

## 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.

```
import cv2
# LOAD AN IMAGE USING 'IMREAD'
img = cv2.imread("Maks12.jpg")
# DISPLAY
cv2.imshow("Maks1", img)
cv2.waitKey(0)
```

Рис 1. Код файлу LR\_8\_task\_1



Рис 2. Результат файлу LR\_8\_task\_1

Завдання №2: Дослідження перетворень зображення.

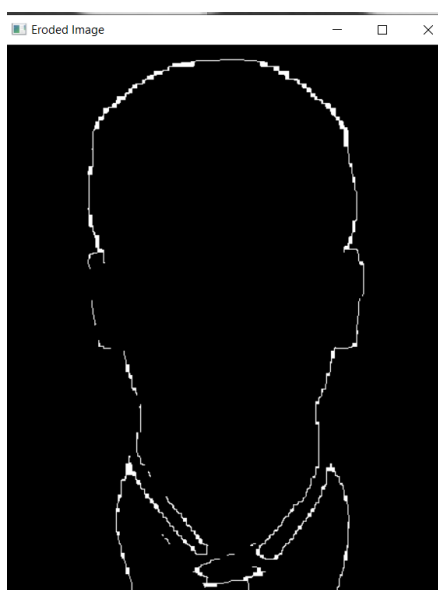
					ДУ «Житомирська політехніка».20.121.12.							
Змн.	Арк.	№ докум.	Підпис	Дата								
Розроб.		Надворний М.Ю.							Літ.	Арк.	Аркуші	
Перевір.		Філіпов В.О.									1	13
Керівник									ФІКТ Гр. ІПЗк-20-1			
Н. контр.												
Зав. каф.												

```

import cv2
import numpy as np
img = cv2.imread("Maks12.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 3. Код файлу LR\_8\_task\_2



		Надворний М.Ю			ДУ «Житомирська політехніка».20.121.12 – Лр8	Арк.
		Філіпов В.О.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

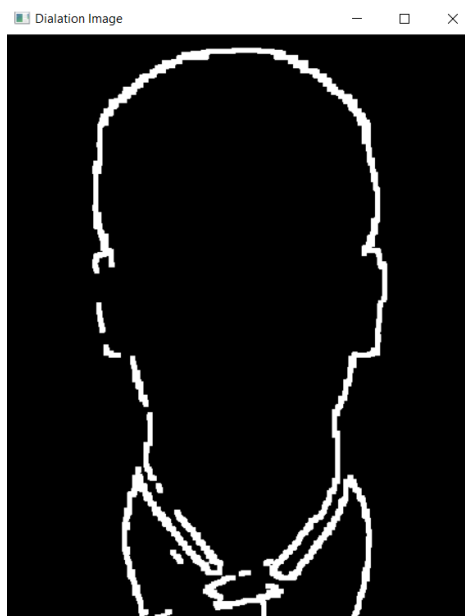


Рис 4. Результат файлу LR\_8\_task\_2

		Надворний М.Ю			ДУ «Житомирська політехніка».20.121.12 – Лр8	Арк.
		Філіпов В.О.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

Метод **cvtColor** використовується для перетворення зображення з одного колірного простору в інший, в результаті його використання було отримано зображення у градації сірого кольору.

Метод **GaussianBlur** використовується для застосування Гаусового згладжування до зображення, в результаті його використання було отримано замилене зображення.

Метод **Canny** використовується для виявлення країв зображення, в результаті його використання було отримано зображення з контурами обличчя.

Метод **dilate** використовується для зменшення особливостей зображення, в результаті його використання було отримано зображення з контурами обличчя.

Метод **erode** використовується для підкреслення рис, в результаті його використання було отримано зображення з розмитим контуром обличчя.

Завдання №3: Вирізання частини зображення.

```
import cv2
import numpy as np
img = cv2.imread("Maks12.jpg")
print(img.shape)
imgResize = cv2.resize(img, (1000, 500))
print(imgResize.shape)
imgCropped = img[75:400, 30:350]
cv2.imshow("Image", img)
# cv2.imshow("Image Resize", imgResize)
cv2.imshow("Image Cropped", imgCropped)
cv2.waitKey(0)
```

Рис 5. Код файлу LR\_8\_task\_3

		Надворний М.Ю			ДУ «Житомирська політехніка».20.121.12 – Лр8	Арк.
		Філіпов В.О.				4
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис 6. Результат файлу LR\_8\_task\_3

Завдання №4: Розпізнавання обличчя на зображенні.

