**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**

**Посилання на GitHub:** <https://github.com/Max2002/AI_IPZ-19-3_LMV>

**Завдання 1.** Завантаження зображень та відео в OpenCV

Лістинг LR\_8\_Task\_1.py:

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("Liashuk.png")  
cv2.imshow("Liashuk ",img)  
cv2.waitKey(0)

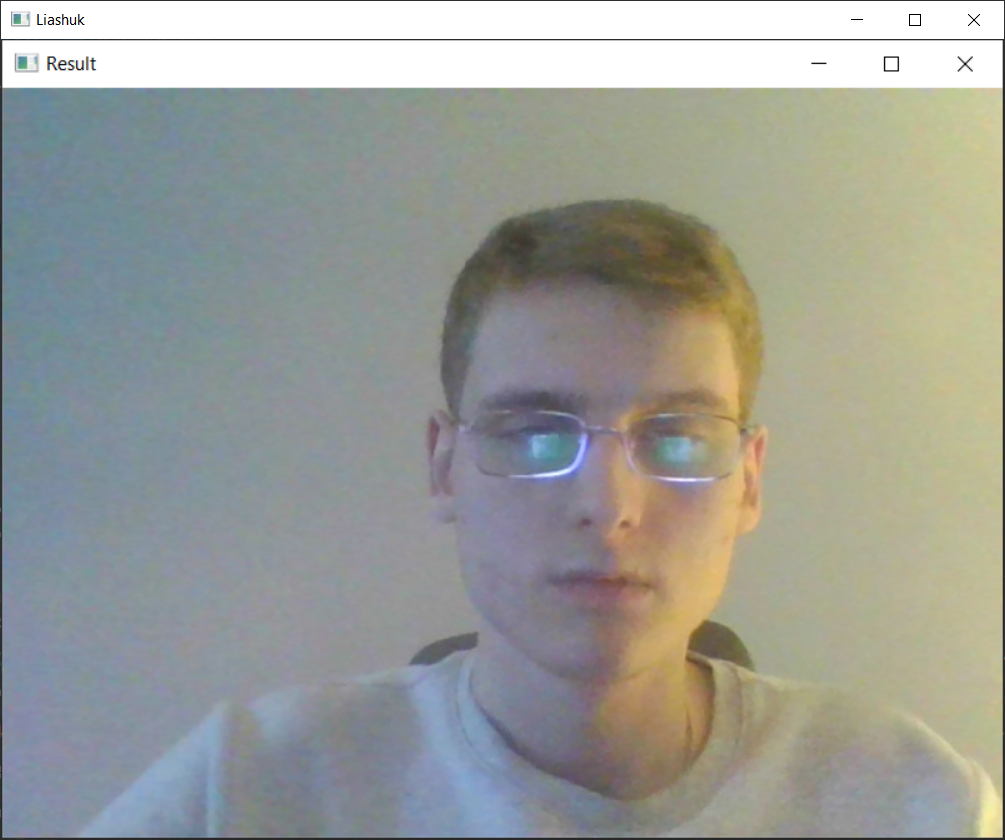


Рис.1. Отримане зображеня

**Завдання 2.** Дослідження перетворень зображення

Лістинг LR\_8\_Task\_2.py:

import cv2  
import numpy as np  
  
img = cv2.imread("Liashuk.png")  
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,50,50)  
imgDialation = cv2.dilate(imgCanny,kernel,iterations=1)  
imgEroded = cv2.erode(imgDialation,kernel,iterations=1)  
  
cv2.imshow("Liashuk ",img)  
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)

