

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 2.1. Завантаження зображень та відео в OpenCV

Лістинг програми:

```
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("dubinchenko.jpg")
cv2.imshow("dubinchenko", img)
cv2.waitKey(0)
```

Результат виконання програми:

					ДУ «Житомирська політехніка».23.121.05.000 – Лр8			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Дубинченко Б.М.			Звіт з лабораторної роботи		Літ.	Арк.
Перевір.		Іванов Д.А.						1
Керівник								18
Н. контр.							ФІКТ Гр. ІПЗ-20-4[1]	
Зав. каф.								

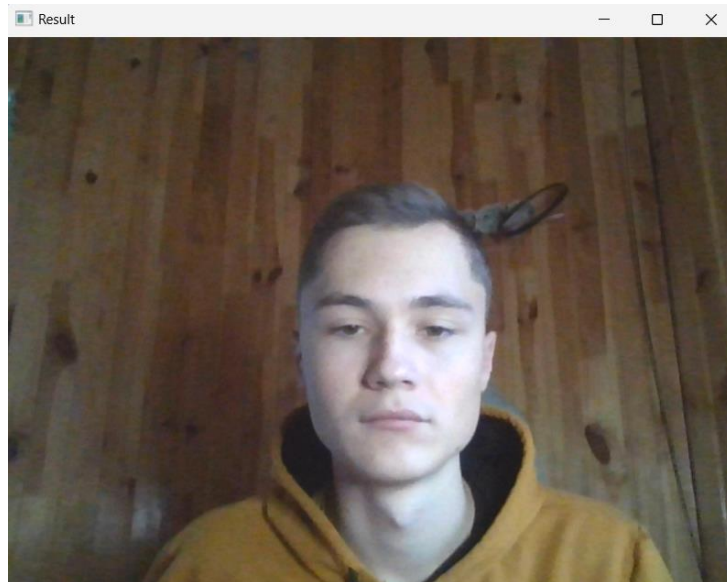


Рис. 7.1. Результат виконання програми

Завдання 2.2. Дослідження перетворень зображення

Лістинг програми:

```
import cv2
import numpy as np

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

Результат виконання програми:

		Дубинченко Б.М.			ДУ «Житомирська політехніка».23.121.05.000 – Лр8	Арк.
		Іванов Д.А.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

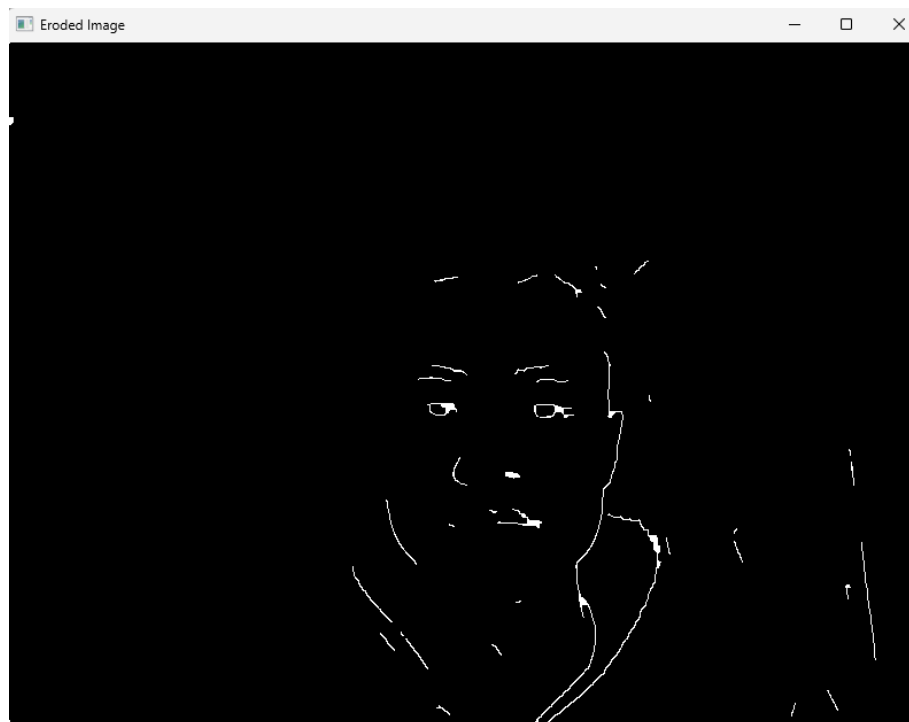


