

**Міністерство освіти і науки України
Національний технічний університет України «КПІ» імені Ігоря Сікорського
Кафедра обчислювальної техніки ФІОТ**

**ЗВІТ
з лабораторної роботи №5
з навчальної дисципліни «Computer Vision»**

Тема:

**ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ ПОКРАЩЕННЯ ЯКОСТІ ЦИФРОВИХ
ЗОБРАЖЕНЬ ДЛЯ ЗАДАЧ COMPUTER VISION**

Виконав:

Студент 3 курсу кафедри ІПІ ФІОТ,
Навчальної групи ІП-11
Панченко С.В.

Перевірив:

Професор кафедри ОТ ФІОТ
Писарчук О.О.

Київ 2024 р.

I. Мета:

Дослідити принципи та особливості практичного застосування технологій сегментації та кластеризації цифрових зображень для задач Computer Vision з використанням спеціалізованих програмних бібліотек.

II. Завдання.

Розробити програмний скрипт, що забезпечує цифрову обробку зображень для розрізнення та ідентифікації обраних об'єктів на цифровому знімку земної поверхні з низькою роздільною здатністю за цифровими зображеннями відкритих джерел даних дистанційного зондування Землі (ДЗЗ) із космосу.

Порядок організаційних дій та функціонал програмного скрипта:

1. Обрати район спостереження та об'єкти ідентифікації – однакові за оперативними та високоточними джерелами даних ДЗЗ – див. табл.

2. Отримати цифрові растрові знімки обраного району земної поверхні з оперативних та високоточних джерел даних ДЗЗ із збереженням їх у файлі відповідного типу.

3. За допомогою програмного скрипта провести кольорову корекцію та / або фільтрацію даних ДЗЗ від оперативних та високоточних джерел відносно об'єкта ідентифікації.

4. Реалізувати програмно кольорову кластеризацію покращених в п.3 зображень об'єкта ідентифікації на даних ДЗЗ від оперативних та високоточних джерел.

5. Здійснити сегментацію кластеризованих в п.4 цифрових зображень від оперативних та високоточних джерел даних ДЗЗ із виділенням контуру об'єкта ідентифікації.

6. Шляхом візуального та / або програмного порівняння контурів обраних об'єктів векторизованих зображень від оперативних та високоточних джерел даних ДЗЗ здійснити ідентифікацію цих об'єктів.

Вимоги та обмеження:

Об'єктами для ідентифікації можуть бути площадні або точкові об'єкти: лісові насадження, вирубки лісів, водойми, площі ерозії поверхні Землі, сільськогосподарські угіддя, посівні площі, будівлі, автівки, техногенні / критичні об'єкти.

Ідентифікація – полягає у встановленні лінгвістичної назви об'єкта та здійснюється за геометрією контура.

Всі процеси обробки повинні бути спрямовані та реалізовані відносно об'єкта ідентифікації.

Варіант (місяць народження) 06.03.2004	Технічні умови	Графічна Алгоритми фігура.
1-6	1. Оперативні: https://apps.sentinel-hub.com/eo-browser/?zoom=14&lat=52.04212&lng=29.27444&the	Район спостереження – обрати самостійно. Об'єкти ідентифікації – обрати самостійно.

meId=WILDFIRES- NORMAL-MODE&visualizationUrl=https %3A%2F%2Fservices.sentinel- hub.com%2Fogc%2Fwms%2Faae18701- 6b25-4001-8b2a- b98a1b3806c1&datasetId=S2L2A&fromTime =2022-03- 16T00%3A00%3A00.000Z&toTime=2022- 03- 16T23%3A59%3A59.999Z&layerId=1_FALS E-COLOR 2. Високоточні: https://www.google.com.ua/maps	Дата оперативних даних – обрати самостійно. Метод і технологія кластеризації / сегментації – повинні забезпечувати можливість розрізнення та ідентифікацію обраних об'єктів спостереження.
--	--

III. Результати виконання лабораторної роботи.

3.1. Синтезована математична модель перетворень графічних об'єктів відповідно до завдання.

