

## ЛАБОРАТОРНА РОБОТА № 8

### ДОСЛІДЖЕННЯ МЕТОДІВ КОМП'ЮТЕРНОГО ЗОРУ

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

#### Завдання 2.1:

```
import cv2
# LOAD AN IMAGE USING 'IMREAD'
img = cv2.imread("demchenko.jpg")
# DISPLAY
cv2.imshow("demchenko", img)
cv2.waitKey(0)
```

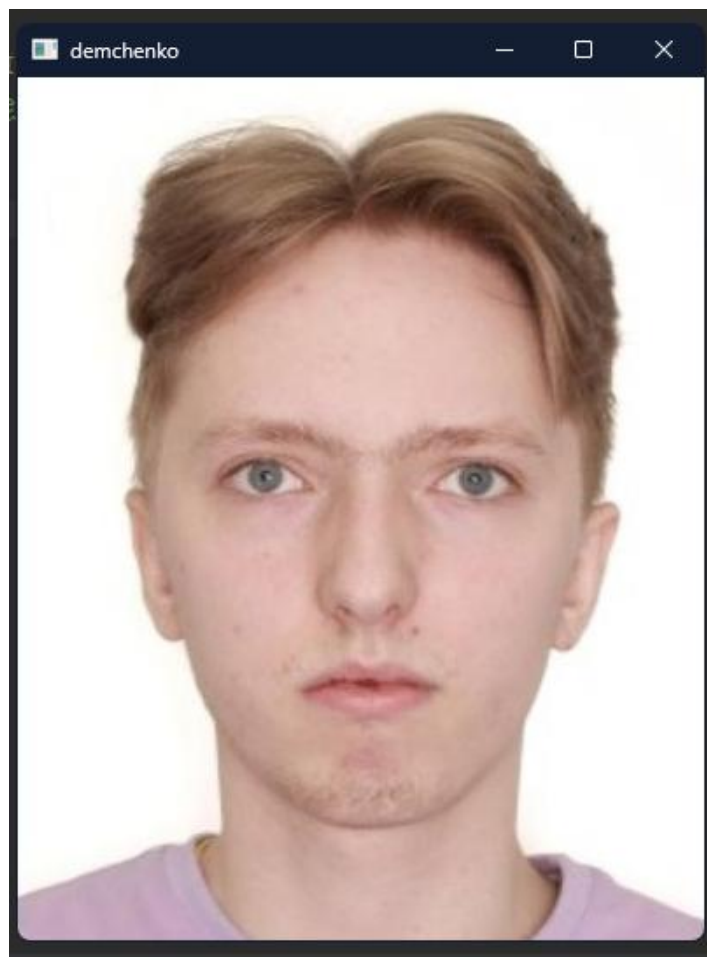


Рис. 1. Результат виконання програми

					ДУ «Житомирська політехніка».22.121.04.000 – Лр8								
Змн.	Арк.	№ докум.	Підпис	Дата									
Розроб.		Демченко Я. Д.			Звіт з  лабораторної роботи			Літ.		Арк.		Аркушів	
Перевір.		Філіпов В. О.								1		5	
Керівник								ФІКТ Гр. ІПЗ-19-2					
Н. контр.													
Зав. каф.													

## Завдання 2.2:

```
import cv2
import numpy as np
img = cv2.imread("demchenko.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)
```



Рис. 2. Результат виконання програми

		Демченко Я. Д.			ДЧ «Житомирська політехніка».22.121.04.000 – Лр8	Арк.
		Філіпов В. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		2