```
import cv2
faceCascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
img = cv2.imread('Maks12.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 7. Код файлу LR\_8\_task\_4



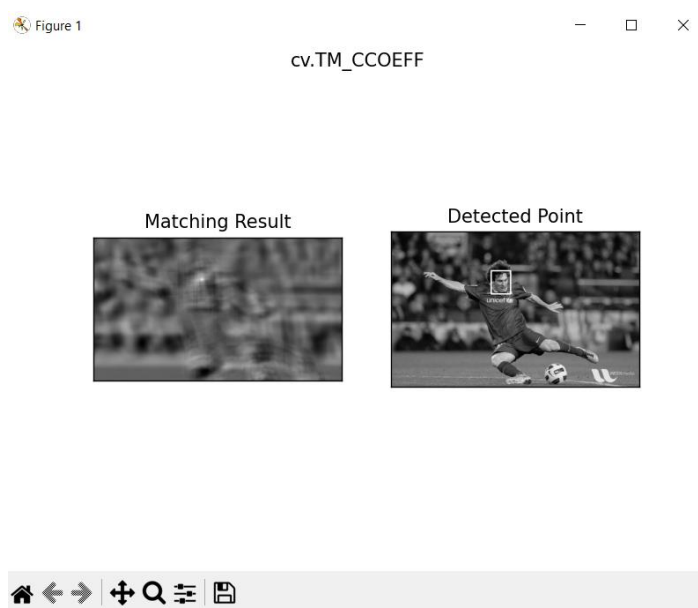
Рис 8. Результат файлу LR\_8\_task\_4

		Надворний М.Ю			ДУ «Житомирська політехніка».20.121.12 – Лр8	Арк.
		Філіпов В.О.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

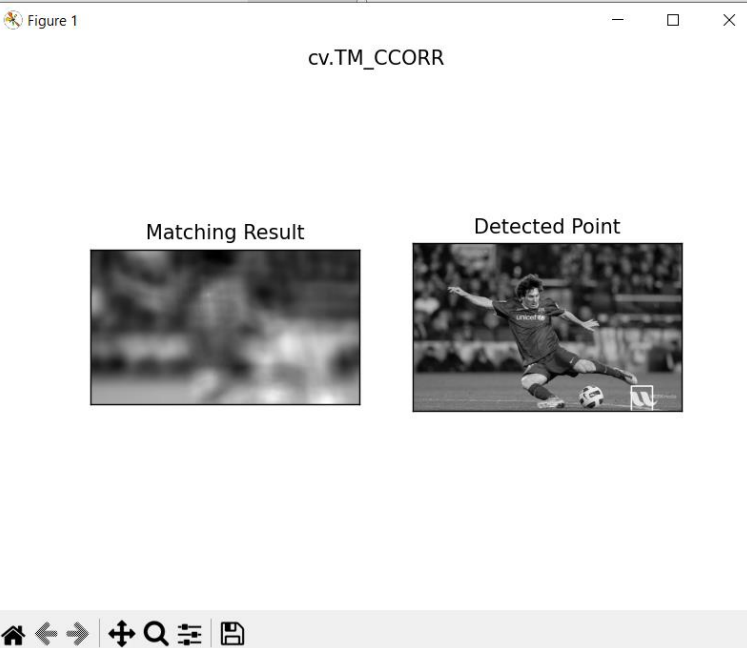
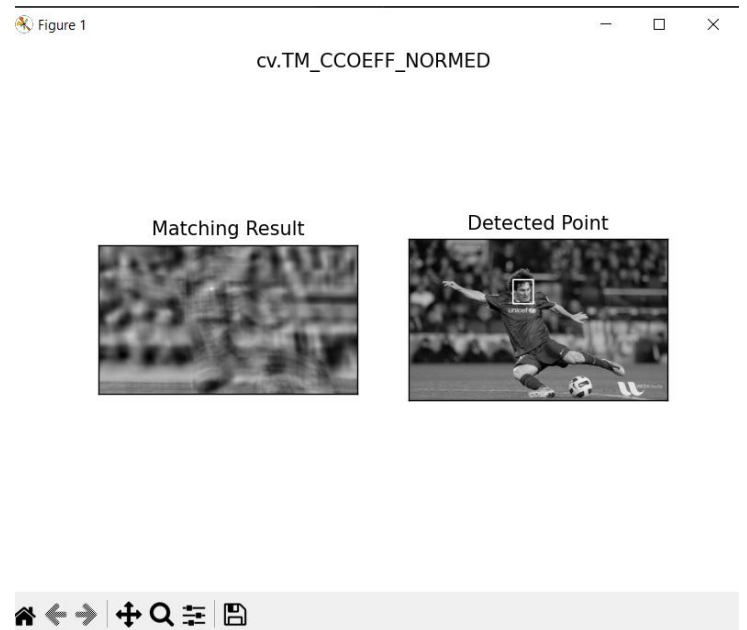
## Завдання №5: Розпізнавання об'єктів на зображенні за допомогою методів зіставлення шаблонів (Template Matching).