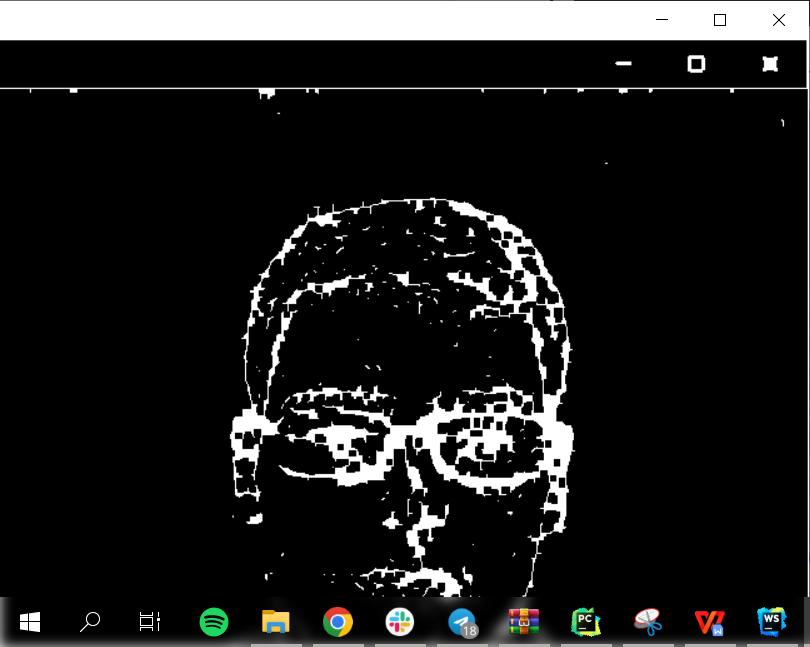
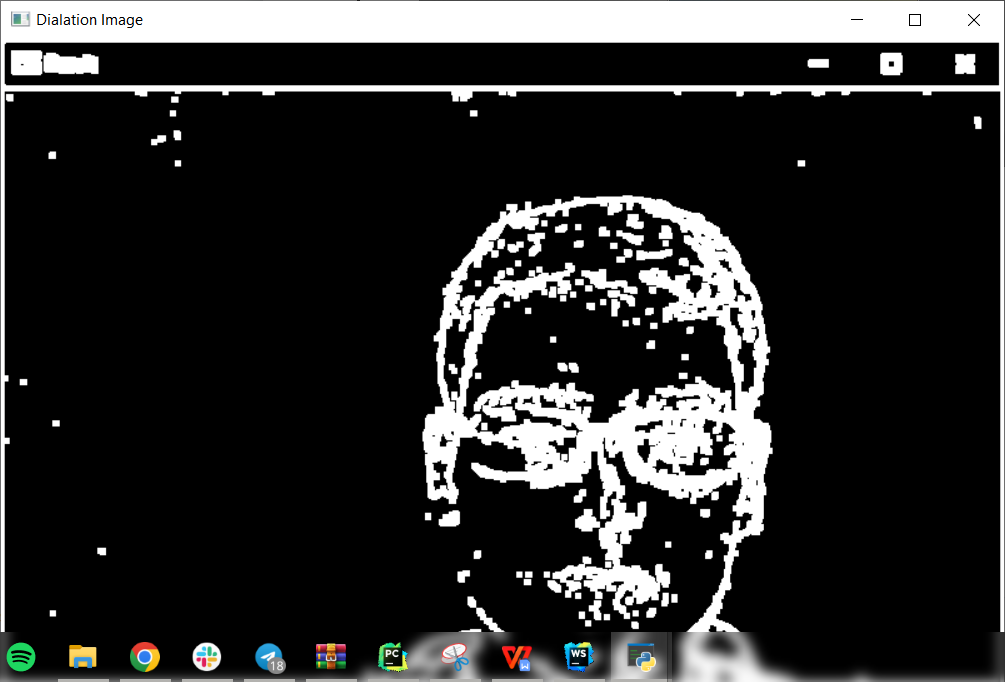