Рис. 3. Результат виконання програми

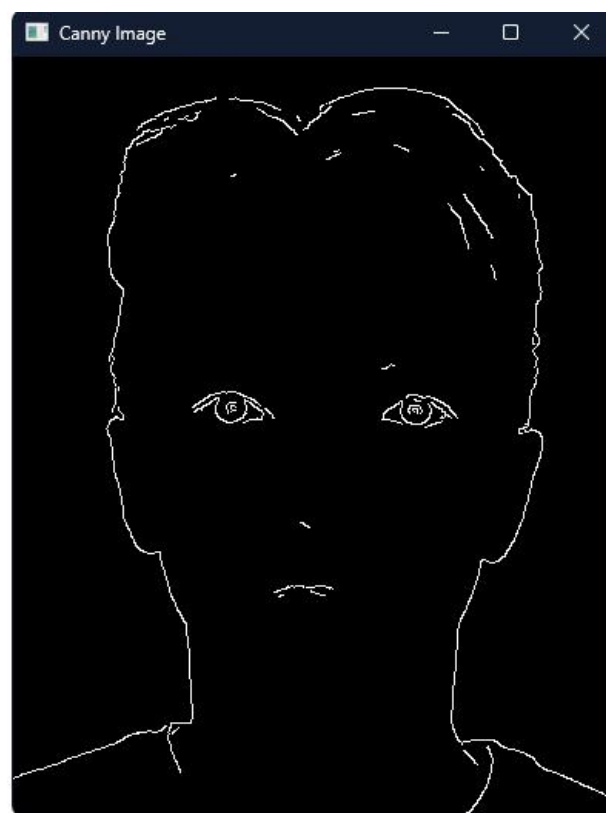


Рис. 4. Результат виконання програми

		Демченко Я. Д.			ДЧ «Житомирська політехніка».22.121.04.000 – Лр8	Арк.
		Філіпов В. О.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

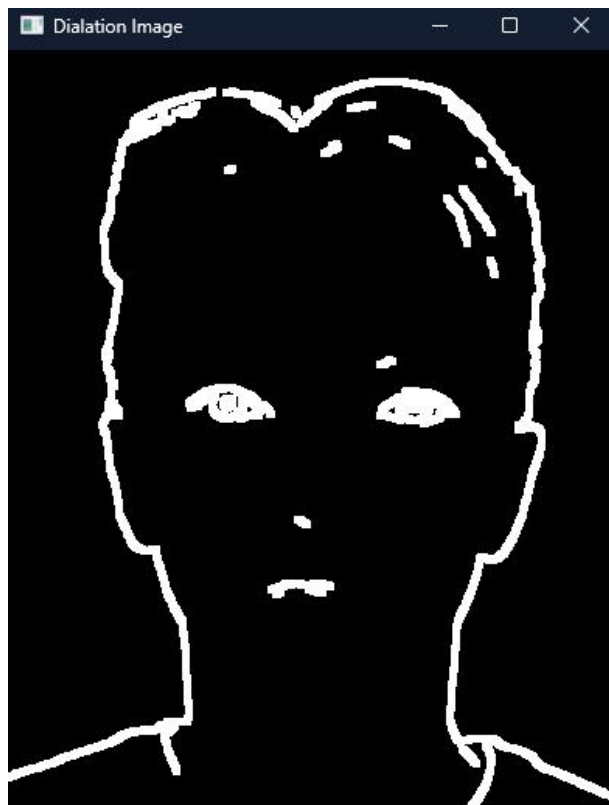


Рис. 5. Результат виконання програми

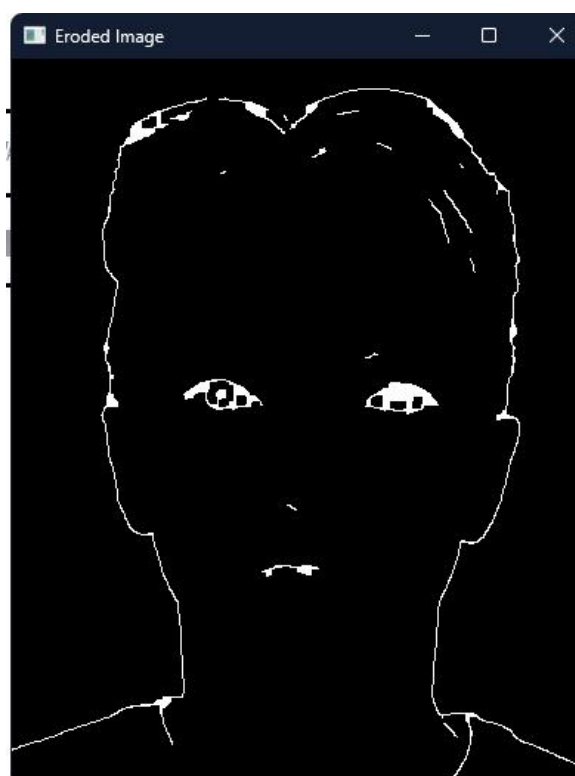


Рис. 6. Результат виконання програми

**cvtColor** можна використовувати для зміни колірного простору зображення, в результаті було отримано чорно-біле зображення

		Демченко Я. Д.			ДЧ «Житомирська політехніка».22.121.04.000 – Лр8	Арк.
		Філіпов В. О.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

**GaussianBlur** можна використовувати для застосування Гаусового згладжування, в результаті було отримано розмите зображення.