У цій роботі було застосовано перетворення колірного простору з RGB на HSV. Колірна модель HSV складається з трьох компонентів:

- Hue (колірний тон) - визначає основний відтінок кольору;
- Saturation (насиченість) - характеризує чистоту або інтенсивність кольору;
- Value (значення або яскравість) - описує світлість кольору. Перехід від моделі RGB до HSV здійснюється за певними математичними формулами, які дозволяють отримати значення H, S та V на основі початкових значень червоного (R), зеленого (G) та синього (B) каналів зображення.

На рисунку 1 можна розглянути формули.

$$H \in [0, 360), S, V, R, G, B \in [0, 1]$$

Let MAX be the maximum value of R, G, and B, and MIN be the minimum among them.

$$H = \begin{cases} 0, & \text{if MAX} = \text{MIN (i.e., grayscale)} \\ 60 \times \frac{G-B}{\text{MAX}-\text{MIN}} + 0, & \text{if MAX} = R \text{ and } G \geq B \text{ (red region)} \\ 60 \times \frac{G-B}{\text{MAX}-\text{MIN}} + 360, & \text{if MAX} = R \text{ and } G < B \text{ (wrapping case)} \\ 60 \times \frac{B-R}{\text{MAX}-\text{MIN}} + 120, & \text{if MAX} = G \text{ (green region)} \\ 60 \times \frac{R-G}{\text{MAX}-\text{MIN}} + 240, & \text{if MAX} = B \text{ (blue region)} \end{cases}$$

$$S = \begin{cases} 0, & \text{if MAX} = 0 \text{ (no saturation - black);} \\ 1 - \frac{\text{MIN}}{\text{MAX}}, & \text{otherwise (normalized saturation)} \end{cases}$$

$$V = \text{MAX}$$

Рисунок 1 - перетворення колірного простору з RGB на HSV

3.2. Блок схема алгоритму завдання та її опис.

На рисунку 2 можна розглянути схему роботи програмного скрипта.