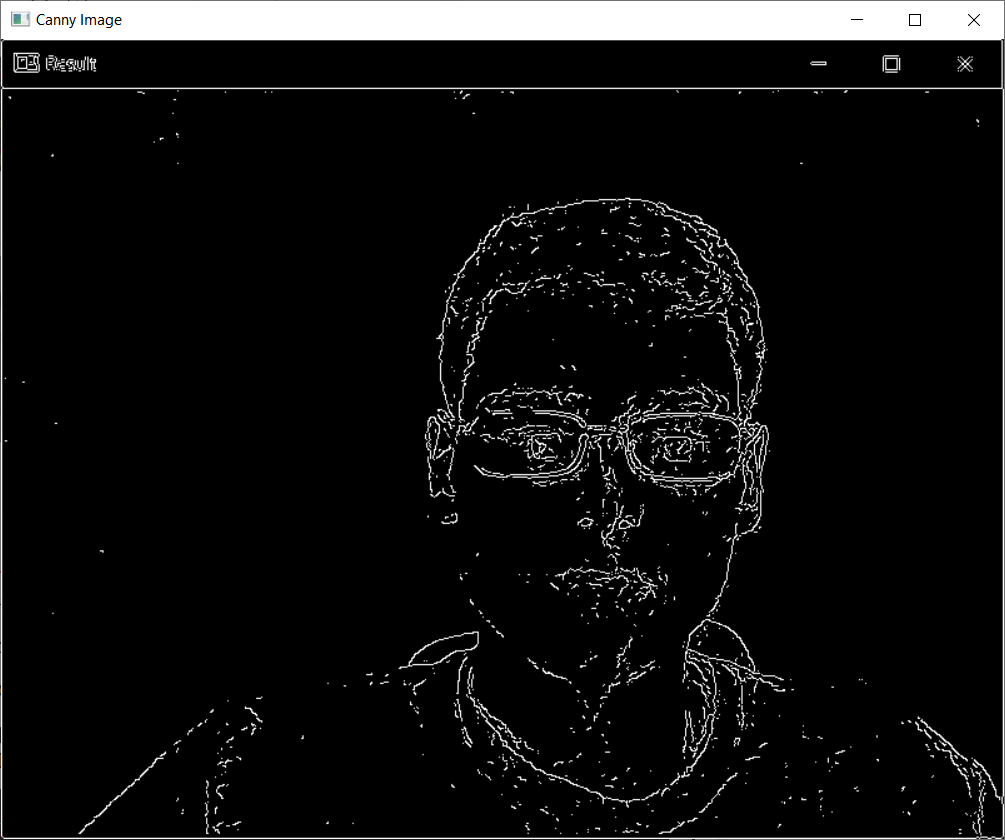