**Canny** можна використовувати для виявлення країв зображення, в результаті були отримані контури обличчя.

**Dilate** можна використовувати для того, щоб зменшити особливості зображення, в результаті були отримані контури обличчя.

**Erode** можна використовувати для підкреслення рисунку, в результаті було отримано розмитий контур обличчя.

### Завдання 2.3:

```
import cv2

img = cv2.imread("demchenko.jpg")
print(img.shape)
imgResize = cv2.resize(img, (350, 470))
print(imgResize.shape)
imgCropped = img[10:450, 50:350]
cv2.imshow("Image", img)
cv2.imshow("Image Cropped", imgCropped)
cv2.waitKey(0)
```



Рис. 7. Результат виконання програми

		Демченко Я. Д.			ДЧ «Житомирська політехніка».22.121.04.000 – Лр8	Арк.
		Філіпов В. О.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

### Завдання 2.4:

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

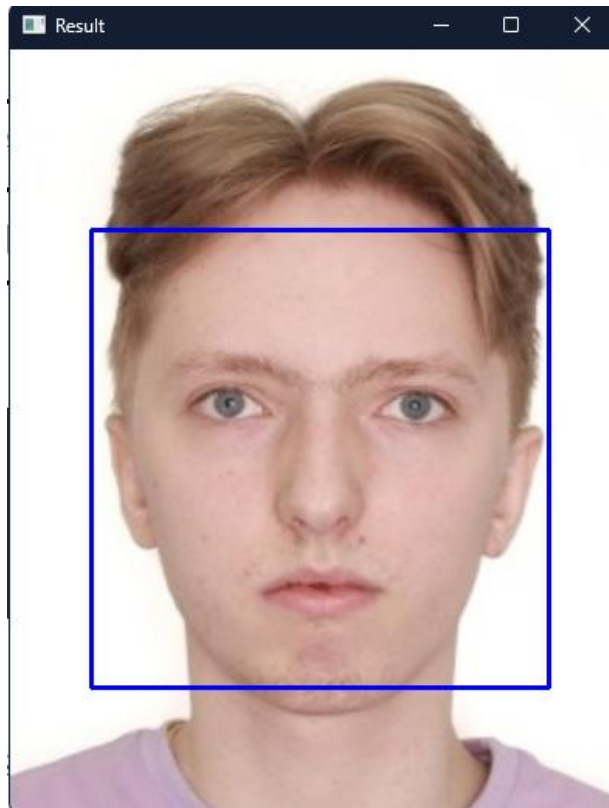


Рис. 8. Результат виконання програми

Були отримані досить непогані результати з розпізнаванням власного обличчя

### Завдання 2.5:

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

img = cv.imread('messi_full.jpg', 0)
img2 = img.copy()
template = cv.imread('messi_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:
```

		Демченко Я. Д.			ДЧ «Житомирська політехніка».22.121.04.000 – Лр8	Арк.
		Філіпов В. О.				6
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()

```

cv.TM\_CCOEFF



Рис. 9. Результат виконання програми

		Демченко Я. Д.			ДЧ «Житомирська політехніка».22.121.04.000 – Лр8	Арк.
		Філіпов В. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

cv.TM\_CCOEFF\_NORMED

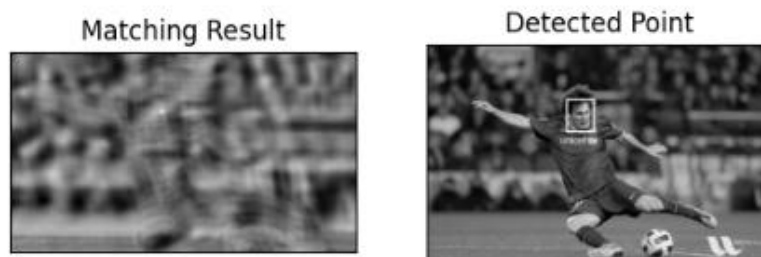


Рис. 10. Результат виконання програми

cv.TM\_CCORR



Рис. 11. Результат виконання програми

		Демченко Я. Д.			ДЧ «Житомирська політехніка».22.121.04.000 – Лр8	Арк.
		Філіпов В. О.				8
Змн.	Арк.	№ докум.	Підпис	Дата		