Рис. 7.2. Результат виконання програми

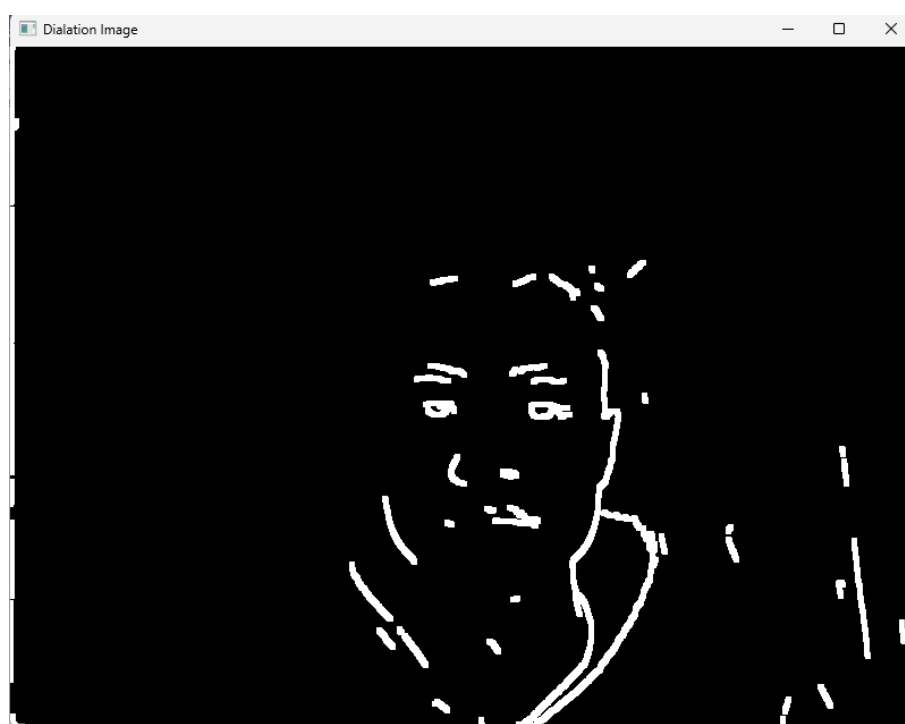


Рис. 7.3. Результат виконання програми

		Дубинченко Б.М.			ДУ «Житомирська політехніка».23.121.05.000 – Лр8	Арк.
		Іванов Д.А.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

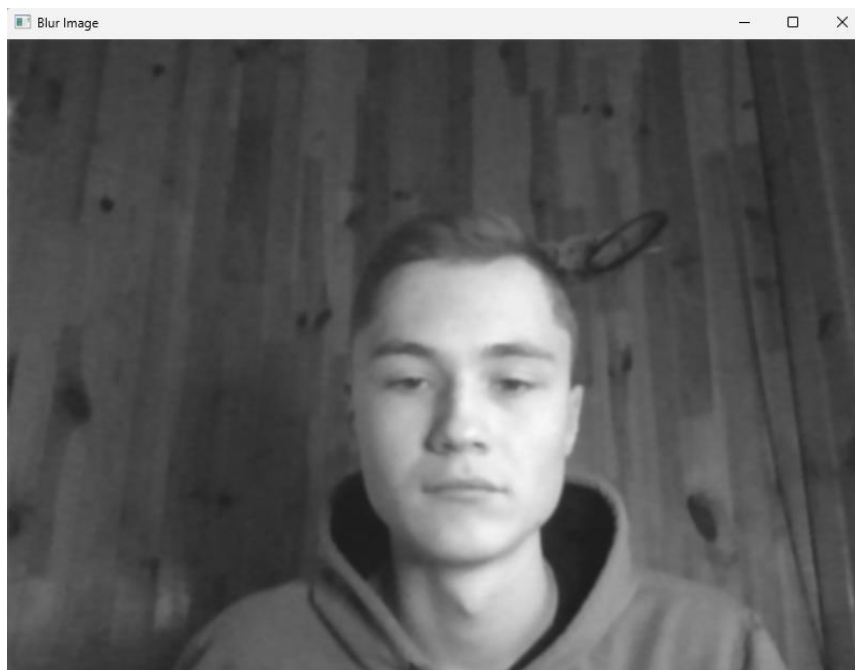


Рис. 7.4. Результат виконання програми

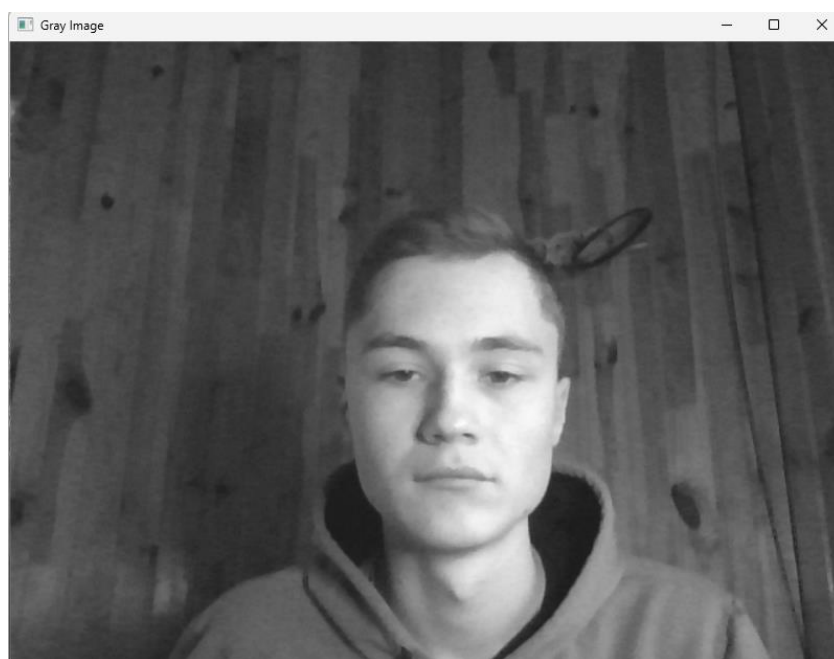


Рис. 7.5. Результат виконання програми

		Дубинченко Б.М.			ДУ «Житомирська політехніка».23.121.05.000 – Лр8	Арк.
		Іванов Д.А.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

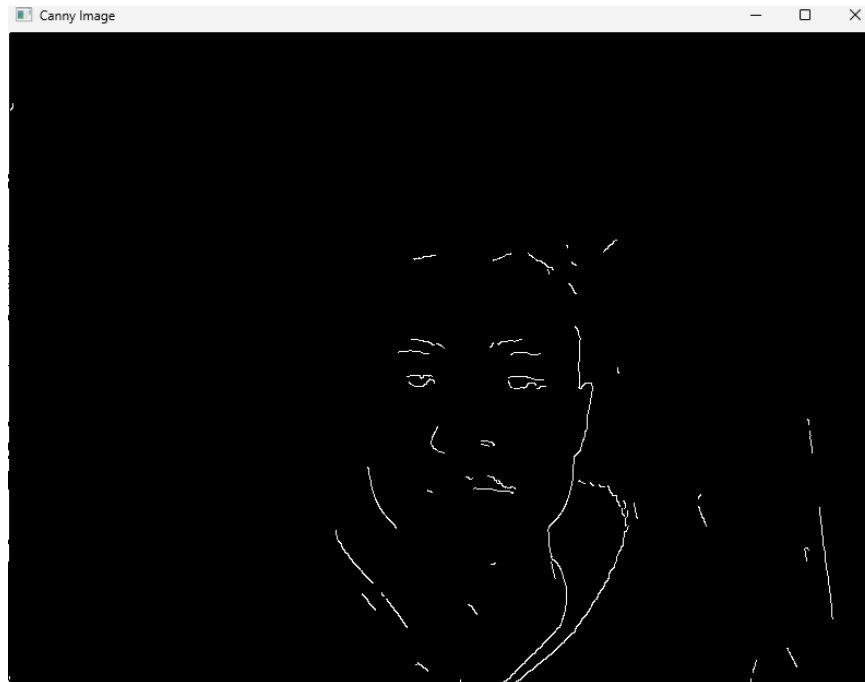


Рис. 7.6. Результат виконання програми

cvtColor - використовується для перетворення зображення з одного колірного простору на інший.

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.

erode - метод використовується для виконання розмивання зображення.

GaussianBlur - розмите зображення.

Canny - використовується для визначення країв зображення.

Завдання 2.3. Вирізання частини зображення

Лістинг програми:

```
import cv2
import numpy as np

img = cv2.imread("shapes.png")
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)
```

		Дубинченко Б.М.			ДУ «Житомирська політехніка».23.121.05.000 – Лр8	Арк.
		Іванов Д.А.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

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

Результат виконання програми:

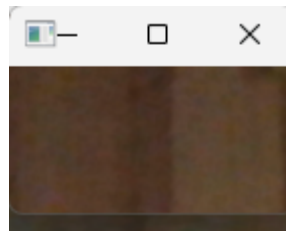


Рис. 7.7. Результат виконання програми

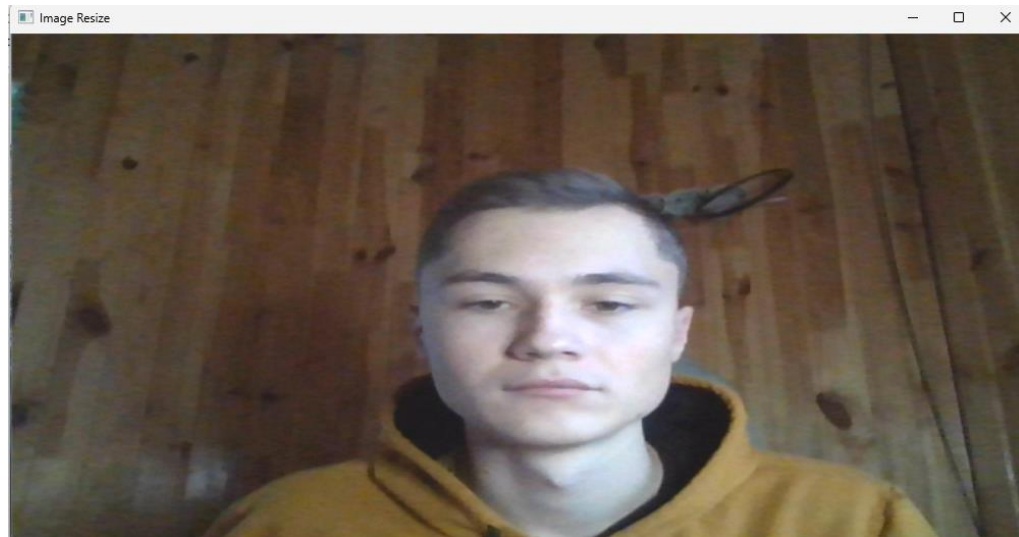


Рис. 7.8. Результат виконання програми

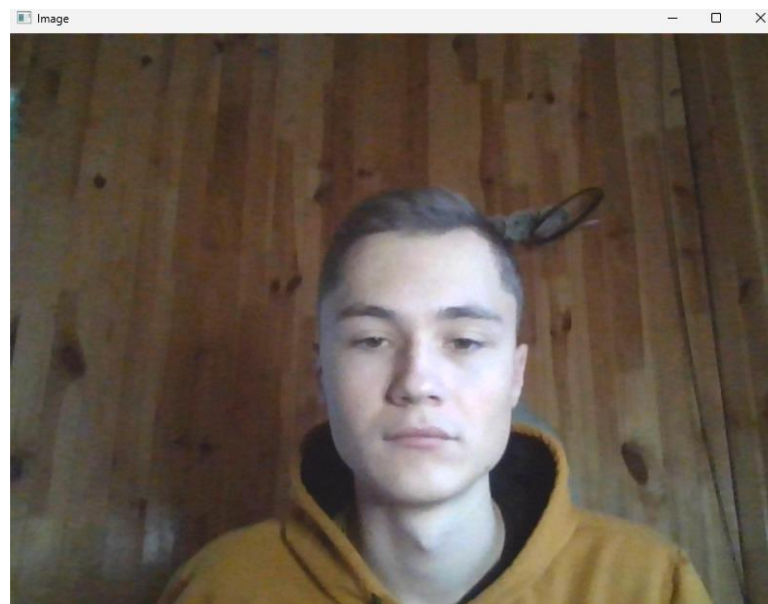


Рис. 7.9. Результат виконання програми

Завдання 2.4. Розпізнавання обличчя на зображенні

Лістинг програми:

		Дубинченко Б.М.			ДУ «Житомирська політехніка».23.121.05.000 – Лр8	Арк.
		Іванов Д.А.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```
import cv2

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

Результат виконання програми:

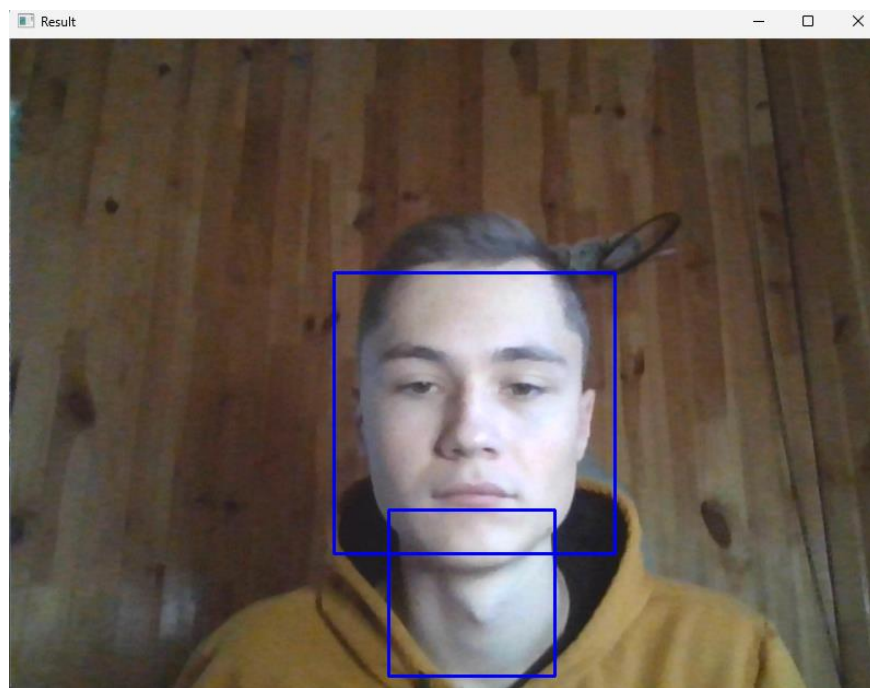


Рис. 7.10. Результат виконання програми

Завдання 2.5. Розпізнавання об'єктів на зображенні за допомогою методів зіставлення шаблонів (Template Matching)

Лістинг програми:

```
import cv2 as cv
import numpy as np
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:
    img = img2.copy()
    method = eval(meth)
```

		Дубинченко Б.М.			ДУ «Житомирська політехніка».23.121.05.000 – Лр8	Арк.
		Іванов Д.А.				7
Змн.	Арк.	№ докум.	Підпис	Дата		


```
# 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()
```

Результат виконання програми:

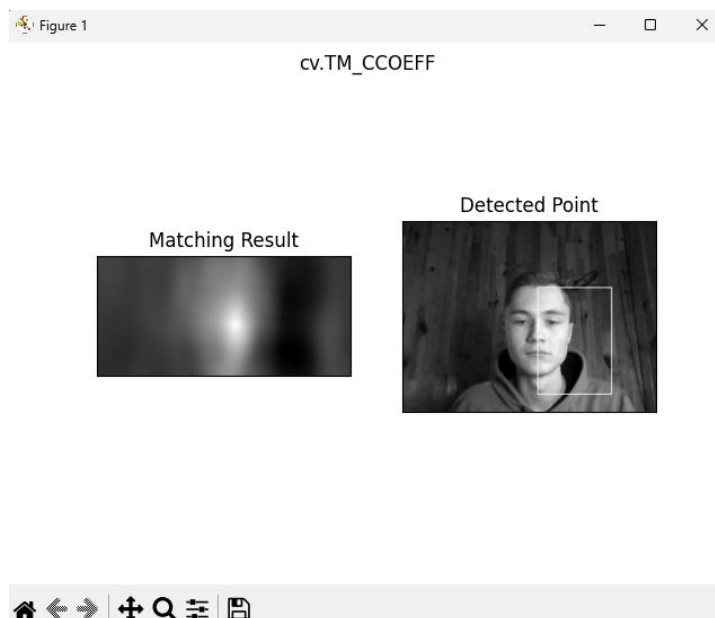
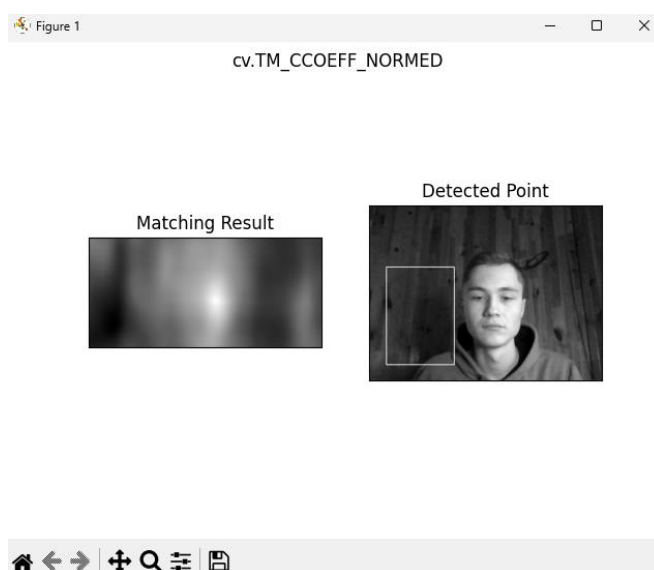


Рис. 7.11. Результат виконання програми



		Дубинченко Б.М.			ДУ «Житомирська політехніка».23.121.05.000 – Пр8	Арк.
