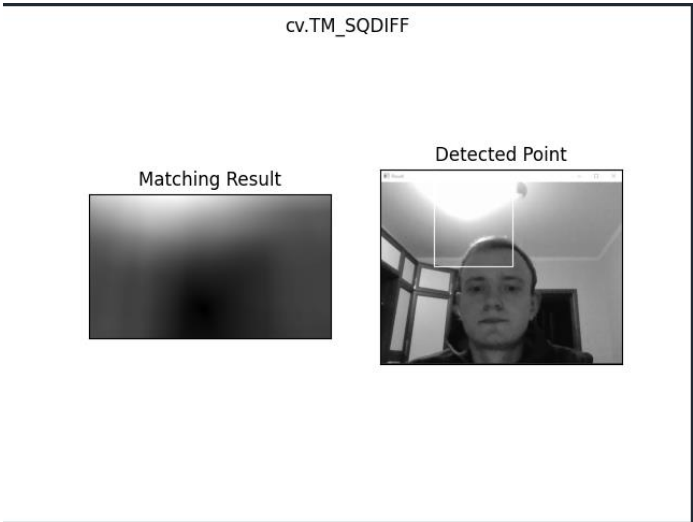


Рис.22 – Результат виконання програми.

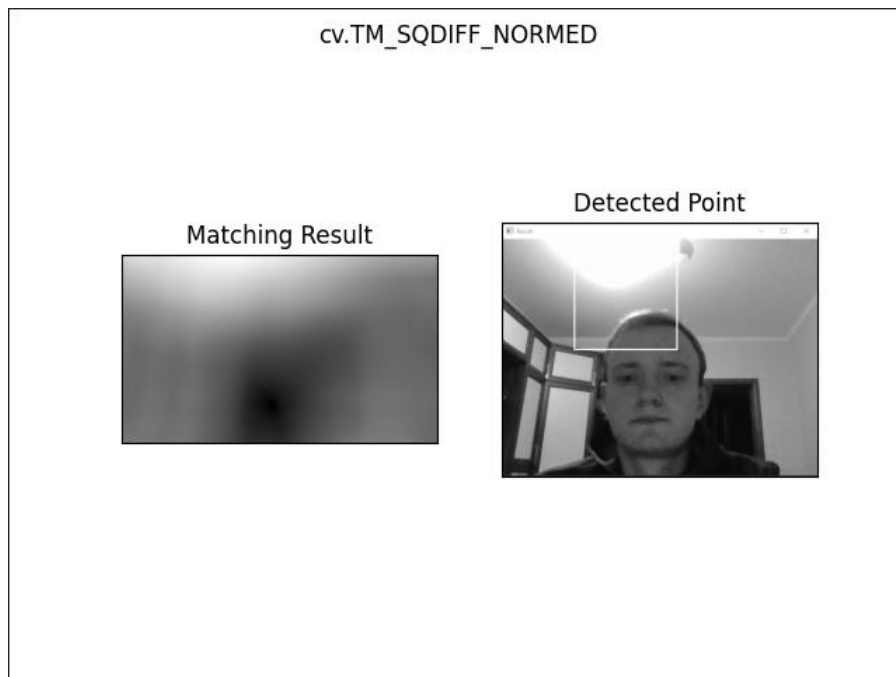


Рис.23 – Результат виконання програми.



Рис.24 – Результат виконання програми.