cv.TM\_CCORR\_NORMED

Matching Result



Detected Point



Рис. 12. Результат виконання програми

cv.TM\_SQDIFF

Matching Result



Detected Point



Рис. 13. Результат виконання програми

		Демченко Я. Д.			ДЧ «Житомирська політехніка».22.121.04.000 – Лр8	Арк.
		Філіпов В. О.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

cv.TM\_SQDIFF\_NORMED

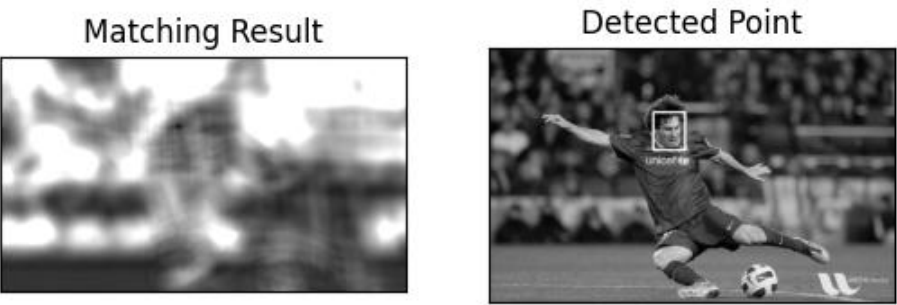


Рис. 14. Результат виконання програми

cv.TM\_CCoeff

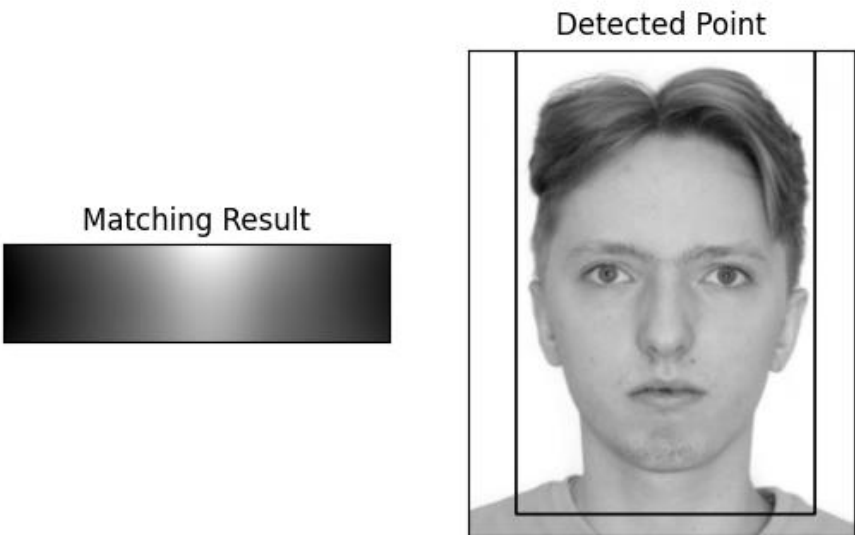


Рис. 15. Результат виконання програми

cv.TM\_CCOEFF\_NORMED



Рис. 16. Результат виконання програми

cv.TM\_CCORR

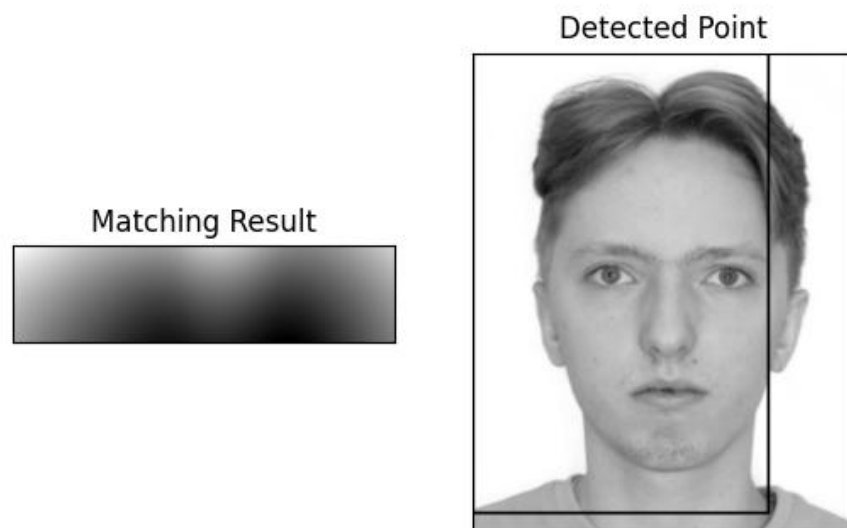


Рис. 17. Результат виконання програми

		Демченко Я. Д.			ДЧ «Житомирська політехніка».22.121.04.000 – Лр8	Арк.
		Філіпов В. О.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

cv.TM\_CCORR\_NORMED

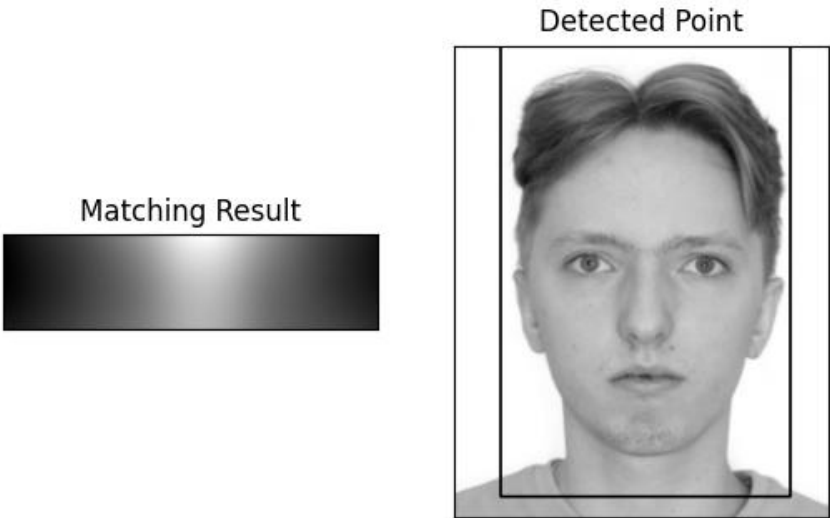


Рис. 18. Результат виконання програми

cv.TM\_SQDIFF



Рис. 19. Результат виконання програми

cv.TM\_SQDIFF\_NORMED



Рис. 20. Результат виконання програми

cv.TM\_CCoeff:

$$R(x, y) = \sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))$$

where

