

## 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)
```

					ДУ «Житомирська політехніка». 23.121.17.000 – Лр8			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Скаковський В.О.			Звіт з лабораторної роботи №8		Літ.	Арк.
Перевір.		Голенко М.Ю.						
Керівник							1	12
Н. контр.							ФІКТ Гр. ІПЗ-20-1	
Зав. каф.								



Рис. 1.1 – Результат виконання програми

## Завдання 2: Дослідження перетворень зображення.

```
import cv2
import numpy as np

img = cv2.imread("skakovskiy.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)
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)
```

		Скаковський В.О			ДУ «Житомирська політехніка». 23.121.17.000 – Лр8	Арк.
		Голенко М.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		2



Рис. 2.1 – Результат виконання програми



Рис. 2.2 – Результат виконання програми

		Скаковський В.О			ДУ «Житомирська політехніка».23.121.17.000 – Лр8	Арк.
		Голенко М.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		3



Рис. 2.3 – Результат виконання програми



Рис. 2.4 – Результат виконання програми

		Скаковський В.О			ДУ «Житомирська політехніка».23.121.17.000 – Лр8	Арк.
		Голенко М.Ю.				4
Змн.	Арк.	№ докум.	Підпис	Дата		



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)
```

		Скаковський В.О			ДУ «Житомирська політехніка».23.121.17.000 – Лр8	Арк.
		Голенко М.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		5



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)
```

		Скаковський В.О			ДУ «Житомирська політехніка».23.121.17.000 – Лр8	Арк.
		Голенко М.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		6



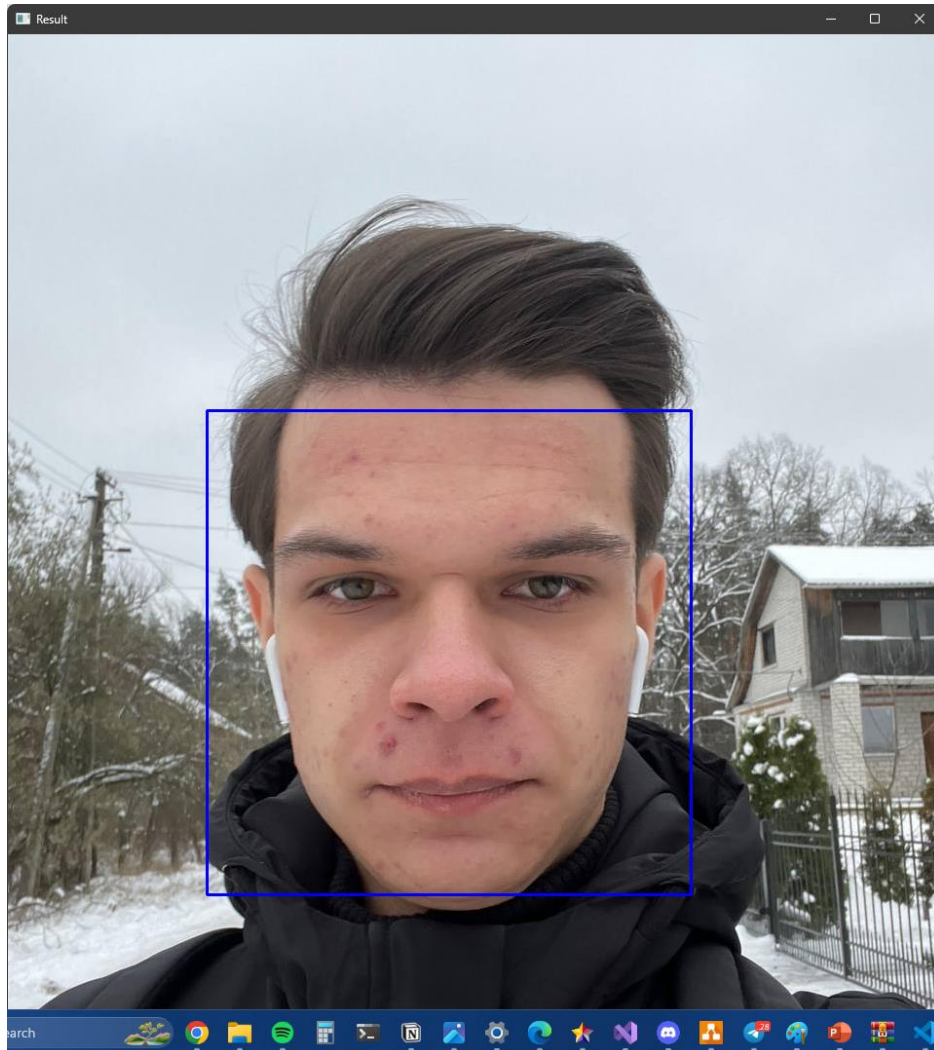


Рис. 4.1 – Результат виконання програми

**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:
```

