## **Протокол**

**Мета**: Конвертація кольорового зображення в чорно-біле, бінаризація за методом Крістіана та вирізання об'єкту за допомогою отриманої маски на чотирьох зображеннях.

### **Кроки виконання:**

1. Імпортуємо необхідні бібліотеки: `PIL` для роботи з зображеннями та `numpy` для обчислень.

```

from PIL import Image

import numpy as np

```

2. Відкриваємо вхідне кольорове зображення та визначаємо шлях для збереження чорно-білого та бінаризованого зображення.

```

input\_image\_path = "ph3.jpg"

output\_image\_path = "чб\_зображення.jpg"

output\_image\_path\_binary = 'бінарізоване\_зображення.jpg'

image = Image.open(input\_image\_path)

```

3. Отримуємо розміри вхідного зображення.

```

width, height = image.size

```

4. Створюємо нове чорно-біле зображення такого ж розміру.

```

bw\_image = Image.new("L", (width, height))

```

5. Обчислюємо яскравість кожного пікселя у вхідному кольоровому зображенні і перетворюємо його у градацію сірого кольору.

```

for x in range(width):

for y in range(height):

pixel\_color = image.getpixel((x, y))

grayscale = int(0.36 \* pixel\_color[0] + 0.53 \* pixel\_color[1] + 0.11 \* pixel\_color[2])

bw\_image.putpixel((x, y), grayscale)

```

6. Зберігаємо чорно-біле зображення.

```

bw\_image.save(output\_image\_path)

```

7. Перетворюємо чорно-біле зображення у масив NumPy для обчислення гістограми яскравості.

```

bw\_array = np.array(bw\_image)

```

8. Обчислюємо гістограму яскравості.

```

histogram = np.histogram(bw\_array, bins=256, range=(0, 256))[0]

```

9. Розраховуємо порогове значення за методом Крістіана.

```

total = bw\_array.size

threshold = 0

for t in range(1, 255):

w0 = sum(histogram[:t]) / total

w1 = sum(histogram[t:]) / total

u0 = sum(i \* histogram[i] for i in range(t)) / sum(histogram[:t])

u1 = 0

if sum(histogram[t:]) != 0:

u1 = sum(i \* histogram[i] for i in range(t, 256)) / sum(histogram[t:])

var = w0 \* w1 \* (u0 - u1) \*\* 2

if var > threshold:

threshold = var

best\_threshold = t

```

10. Бінаризуємо зображення, використовуючи порогове значення.

```

binary\_image = (bw\_array > best\_threshold) \* 255

```

11. Створюємо зображення з масива NumPy.

```

binary\_image = Image.fromarray(binary\_image.astype(np.uint8))

```

12. Зберігаємо бінаризоване зображення.

```

binary\_image.save(output\_image\_path\_binary)

```

13. Вирізаємо об'єкт з вихідного зображення за допомогою маски.

```

object\_image = Image.new("RGB", image.size)

for x in range(image.width):

for y in range(image.height):

if binary\_image.getpixel((x, y)) == 255: # Пиксель у масці білий (об'єкт)

pixel\_color = image.getpixel((x, y))

object\_image.putpixel((x, y), pixel\_color)

```

14. Зберігаємо вирізаний об'єкт.

```

object\_image.save("вирізаний\_об'єкт.jpg")

```

15. Закриваємо всі відкриті зображення.

```

image.close()

bw\_image.close()

binary\_image.close()

object\_image.close()

```