		Іванов Д.А.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

Рис. 7.12. Результат виконання програми

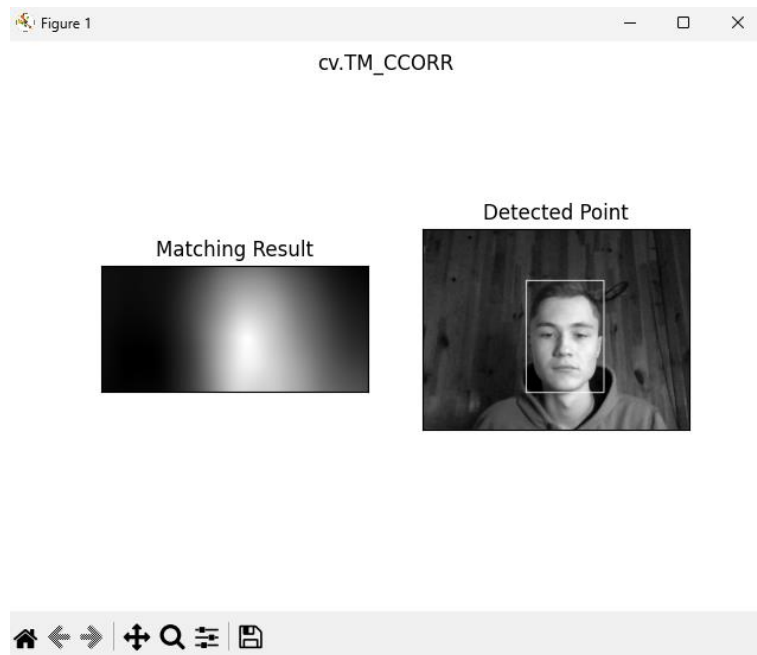


Рис. 7.13. Результат виконання програми

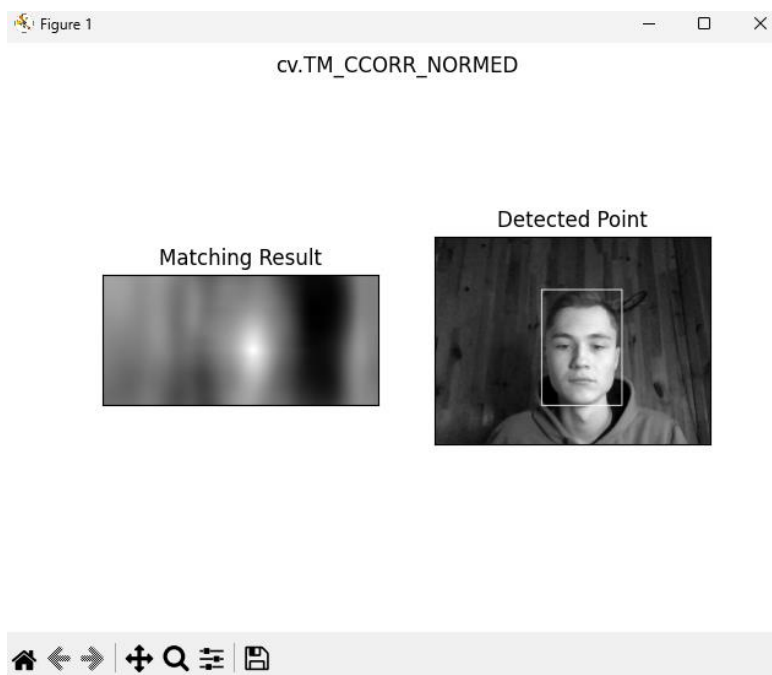


Рис. 7.14. Результат виконання програми

		Дубинченко Б.М.			ДУ «Житомирська політехніка».23.121.05.000 – Лр8	Арк.
		Іванов Д.А.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

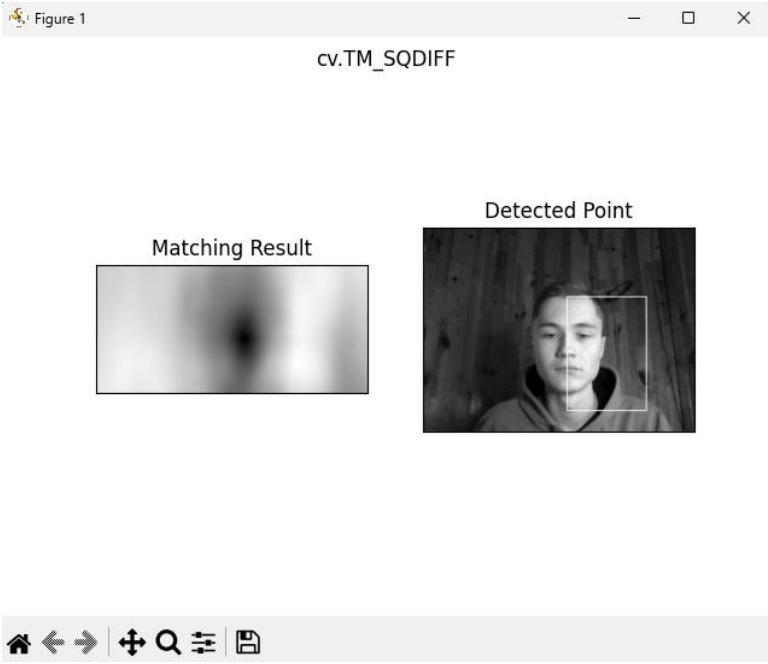


Рис. 7.15. Результат виконання програми

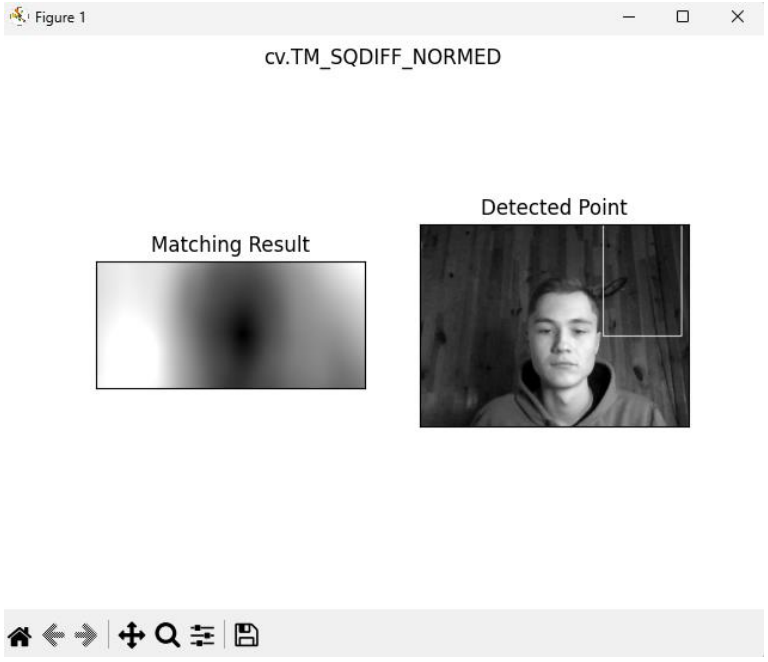