$$T'(x', y') = T(x', y') - 1/(w \cdot h) \cdot \sum_{x'', y''} T(x'', y'')$$

$$I'(x + x', y + y') = I(x + x', y + y') - 1/(w \cdot h) \cdot \sum_{x'', y''} I(x + x'', y + y'')$$

with mask:

$$T'(x', y') = M(x', y') \cdot \left( T(x', y') - \frac{1}{\sum_{x'', y''} M(x'', y'')} \cdot \sum_{x'', y''} (T(x'', y'') \cdot M(x'', y'')) \right)$$

$$I'(x + x', y + y') = M(x', y') \cdot \left( I(x + x', y + y') - \frac{1}{\sum_{x'', y''} M(x'', y'')} \cdot \sum_{x'', y''} (I(x + x'', y + y'') \cdot M(x'', y'')) \right)$$

cv.TM\_CCoeff\_NORMED:

$$R(x, y) = \frac{\sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))}{\sqrt{\sum_{x', y'} T'(x', y')^2 \cdot \sum_{x', y'} I'(x + x', y + y')^2}}$$

		Демченко Я. Д.			ДЧ «Житомирська політехніка».22.121.04.000 - Лр8	Арк.
		Філіпов В. О.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

cv.TM\_CCORR:

$$R(x, y) = \sum_{x', y'} (T(x', y') \cdot I(x + x', y + y'))$$

with mask:

$$R(x, y) = \sum_{x', y'} (T(x', y') \cdot I(x + x', y + y') \cdot M(x', y')^2)$$

cv.TM\_CCORR\_NORMED:

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') \cdot I(x + x', y + y'))}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}}$$

with mask:

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') \cdot I(x + x', y + y') \cdot M(x', y')^2)}{\sqrt{\sum_{x', y'} (T(x', y') \cdot M(x', y'))^2 \cdot \sum_{x', y'} (I(x + x', y + y') \cdot M(x', y'))^2}}$$

cv.TM\_SQDIFF:

$$R(x, y) = \sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2$$

with mask:

$$R(x, y) = \sum_{x', y'} ((T(x', y') - I(x + x', y + y')) \cdot M(x', y'))^2$$

cv.TM\_SQDIFF\_NORMED:

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}}$$

with mask:

$$R(x, y) = \frac{\sum_{x', y'} ((T(x', y') - I(x + x', y + y')) \cdot M(x', y'))^2}{\sqrt{\sum_{x', y'} (T(x', y') \cdot M(x', y'))^2 \cdot \sum_{x', y'} (I(x + x', y + y') \cdot M(x', y'))^2}}$$

cv.TM\_CCORR показав найгірший результат, інші показали однаковий результат

### Завдання 6:

```
import numpy as np
import cv2
```

		Демченко Я. Д.			ДЧ «Житомирська політехніка».22.121.04.000 - Лр8	Арк.
		Філіпов В. О.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

```

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)

```



Рис. 21. Результат виконання програми

		Демченко Я. Д.			ДЧ «Житомирська політехніка».22.121.04.000 - Лр8	Арк.
		Філіпов В. О.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

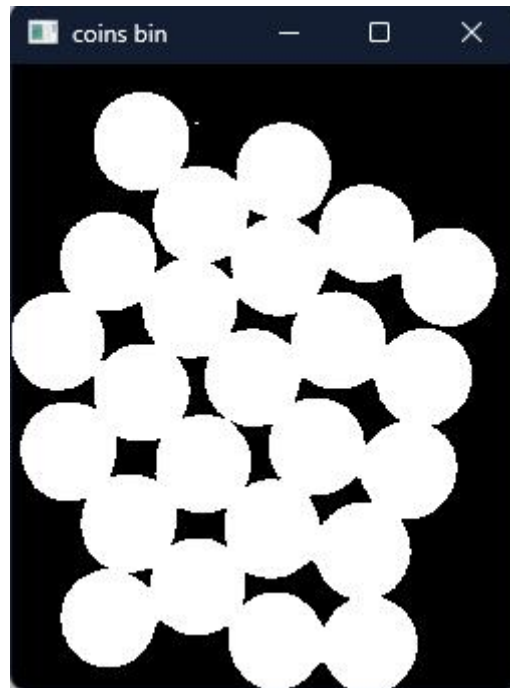


Рис. 22. Результат виконання програми

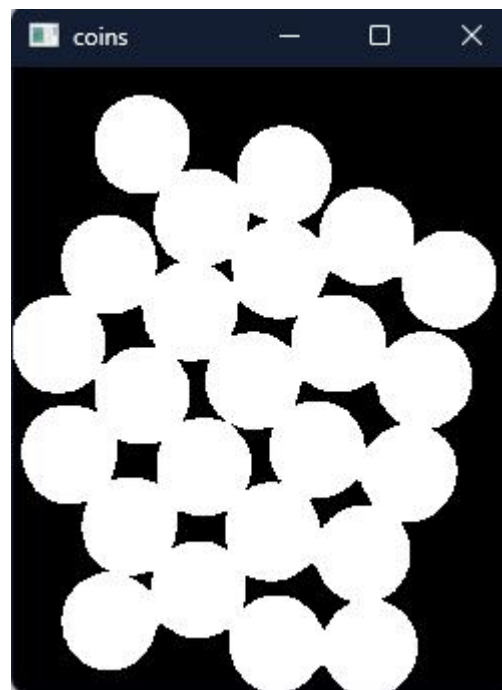


Рис. 23. Результат виконання програми





Рис. 24. Результат виконання програми

Більшості монет було правильно виділено, але для областей, де монети торкаються одне одного були проблеми. Тому області були не зовсім коректно визначені.