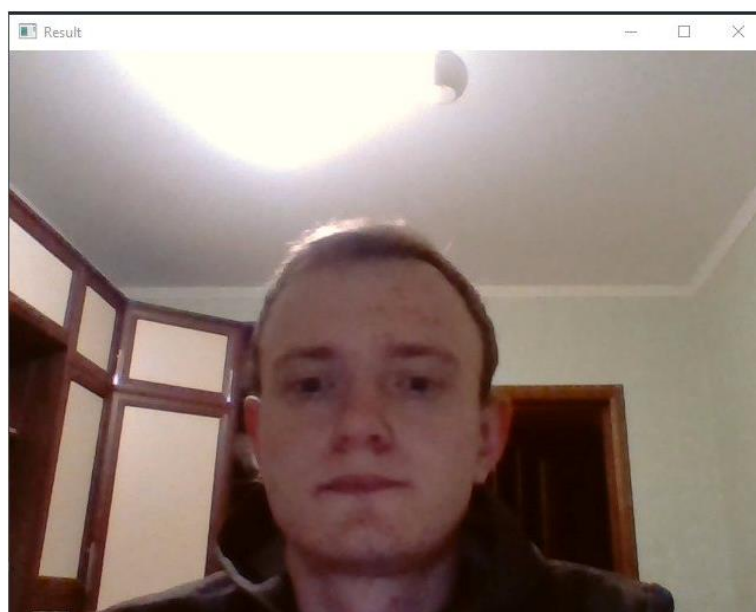


Рис.25 – Результат виконання програми.

		Кравченко О.І.			Житомирська політехніка.22.121.08.000 – Лр8	Арк.
		Філіпов В.О.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

## Завдання 6. Сегментація зображення алгоритмом водорозподілу.

Лістинг коду файлу LR\_8\_task\_6.py:

```
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 фоновна 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)
```

		Кравченко О.І.			Житомирська політехніка.22.121.08.000 – Лр8	Арк.
		Філіпов В.О.				15
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис.26 – Результат виконання програми.

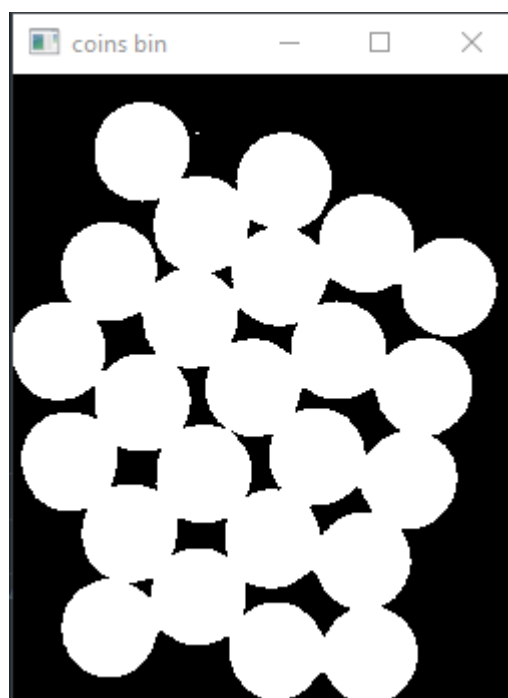


Рис.27 – Результат виконання програми.



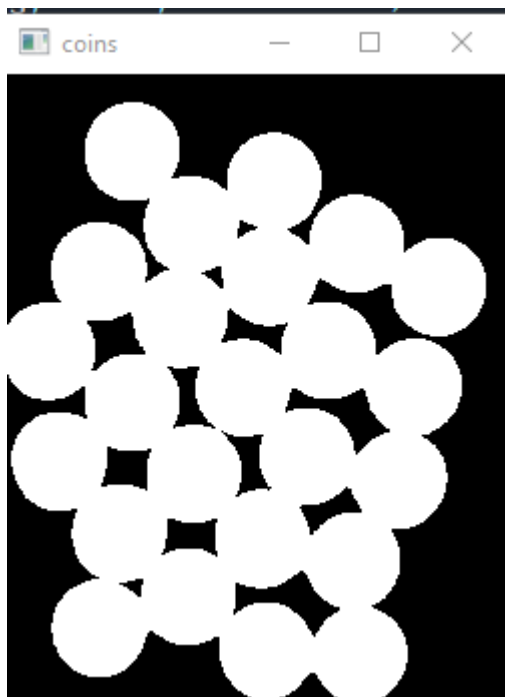


Рис.28 – Результат виконання програми.



Рис.29 – Результат виконання програми

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

Код програми зберігається у репозиторії за посиланням:  
[https://github.com/krava02/system\\_LR](https://github.com/krava02/system_LR)

		Кравченко О.І.			Житомирська політехніка.22.121.08.000 – Лр8	Арк.
		Філіпов В.О.				17
Змн.	Арк.	№ докум.	Підпис	Дата		