Рис. 7.16. Результат виконання програми

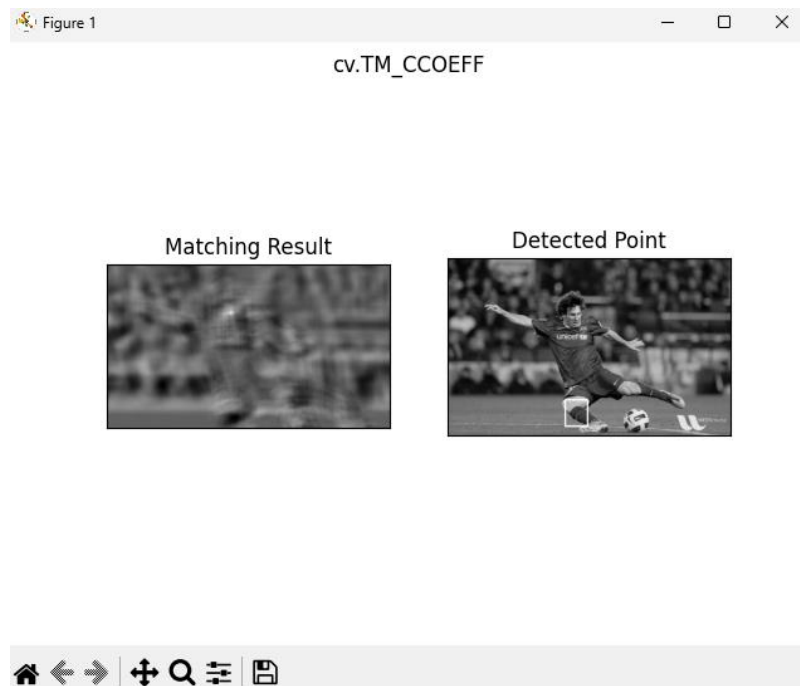


Рис. 7.17. Результат виконання програми

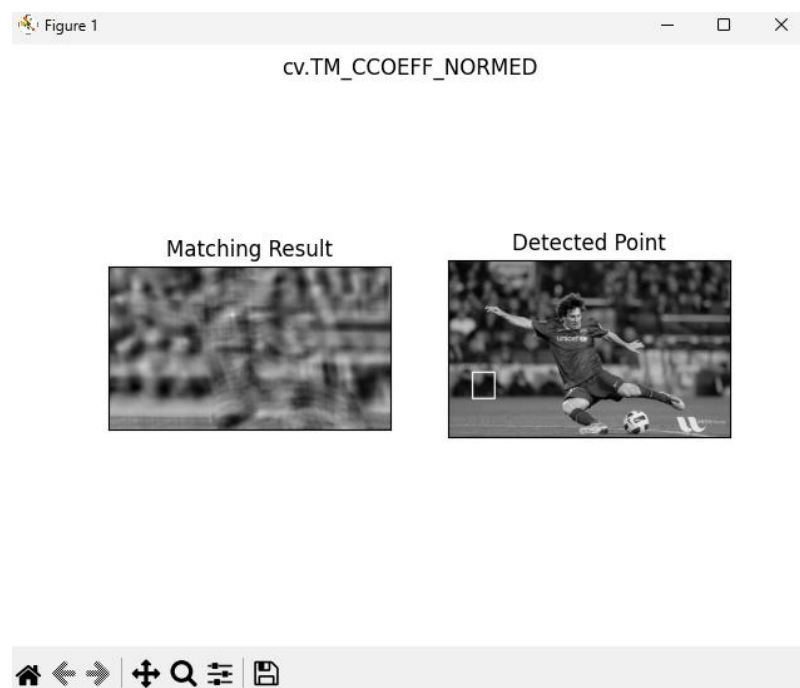


Рис. 7.18. Результат виконання програми

		Дубинченко Б.М.			ДУ «Житомирська політехніка».23.121.05.000 – Лр8	Арк.
		Іванов Д.А.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

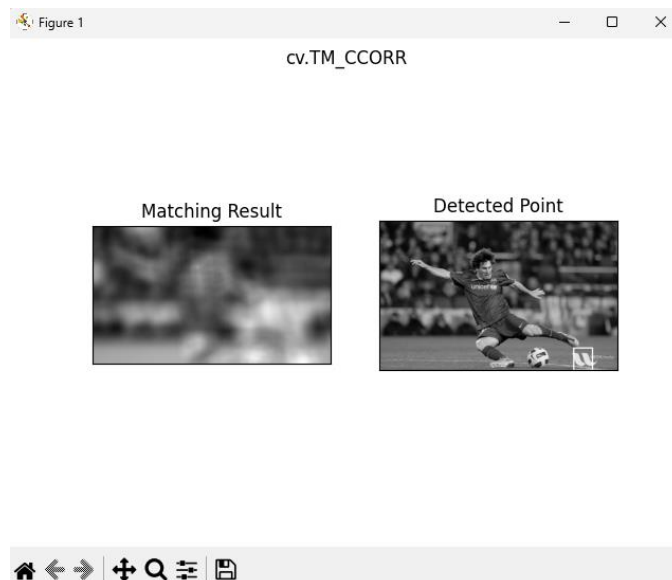


Рис. 7.19. Результат виконання програми

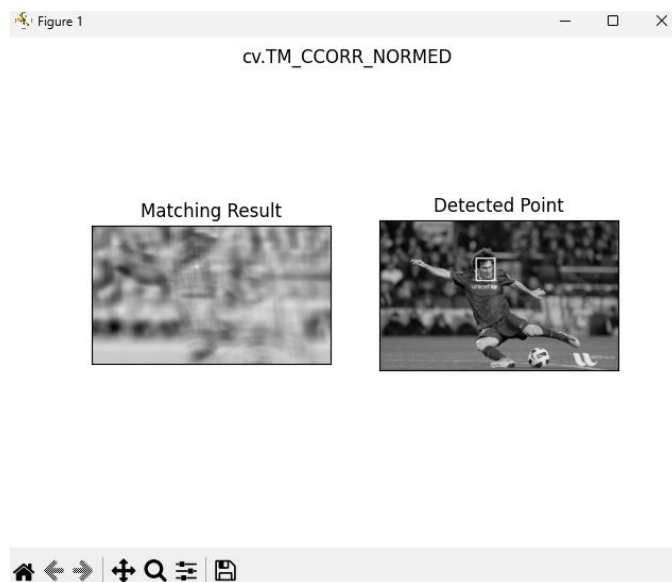


Рис. 7.20. Результат виконання програми

		Дубинченко Б.М.			ДУ «Житомирська політехніка».23.121.05.000 – Лр8	Арк.