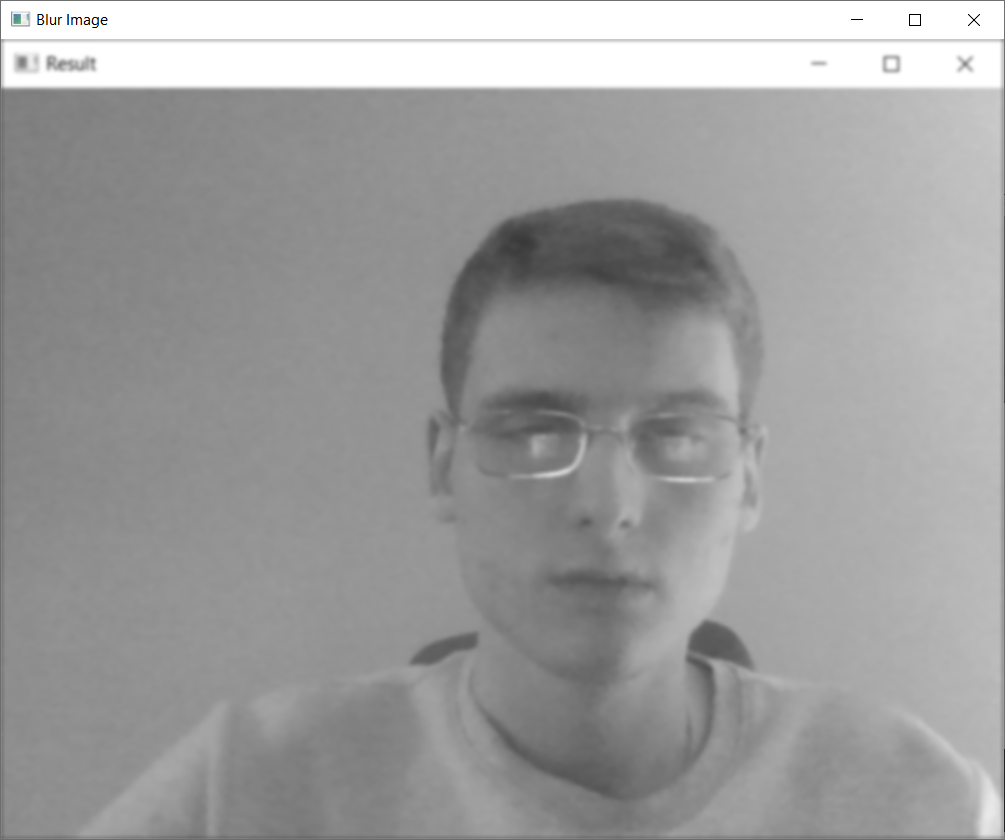
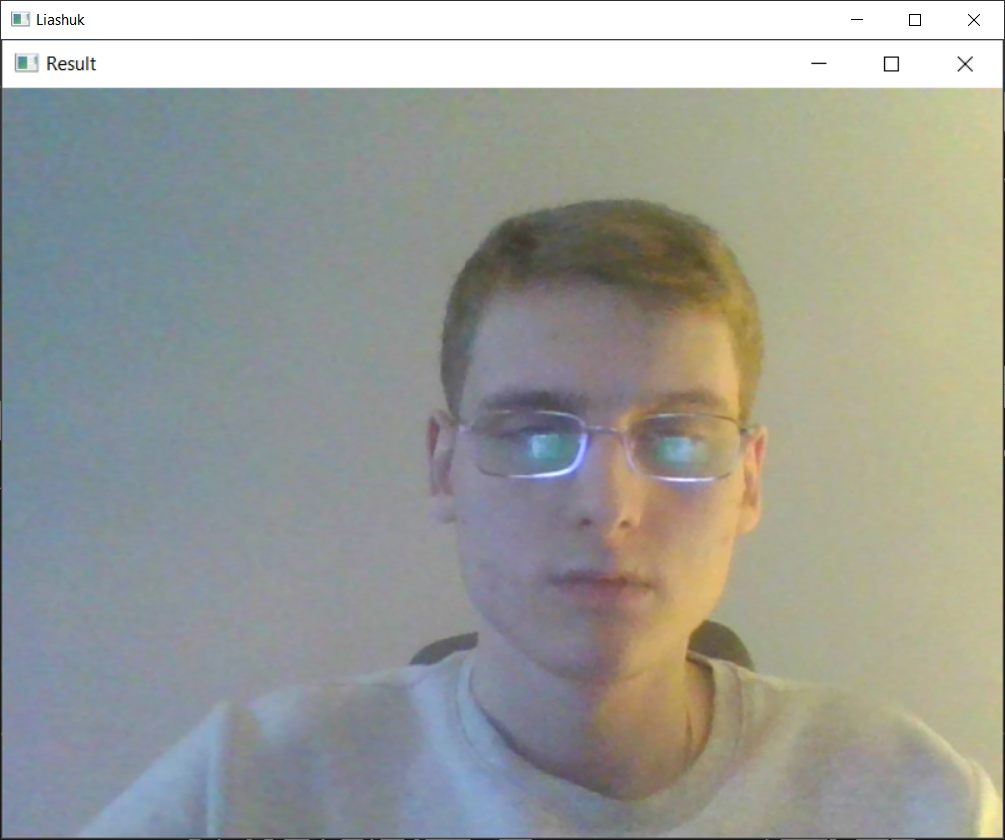