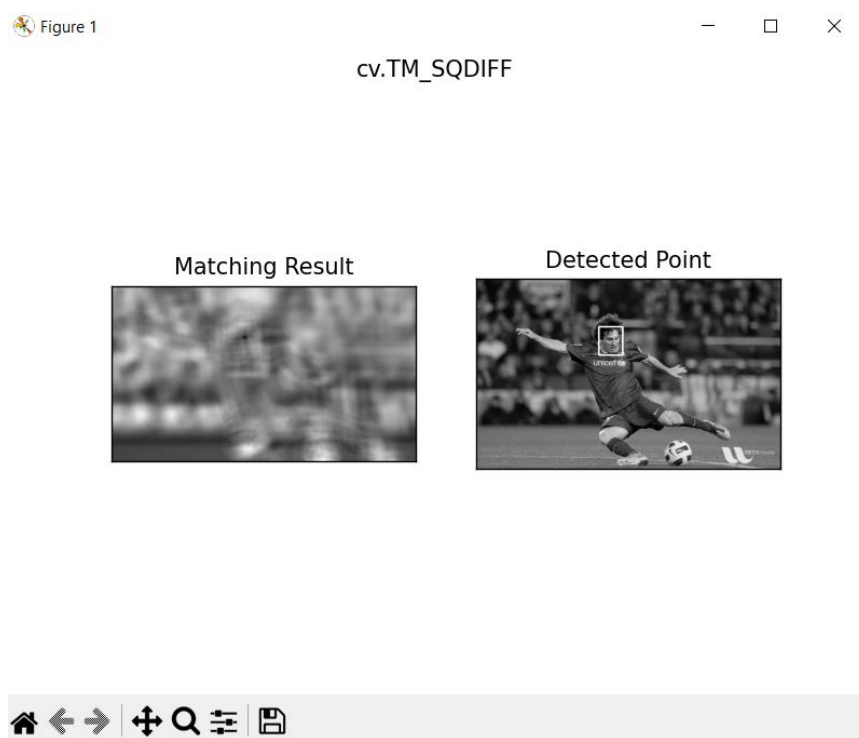
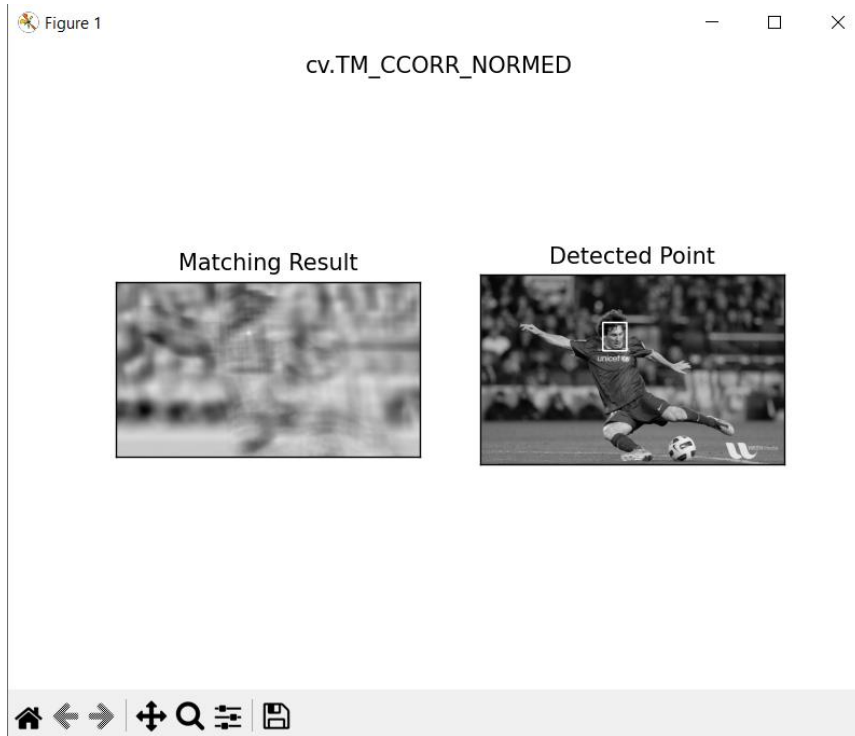
```
import cv2 as cv
from matplotlib import pyplot as plt
img = cv.imread('messi_full.jpg', 0)
img2 = img.copy()
template = cv.imread('messi_face.jpg', 0)
w, h = template.shape[::-1]
# All the 6 methods for comparison in a list
methods = ['cv.TM_CCOEFF', 'cv.TM_CCOEFF_NORMED', 'cv.TM_CCORR',
           'cv.TM_CCORR_NORMED', 'cv.TM_SQDIFF', 'cv.TM_SQDIFF_NORMED']
for meth in methods:
    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 9. Код файлу LR\_8\_task\_5



		Надворний М.Ю			ДУ «Житомирська політехніка».20.121.12 – Лр8	Арк.
		Філіпов В.О.				6
Змн.	Арк.	№ докум.	Підпис	Дата		





		Надворний М.Ю			ДУ «Житомирська політехніка».20.121.12 – Лр8	Арк.
		Філіпов В.О.				8
Змн.	Арк.	№ докум.	Підпис	Дата		





Рис 10. Результат файлу LR\_8\_task\_5

cv.TM\_CCORR:

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

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\_CCORR\_NORMED:

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

		Надворний М.Ю			ДУ «Житомирська політехніка».20.121.12 – Лр8	Арк.
		Філіпов В.О.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

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

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:

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

На мою думку, cv2.TM\_SQDIFF – найкращий метод для поставленої задачі, бо мінімальне значення дає найкращий збіг.

Завдання №6: Сегментація зображення алгоритмом водорозподілу. Лістинг програми:

		Надворний М.Ю			ДУ «Житомирська політехніка».20.121.12 – Лр8	Арк.
		Філіпов В.О.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