		Скаковський В.О.			ДУ «Житомирська політехніка». 23.121.17.000 – Лр8	Арк.
		Голенко М.Ю.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

```

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()

```

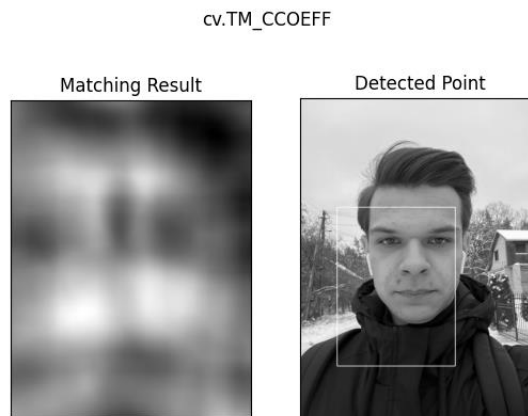


Рис. 5.1 – Результат виконання програми

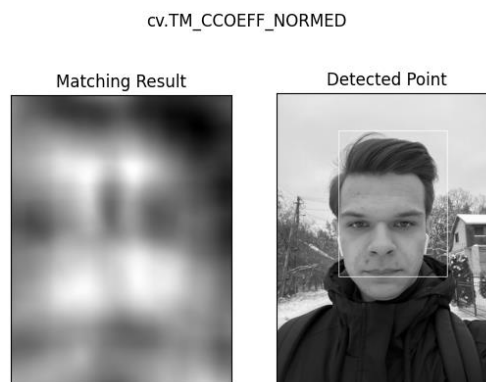


Рис. 5.2 – Результат виконання програми

		Скаковський В.О.			ДУ «Житомирська політехніка».23.121.17.000 – Лр8	Арк.
		Голенко М.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		8



