

Лабораторна робота №8

ДОСЛІДЖЕННЯ МЕТОДІВ КОМП'ЮТЕРНОГО ЗОРУ

Мета роботи: 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
# LOAD AN IMAGE USING 'IMREAD'
img = cv2.imread("Hrunytskyi.jpg")
# DISPLAY
cv2.imshow("Hrunytskyi",img)
cv2.waitKey(0)
```

Результат виконання програми:

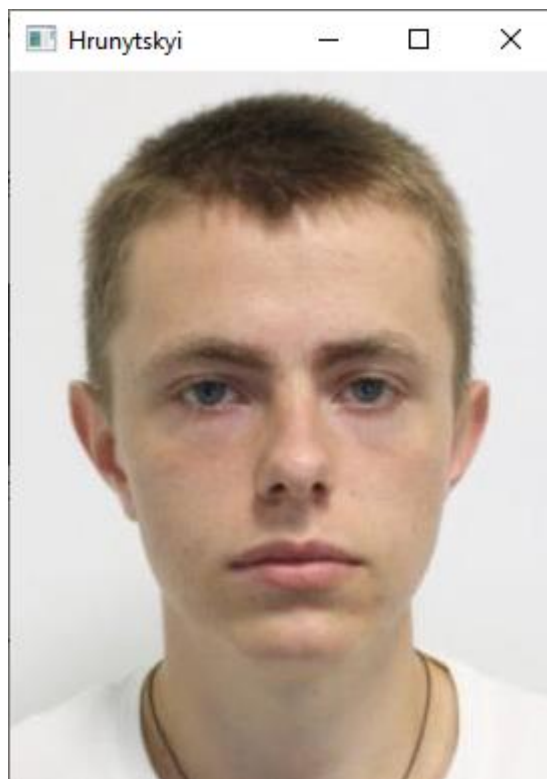


Рис. 2.1.1 – Результат виконання завдання.

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.6.000 – Лр.8									
Змн.	Арк.	№ докум.	Підпис	Дата										
Розроб.		Груницький Д.С.			Звіт з лабораторної роботи №8				Літ.		Арк.		Аркушів	
Перевір.		Голенко М.Ю.									1		17	
Реценз.									ФІКТ, гр.ІПЗ-20-3					
Н. Контр.														
Зав.каф.														

Висновок

У висновку можна сказати, що цей код просто завантажує та відображує зображення "Hrunytskyi.jpg".

Завдання 2.2. Дослідження перетворень зображення.

Лістинг програми:

```
import cv2
import numpy as np

img = cv2.imread("Hrunytskyi.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("Dialation Image", imgDilation)
cv2.imshow("Eroded Image", imgEroded)
cv2.waitKey(0)
```

Результат виконання програми:



Рис. 2.2.1 – Результат виконання завдання.

		Груницький Д.С.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.6.000 – Лр.8	Арк.
		Голенко М.Ю.				2
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 2.2.2 – Результат виконання завдання.

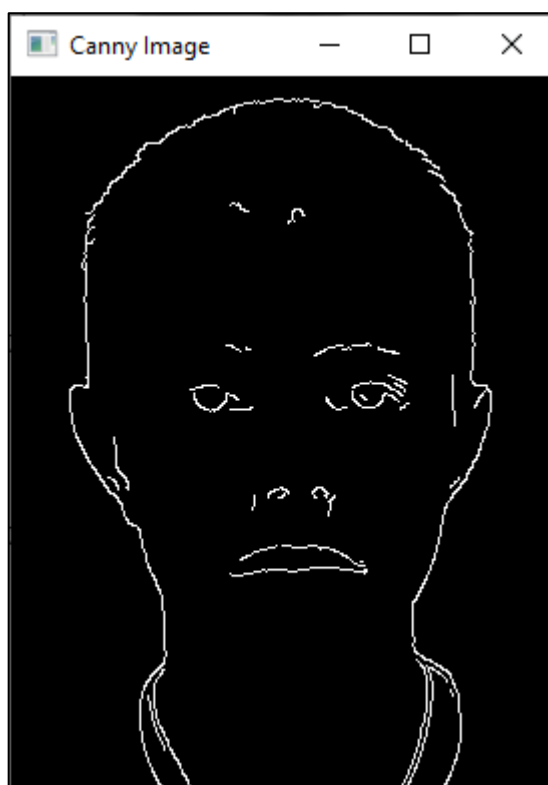


Рис. 2.2.3 – Результат виконання завдання.

		Груницький Д.С.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.6.000 – Лр.8	Арк.
		Голенко М.Ю.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

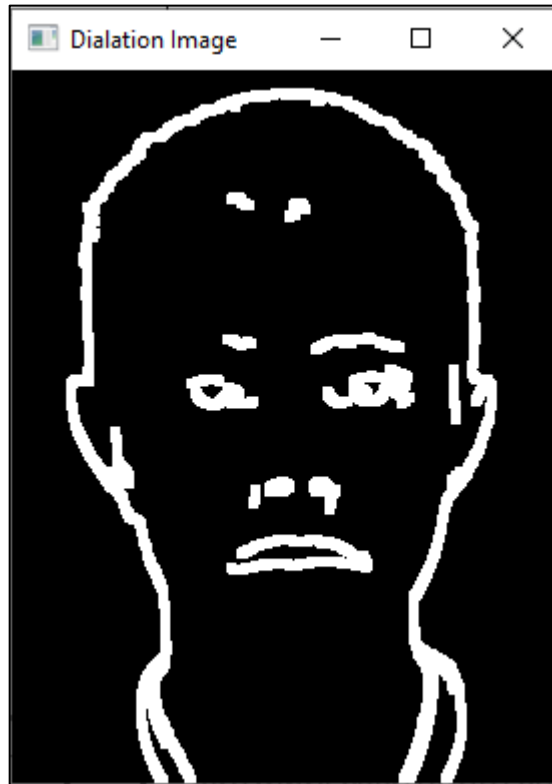


Рис. 2.2.4 – Результат виконання завдання.

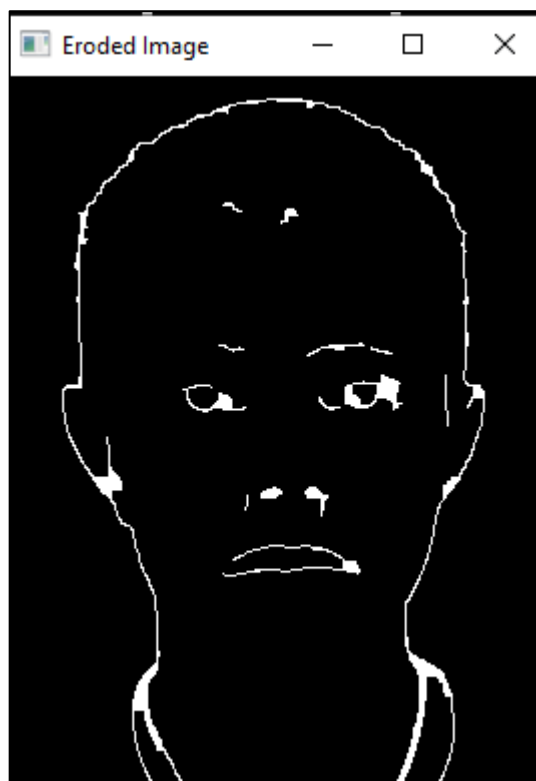


Рис. 2.2.5 – Результат виконання завдання.

		Груницький Д.С.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.6.000 – Лр.8	Арк.
		Голенко М.Ю.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновок

Метод cvtColor:

Цей метод використовується для зміни кольорового простору зображення. У даному випадку, `cv2.COLOR_BGR2GRAY` перетворює кольорове зображення у відтінки сірого. В результаті було отримано чорно-біле (відтінки сірого) зображення, що зручно для подальших обробок, таких як виявлення контурів.

Метод GaussianBlur:

Використовується для розмиття зображення та зменшення шуму. В результаті було отримано розмите зображення, де менше видно дрібних деталей та шуму, що полегшує подальші обчислення.

Метод Canny:

Використовується для виявлення контурів на зображенні. В результаті було отримано зображення, де висвічені контури об'єктів на темному тлі, що допомагає виділити області зі значущими змінами яскравості.

Метод dilate:

Використовується для розширення областей світлішого кольору на зображенні, що допомагає об'єднати близькі контури та заповнити дрібні отвори. В результаті було отримано збільшені та об'єднані контури об'єктів.

Метод erode:

Використовується для стискання областей світлішого кольору на зображенні, що допомагає розділити об'єкти, які знаходяться близько один до одного. В результаті було отримано зменшені та відокремлені контури об'єктів.

Завдання 2.3. Вирізання частини зображення.

		Груницький Д.С.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.6.000 – Лр.8	Арк.
		Голенко М.Ю.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

Лістинг програми:

```
import cv2

img = cv2.imread("Hrunytskyi.jpg")
print(img.shape)
imgResize = cv2.resize(img, (1000, 500))
print(imgResize.shape)
imgCropped = img[20:400, 30:240]
cv2.imshow("Image", img)
cv2.imshow("Image Resize", imgResize)
cv2.imshow("Image Cropped", imgCropped)
cv2.waitKey(0)
```

Результат виконання програми:

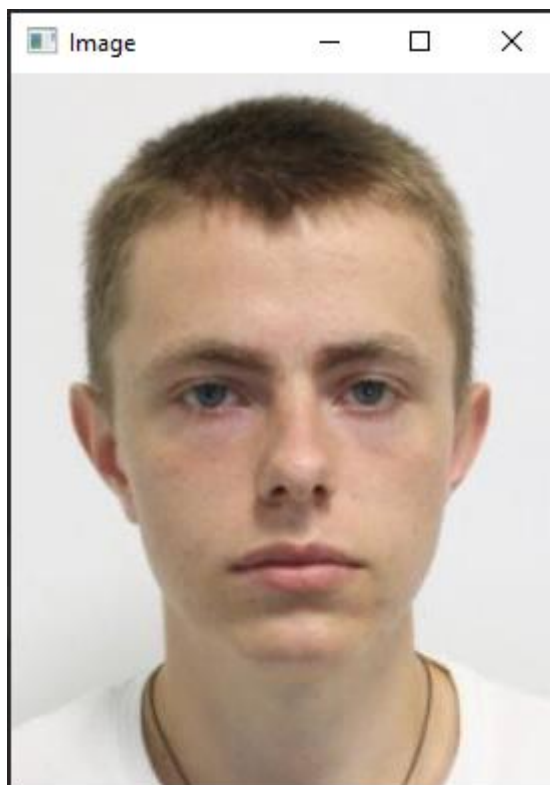


Рис. 2.3.1 – Результат виконання завдання.

		Груницький Д.С.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.6.000 – Лр.8	Арк.
		Голенко М.Ю.				6
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 2.3.2 – Результат виконання завдання.



Рис. 2.3.3 – Результат виконання завдання.

```
(356, 273, 3)
(500, 1000, 3)
```

Рис. 2.3.4 – Результат виконання завдання.

		Груницький Д.С.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.6.000 – Лр.8	Арк.
		Голенко М.Ю.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 2.4. Розпізнавання обличчя на зображенні.

Лістинг програми:

```
import cv2

faceCascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
img = cv2.imread('Hrunytskyi.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. 2.4.1 – Результат виконання завдання.

Висновок

В результаті відбувається перетворення у відтінки сірого, потім використовується класифікатор каскаду Наар для визначення координат обличчя на зображенні. Знайдені координати використовуються для намалювання прямокутника навколо кожного обличчя синім кольором.

		Груницький Д.С.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.6.000 – Лр.8	Арк.
		Голенко М.Ю.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

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

Лістинг програми:

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

img = cv.imread('Hrunytskyi.jpg', 0)
img2 = img.copy()
template = cv.imread('Hrunytskyi_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)
    # 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()
```

Результат виконання програми:

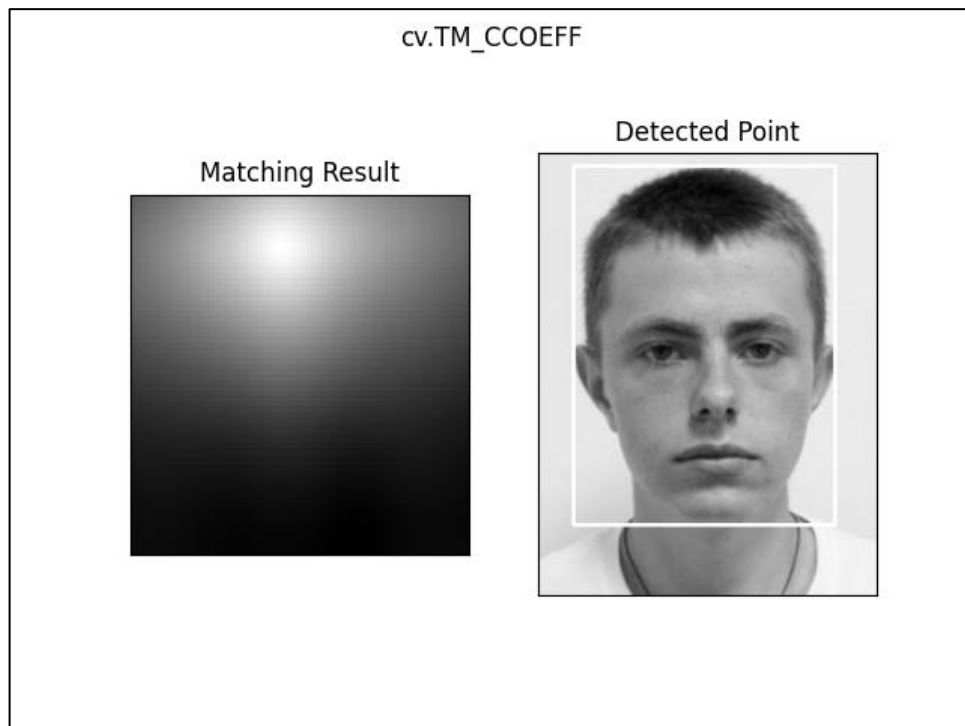


Рис. 2.5.1 – Результат виконання завдання.

		Груницький Д.С.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.6.000 – Лр.8	Арк.
		Голенко М.Ю.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

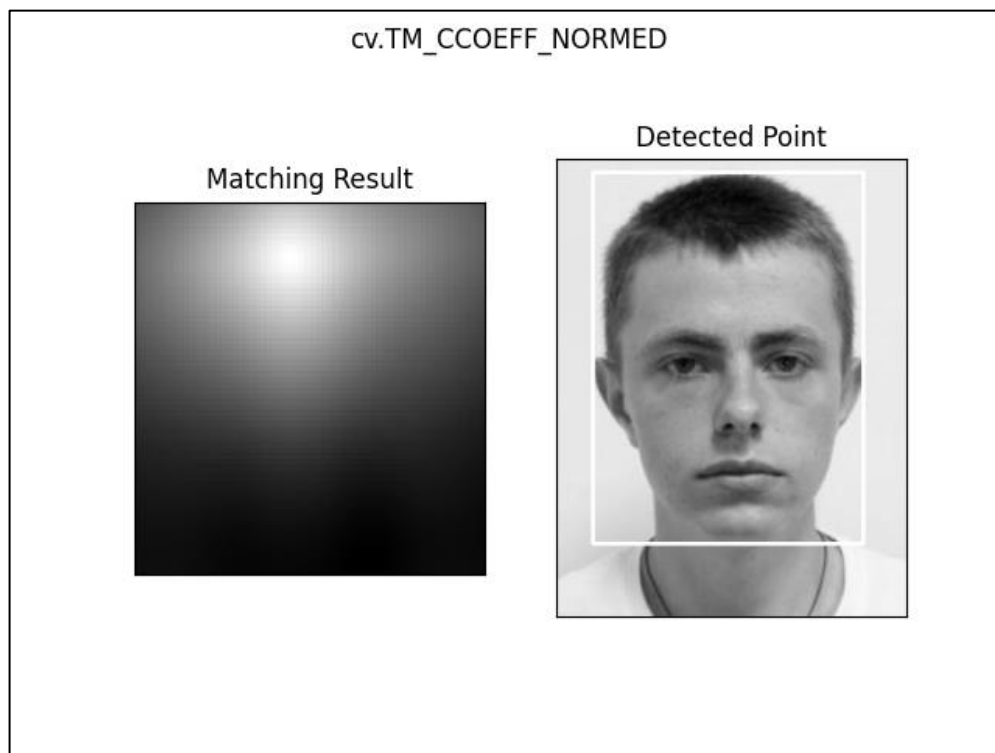


Рис. 2.5.2 – Результат виконання завдання.

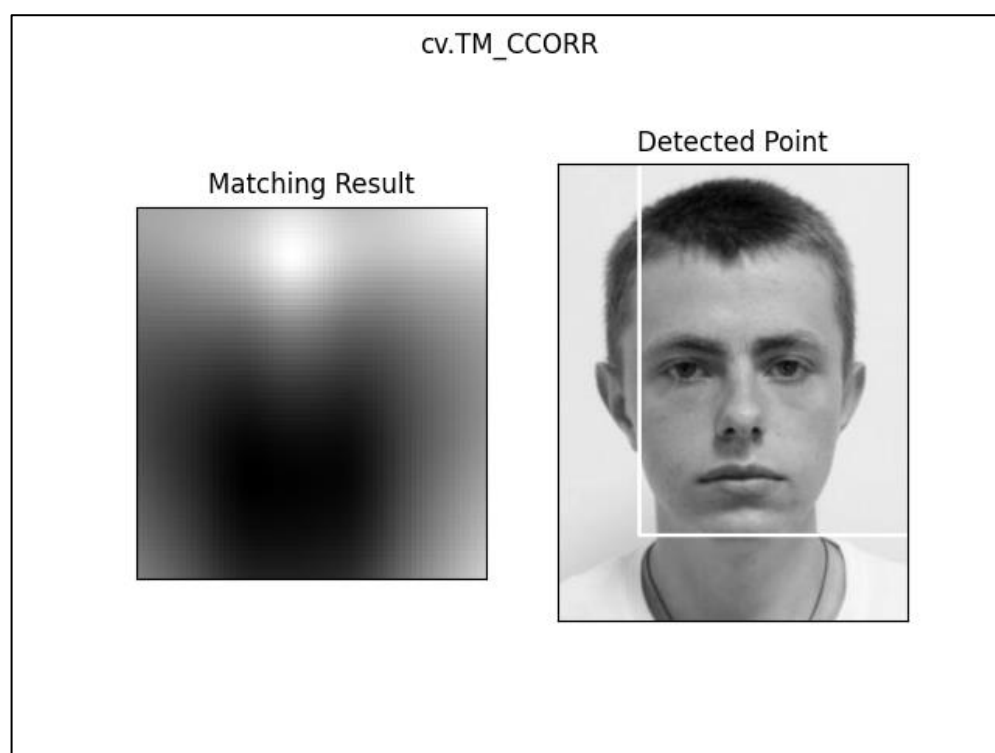


Рис. 2.5.3 – Результат виконання завдання.

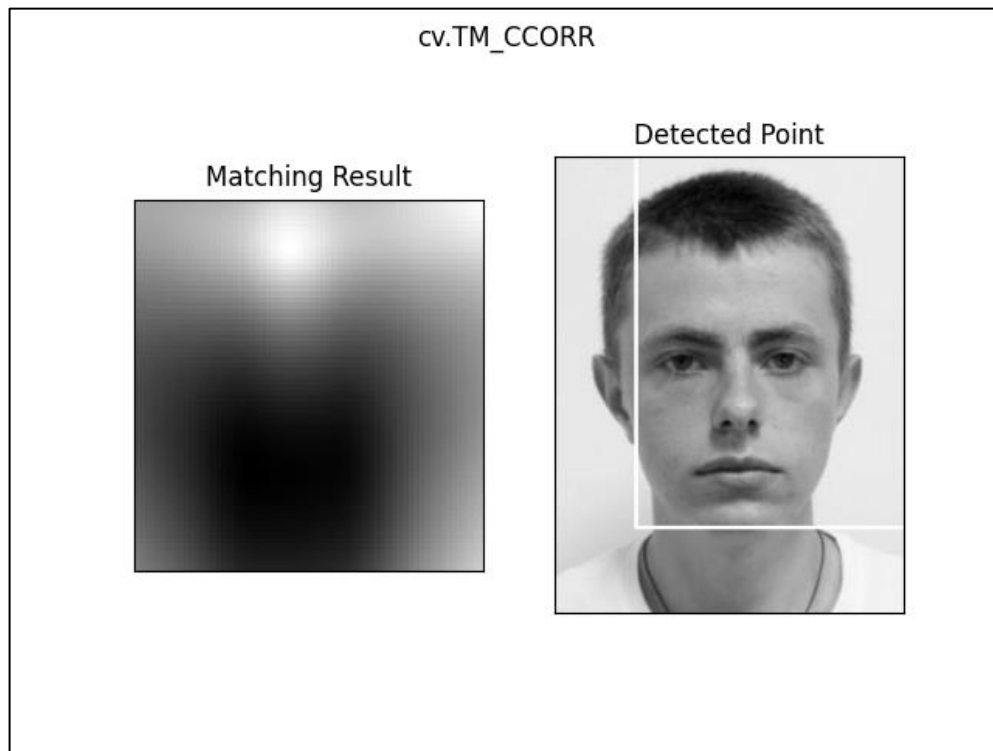


Рис. 2.5.4 – Результат виконання завдання.

Висновок

1. *cv.TM_CCOEFF*: Коефіцієнт кореляції. Вищі значення вказують на краще відповідання.
2. *cv.TM_CCOEFF_NORMED*: Нормалізований коефіцієнт кореляції. Значення в діапазоні від -1 до 1, де 1 вказує на ідеальне відповідання.
3. *cv.TM_CCORR*: Коефіцієнт кореляції. Вищі значення вказують на краще відповідання.
4. *cv.TM_CCORR_NORMED*: Нормалізований коефіцієнт кореляції. Значення в діапазоні від 0 до 1, де 1 вказує на ідеальне відповідання.
5. *cv.TM_SQDIFF*: Квадрат різниці. Нижчі значення вказують на краще відповідання.
6. *cv.TM_SQDIFF_NORMED*: Нормалізований квадрат різниці. Значення в діапазоні від 0 до 1, де 0 вказує на ідеальне відповідання.

Загалом, вибір оптимального методу залежить від конкретного випадку, але в багатьох випадках, нормалізований коефіцієнт кореляції

		Груницький Д.С.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.6.000 – Лр.8	Арк.
		Голенко М.Ю.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

(*cv.TM_CCOEFF_NORMED*) може бути ефективним для вирішення задачі знаходження обличчя через його високу чутливість до шкал та поворотів.

Завдання 2.6. Сегментація зображення алгоритмом водорозподіл.

Лістинг програми:

```
import numpy as np
import cv2

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 програми:

		Груницький Д.С.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.6.000 – Лр.8	Арк.
		Голенко М.Ю.				12
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 2.6.1 – Результат виконання завдання.

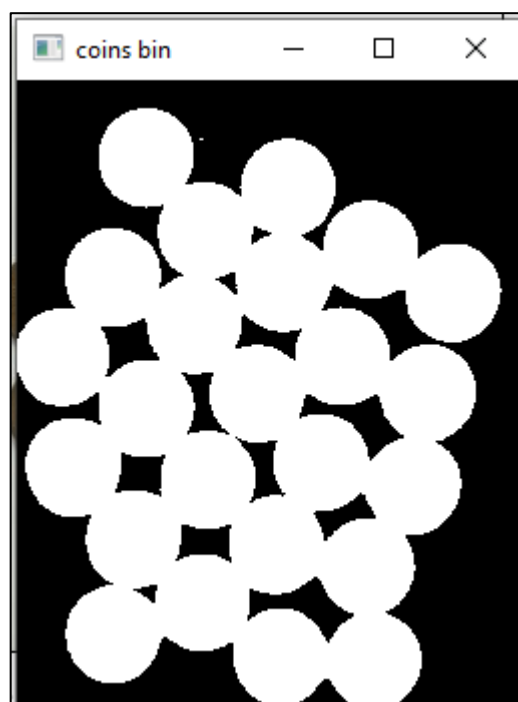


Рис. 2.6.2 – Результат виконання завдання.