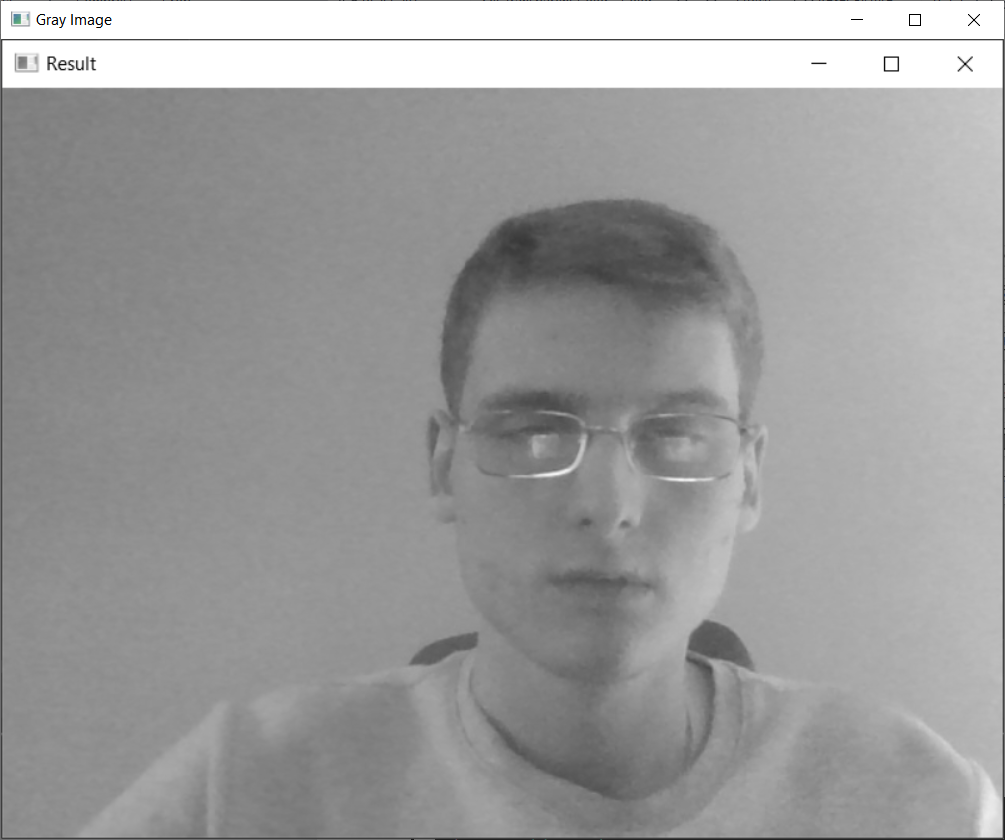