		Іванов Д.А.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

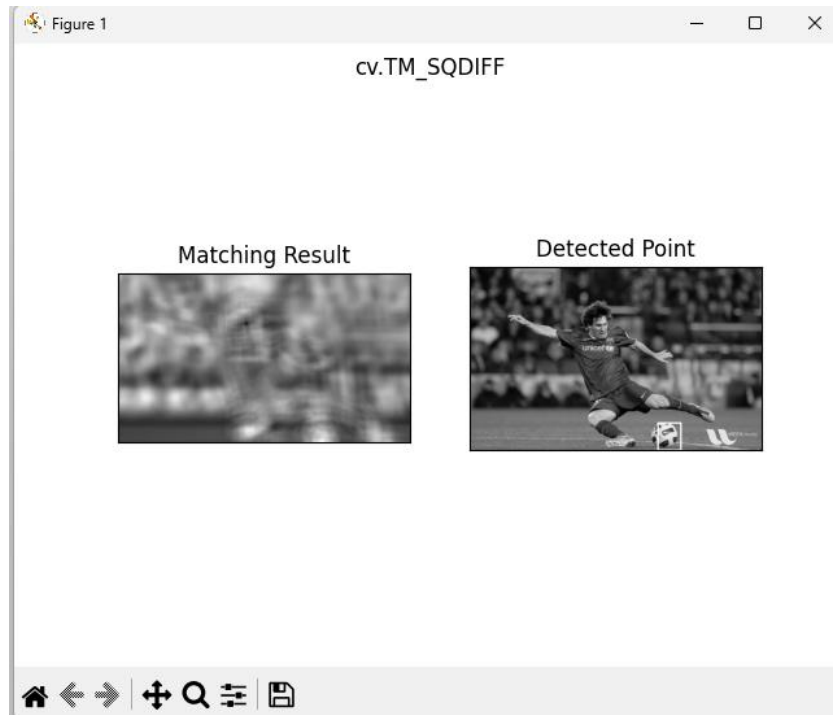


Рис. 7.21. Результат виконання програми

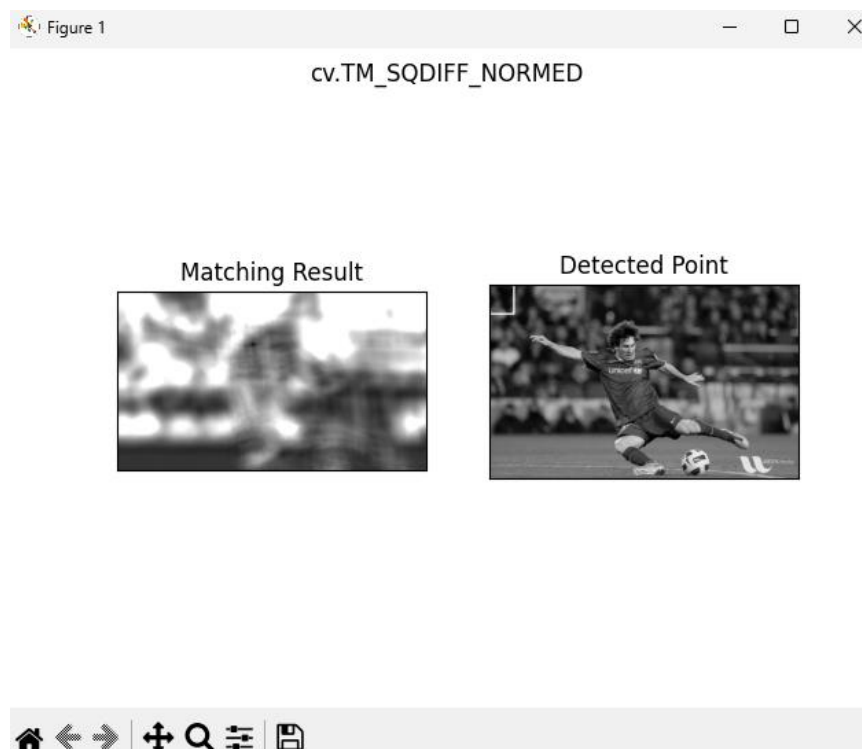


Рис. 7.22. Результат виконання програми

Завдання 2.6. Сегментація зображення алгоритмом водорозподілу

Лістинг програми:

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

		Дубинченко Б.М.			ДУ «Житомирська політехніка».23.121.05.000 – Лр8	Арк.
		Іванов Д.А.				13
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)