```

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]

```

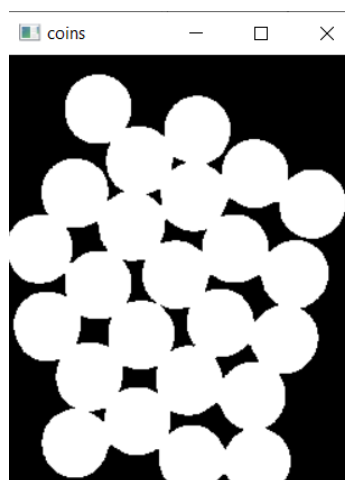
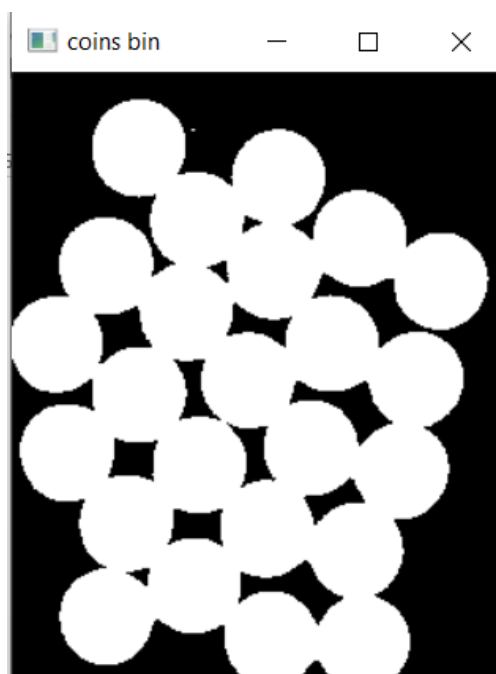
```

# 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 11. Код файлу LR\_8\_task\_6

		Надворний М.Ю			ДУ «Житомирська політехніка».20.121.12 – Лр8	Арк.
		Філіпов В.О.				11
Змн.	Арк.	№ докум.	Підпис	Дата		



		Надворний М.Ю			ДУ «Житомирська політехніка».20.121.12 – Лр8	Арк.
		Філіпов В.О.				12
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис 12. Результат файлу LR\_8\_task\_6

Після виконання програми для більшості монет було правильно відсегментовано області, але проблеми виникли для областей, де монети торкаються одне одного. Через що певні області були не зовсім валідно визначені.

Висновок: Після виконання лаби я дослідив оброблення зображення за допомогою бібліотеки OpenCV, використовуючи спеціалізовані бібліотеки та мову програмування Python. Було розглянуто сегментацію зображення алгоритмом водорозподілу. Було проаналізовано розпізнавання об'єктів на зображенні за допомогою методів зіставлення шаблонів.

Посилання на гітхаб: <https://github.com/Max1648/-Artificial-Intelligence3>

		Надворний М.Ю			ДУ «Житомирська політехніка».20.121.12 – Лр8	Арк.
		Філіпов В.О.				13
Змн.	Арк.	№ докум.	Підпис	Дата		