Рис.2. Результат роботи програми

cvtColor – чорно-біле зображеня

GaussianBlur – розмиття

Canny – знаходження контурів

Dilate – додає пікселі та робить контур ширшим

Erode – прибирає пікселі та робить контур тоншим

**Завдання 3.** Вирізання частини зображення

Лістинг LR\_8\_Task\_3.py:

import cv2  
import numpy as np  
  
img = cv2.imread("Liashuk2.png")  
print(img.shape)  
  
imgResize = cv2.resize(img, (1000, 500))  
print(imgResize.shape)  
  
imgCropped = img[40:420, 90:380]  
  
cv2.imshow("Image Cropped", imgCropped)  
  
cv2.waitKey(0)



Рис.3. Результат роботи програми

**Завдання 4.** Вирізання частини зображення

Лістинг LR\_8\_Task\_4.py:

import cv2  
  
faceCascade= cv2.CascadeClassifier("haarcascade\_frontalface\_default.xml")  
img = cv2.imread('Liashuk2.jpg')  
img = img[1:600, 1:1280]  
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)

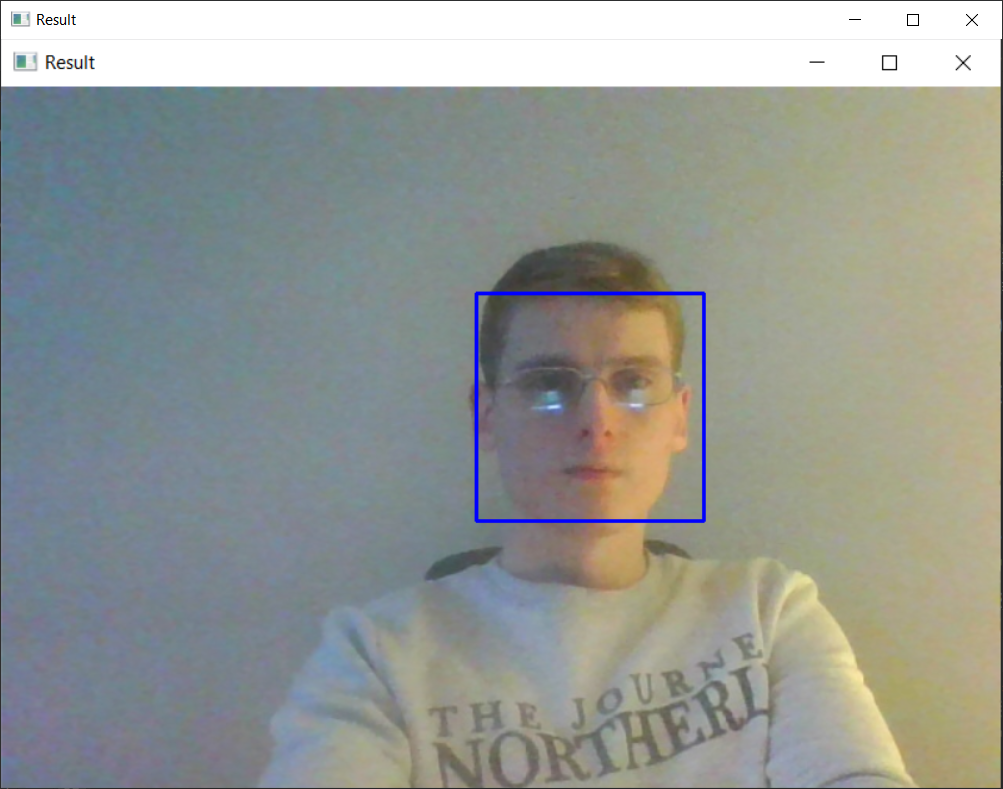
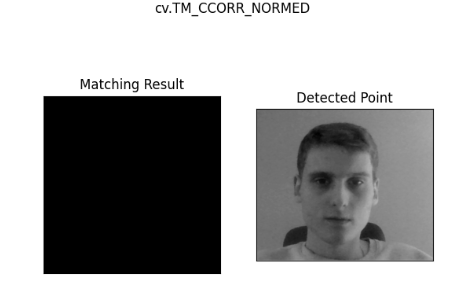
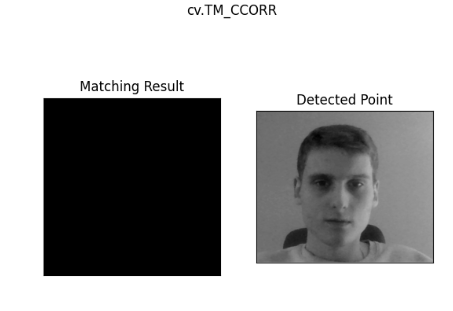
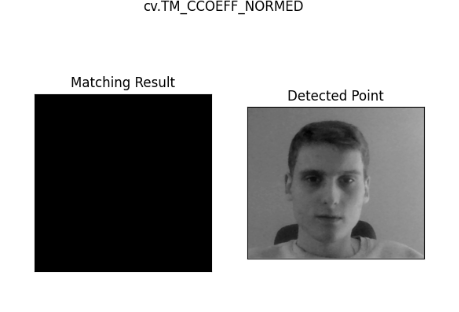