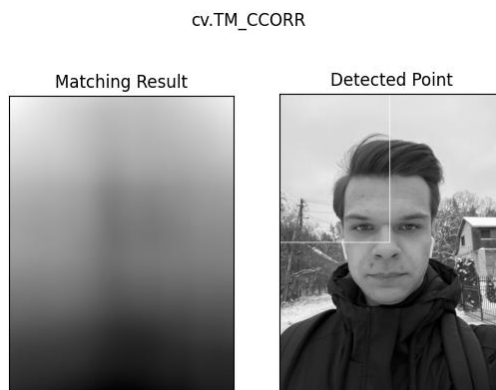


Рис. 5.3 – Результат виконання програми

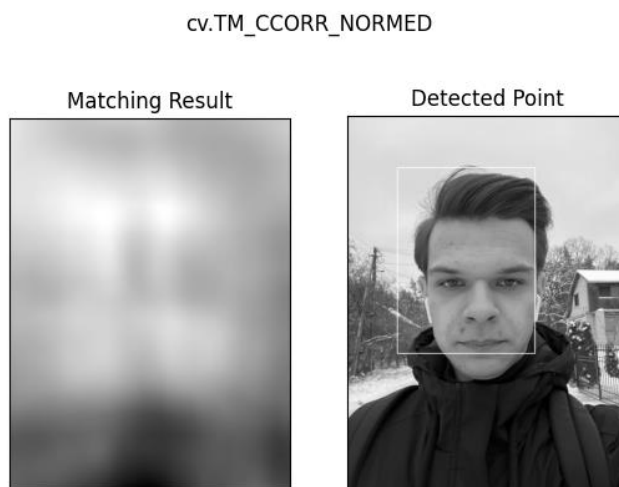


Рис. 5.4 – Результат виконання програми

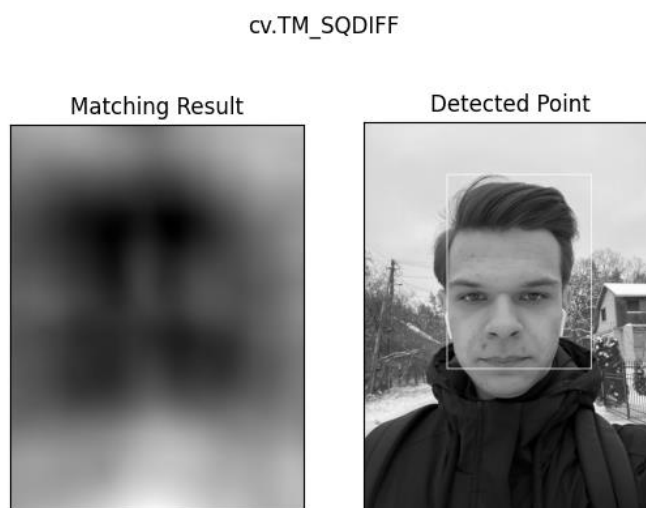


Рис. 5.5 – Результат виконання програми

cv.TM\_SQDIFF\_NORMED



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 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
```

		Скаковський В.О.			ДУ «Житомирська політехніка».23.121.17.000 – Лр8	Арк.
		Голенко М.Ю.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

```

markers = cv2.watershed(img,markers)
img[markers == -1] = [255,0,0]
cv2.imshow("coins_markers",img)
cv2.waitKey(0)

```



Рис. 6.1 – Результат виконання програми

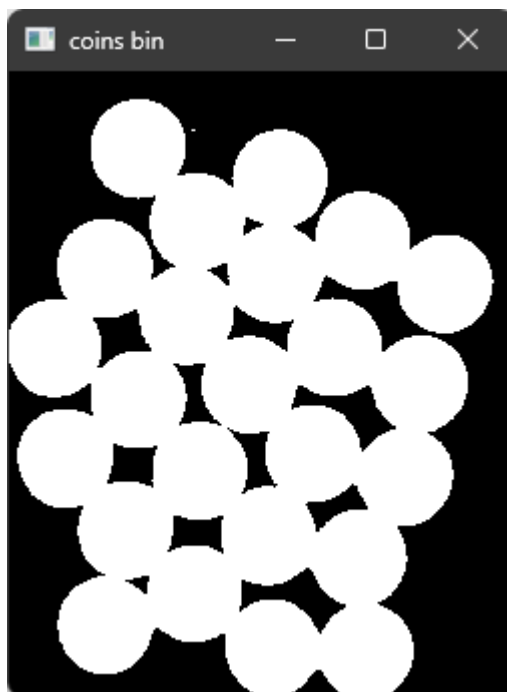


Рис. 6.2 – Результат виконання програми

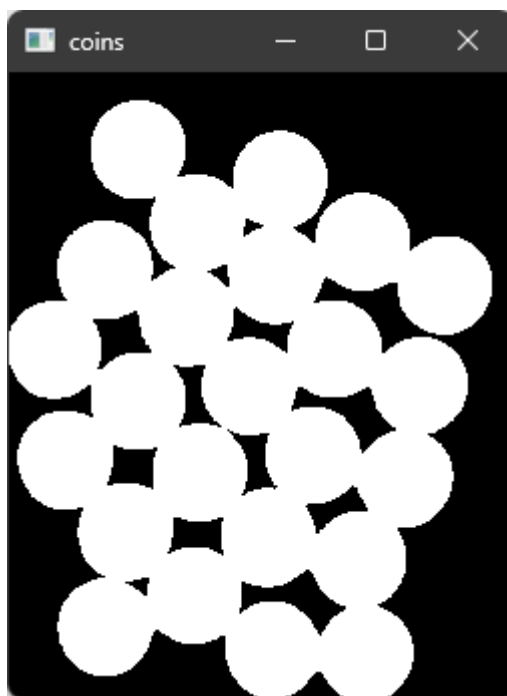


Рис. 6.3 – Результат виконання програми



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.

		Скаковський В.О			ДУ «Житомирська політехніка».23.121.17.000 – Лр8	Арк.
		Голенко М.Ю.				12
Змн.	Арк.	№ докум.	Підпис	Дата		