```

Результат виконання завдання:



Рис. 7.23. Результат виконання програми

		Дубинченко Б.М.			ДУ «Житомирська політехніка».23.121.05.000 – Лр8	Арк.
		Іванов Д.А.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

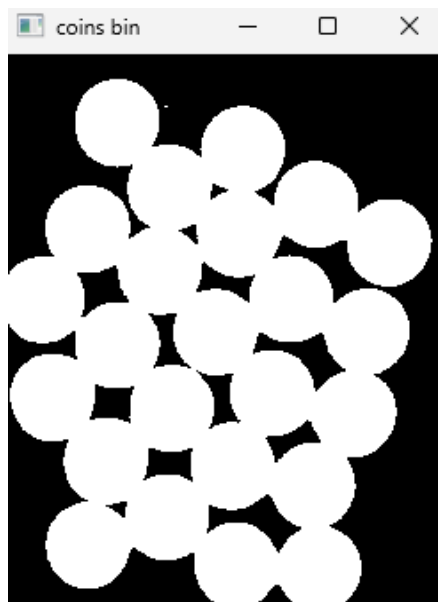


Рис. 7.24. Результат виконання програми

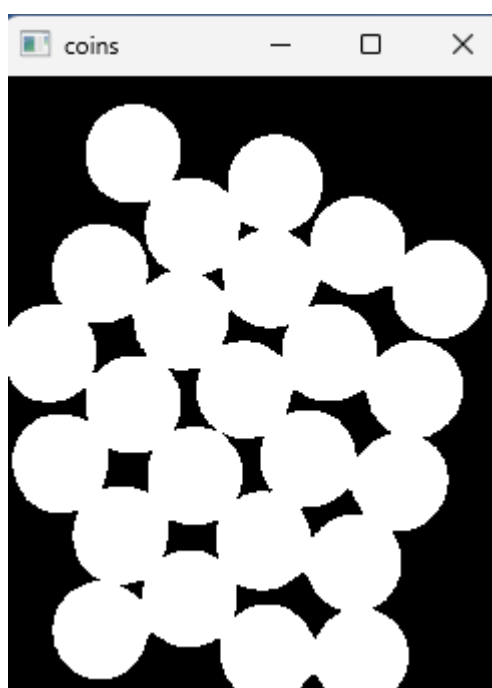


Рис. 7.25. Результат виконання програми

		Дубинченко Б.М.			ДУ «Житомирська політехніка».23.121.05.000 – Лр8	Арк.
		Іванов Д.А.				15
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 7.26. Результат виконання програми

2.7. Сегментація зображення

Лістинг програми:

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

img = cv2.imread('coins_2.JPG')
cv2.imshow("coins", img)
cv2.waitKey(0)

# Конвертуємо зображення в відтінки сірого
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Порогова обробка
ret, thresh = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)

# Видаляємо шум
kernel = np.ones((3, 3), np.uint8)
opening = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, kernel, iterations=2)

# Визначаємо фонову область
sure_bg = cv2.dilate(opening, kernel, iterations=3)

# Відстанційне перетворення
dist_transform = cv2.distanceTransform(opening, cv2.DIST_L2, 5)
ret, sure_fg = cv2.threshold(dist_transform, 0.7 * dist_transform.max(), 255, 0)

# Визначаємо невідому область
sure_fg = np.uint8(sure_fg)
unknown = cv2.subtract(sure_bg, sure_fg)

# Маркування міток
ret, markers = cv2.connectedComponents(sure_fg)

# Додаємо кожній маркованій області свій унікальний колір
```

		Дубинченко Б.М.			ДУ «Житомирська політехніка».23.121.05.000 – Лр8	Арк.
		Іванов Д.А.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

```

colors = []
for i in range(1, ret + 1):
    colors.append(np.random.randint(0, 255, 3))

# Розфарбовуємо зображення відповідно до маркування
colored_markers = np.zeros_like(img)
for label in range(1, ret + 1):
    colored_markers[markers == label] = colors[label - 1]

# Додаємо невідому область червоним кольором
colored_markers[unknown == 255] = [0, 0, 255]

# Відображення результату
cv2.imshow("Coins Segmentation", colored_markers)
cv2.waitKey(0)
cv2.destroyAllWindows()

# 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)

```

Результат виконання програми:



Рис. 7.27. Результат виконання програми

		Дубинченко Б.М.			ДУ «Житомирська політехніка».23.121.05.000 – Лр8	Арк.
		Іванов Д.А.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

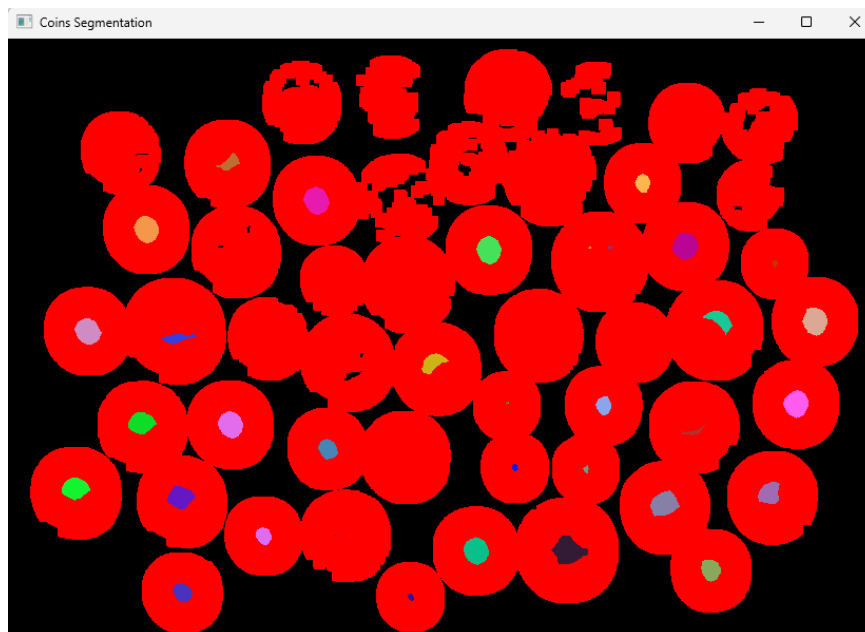


Рис. 7.28. Результат виконання програми



Рис. 7.29. Результат виконання програми

Посилання на GitHub: https://github.com/BogdanStelmah/Basics-of-AI_labs

Висновок: На даній лабораторній роботі ми використовуємо спеціалізовані бібліотеки та мову програмування Python навчилися обробляти зображення за допомогою бібліотеки OpenCV.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».23.121.05.000 – Лр8	Арк.
		Іванов Д.А.				18
Змн.	Арк.	№ докум.	Підпис	Дата		