Рис.4. Результат роботи програми(було зроблено нову фотографію, тому що на старій обличчя не розпізнавалось)

**Завдання 5.** Розпізнавання об’єктів на зображенні за допомогою методів  
зіставлення шаблонів

Лістинг LR\_8\_Task\_5.py:

import cv2 as cv  
import numpy as np  
from matplotlib import pyplot as plt  
  
img = cv.imread('Liashuk2.PNG', 0)  
img2 = img.copy()  
template = cv.imread('Liashuk-face.PNG', 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()



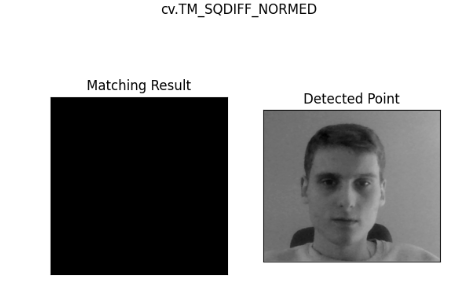


Рис.5. Результат роботи програми

**Завдання 6.** Сегментація зображення алгоритмом водорозподілу

Лістинг LR\_8\_Task\_6.py:

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)  
  
kernel = np.ones((3, 3), np.uint8)  
opening = cv2.morphologyEx(thresh, cv2.MORPH\_OPEN, kernel, iterations=2)  
  
sure\_bg = cv2.dilate(opening, kernel, iterations=3)  
  
dist\_transform = cv2.distanceTransform(opening, cv2.DIST\_L2, 5)  
ret, sure\_fg = cv2.threshold(dist\_transform, 0.7 \* dist\_transform.max(), 255, 0)  
  
sure\_fg = np.uint8(sure\_fg)  
unknown = cv2.subtract(sure\_bg, sure\_fg)  
cv2.imshow("coins ", opening)  
cv2.waitKey(0)  
  
ret, markers = cv2.connectedComponents(sure\_fg)  
  
markers = markers + 1  
  
markers[unknown == 255] = 0  
markers = cv2.watershed(img, markers)  
img[markers == -1] = [255, 0, 0]  
  
cv2.imshow("coins\_markers",img)  
cv2.waitKey(0)

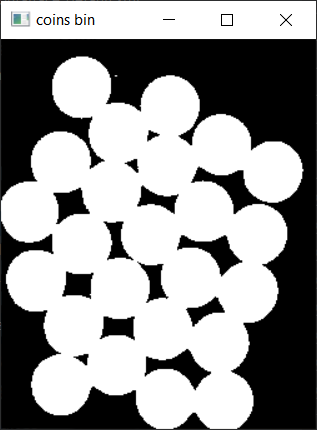
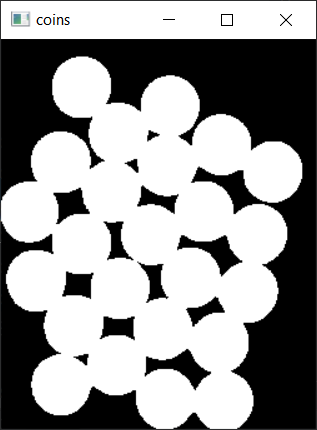
   

Рис.6. Результат роботи програми

***Висновок:*** в ході виконання лабораторної роботи було отримано навички в обробці зображення за допомогою бібліотеки OpenCV. Використовуючи спеціалізовані бібліотеки і мову програмування Python.