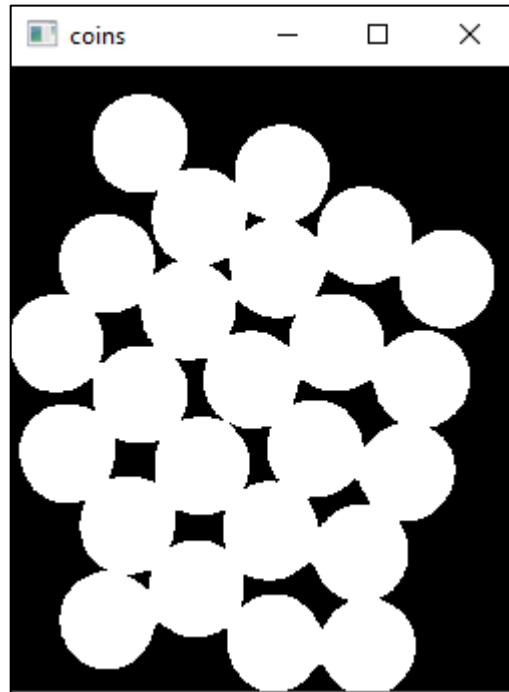


Рис. 2.6.3 – Результат виконання завдання.



Рис. 2.6.4 – Результат виконання завдання.

Висновок

В результаті, в коді використовується метод вододілу для сегментації об'єктів на зображенні та визначення їх границь, що може бути корисним у візуальному аналізі об'єктів на зображенні.

		Груницький Д.С.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.6.000 – Лр.8	Арк.
		Голенко М.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		14

Додаткове завдання

Завдання 2.7. Сегментація зображення.

Лістинг програми:

```
import numpy as np
import cv2

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

#####

shifted = cv2.pyrMeanShiftFiltering(img, 20, 50)
gray = cv2.cvtColor(shifted, cv2.COLOR_BGR2GRAY)
ret, thresh = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)

#####

# Видалення шуму
kernel = np.ones((3, 3), np.uint8)
opening = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, kernel, iterations=3)
# Певна фоновна область
sure_bg = cv2.dilate(opening, kernel, iterations=3)
# Пошук певної області переднього плану
dist_transform = cv2.distanceTransform(sure_bg, cv2.DIST_L2, 5)
# Область переднього плану
ret, sure_fg = cv2.threshold(dist_transform, 0.5 * dist_transform.max(), 255, 0)
# Пошук невідомої області
sure_fg = np.uint8(sure_fg)
unknown = cv2.subtract(sure_bg, sure_fg)

#####

# Маркування міток
ret, markers = cv2.connectedComponents(sure_fg)
# Додаємо один до всіх міток, щоб визначений фон був не 0, а 1
markers += 1
# Позначаємо область невідомого нулем
markers[unknown == 255] = 0

#####

# Алгоритм вододілу
markers = cv2.watershed(img, markers)
# Виділяємо кожну монету із markers різним кольором
for label in range(2, ret + 1): # Починаємо з 2, оскільки фон має значення 1
    img[markers == label] = np.random.randint(0, 255, 3)

#####

cv2.imshow("coins_markers", img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Результат виконання програми:

		Груницький Д.С.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.6.000 – Лр.8	Арк.
		Голенко М.Ю.				15
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 2.7.1 – Результат виконання завдання.

		Груницький Д.С.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.6.000 – Лр.8	Арк.
		Голенко М.Ю.				16
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 2.7.2 – Результат виконання завдання.

Висновок

Код спрямований на виділення різних монет на зображенні однієї купюри. Основні кроки включають в себе використання фільтра `pyrMeanShift` для зменшення кількості кольорів на зображенні, визначення порогового значення за допомогою фільтра `OTSU`, видалення шуму за допомогою морфологічних операцій, визначення фону та переднього плану, використання алгоритму вододілу для розділення областей та накладення кольорових міток на розділені області. Кожна монета на зображенні маркується випадковим кольором, щоб виділити її на фоні.

Посилання на репозиторій: https://github.com/dmytrohrunyskyi/Lab8_AI.git

Висновки по лабораторній роботі: в ході виконання лабораторної роботи використовуючи спеціалізовані бібліотеки та мову програмування Python навчився обробляти зображення за допомогою бібліотеки OpenCV.

		Груницький Д.С.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.6.000 – Лр.8	Арк.
		Голенко М.Ю.				17
Змн.	Арк.	№ докум.	Підпис	Дата		