### Завдання 7:

```
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')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
filtro = cv2.pyrMeanShiftFiltering(img, 20, 40)
gray = cv2.cvtColor(filtro, cv2.COLOR_BGR2GRAY)
_, thresh = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY_INV |
                          cv2.THRESH_OTSU)

contornos, _ = cv2.findContours(thresh, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
buracos = []
for con in contornos:
    area = cv2.contourArea(con)
    if area < 1000:
        buracos.append(con)
cv2.drawContours(thresh, buracos, -1, 255, -1)
dist = ndi.distance_transform_edt(thresh)
dist_visual = dist.copy()
local_max = peak_local_max(dist, indices=False, min_distance=20, labels = thresh)
markers = ndi.label(local_max, structure=np.ones((3, 3)))[0]
labels = watershed(-dist, markers, mask=thresh)
titulos = ['Original image', 'Binary Image', 'Distance Transform', 'Watershed']
imagens = [img, thresh, dist_visual, labels]
fig = plt.gcf()
```

		Демченко Я. Д.			ДЧ «Житомирська політехніка».22.121.04.000 - Лр8	Арк.
		Філіпов В. О.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

```

fig.set_size_inches(16, 12)
for i in range(4):
    plt.subplot(2, 2, i + 1)
    if i == 3:
        cmap = "jet"
    else:
        cmap = "gray"
    plt.imshow(imagens[i], cmap)
    plt.title(titulos[i])
    plt.xticks([], plt.yticks([]))
plt.show()

```

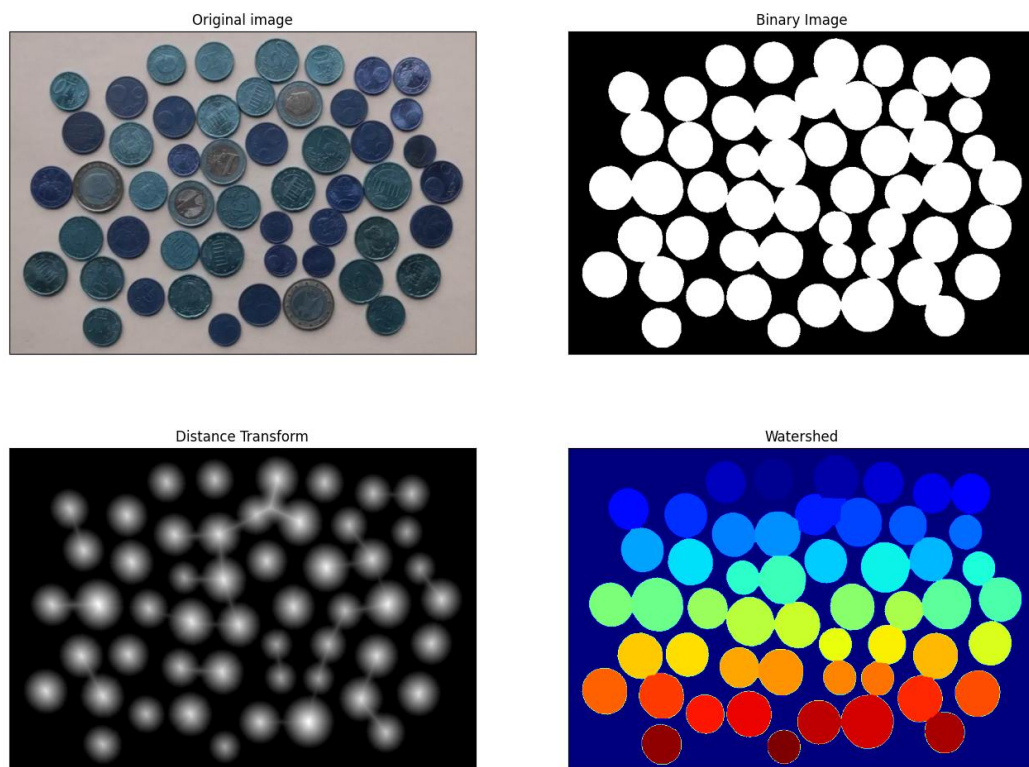


Рис. 25. Результат виконання програми

Отримано зображення з сегментами, для такої якості початкового зображення результат є доволі непоганим.

**Репозиторій:** <https://github.com/ipz192dyad/Artificial-intelligence-systems>

**Висновок:** в ході виконання лабораторної роботи використовуючи спеціалізовані бібліотеки та мову програмування Python отримано навички для обробки зображень за допомогою бібліотеки OpenCV.

		Демченко Я. Д.			ДЧ «Житомирська політехніка».22.121.04.000 – Лр8	Арк.
		Філіпов В. О.				18
Змн.	Арк.	№ докум.	Підпис	Дата		