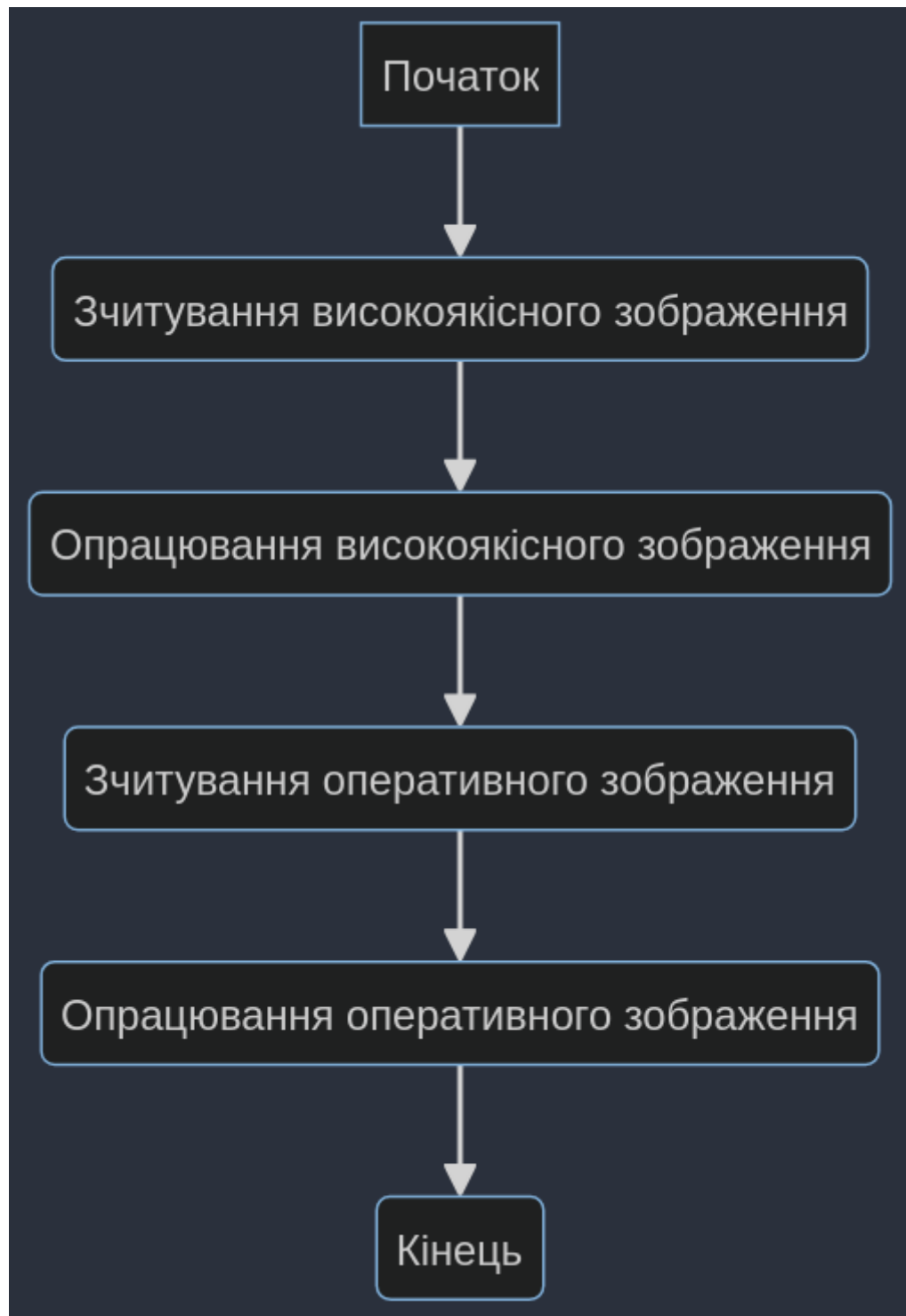


Рисунок 2 - Схема алгоритму

3.3. Опис структури проекту програми в середовищі PyCharm.

Для реалізації розробленого алгоритму мовою програмування Python з використанням можливостей інтегрованого середовища PyCharm сформовано проект.

Проект базується на лінійній бізнес-логіці функціонального програмування та має таку структуру.

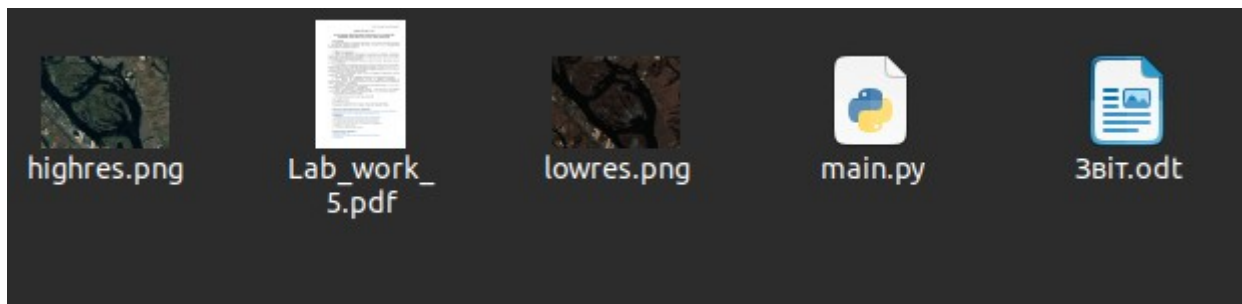


Рисунок 3 - Структура проекту

main.py — скрипт

highres.png — високоякісне зображення

lowres.png — оперативне зображення

Lab_work_5.pdf — умова завдання

Звіт.odt — звіт

3.4. Результати роботи програми відповідно до завдання.

Результатом роботи скрипта є зображення, що показують покриття території населеного пункту Вишеньок 2 водоймами.

Вхідне високоточне зображення:

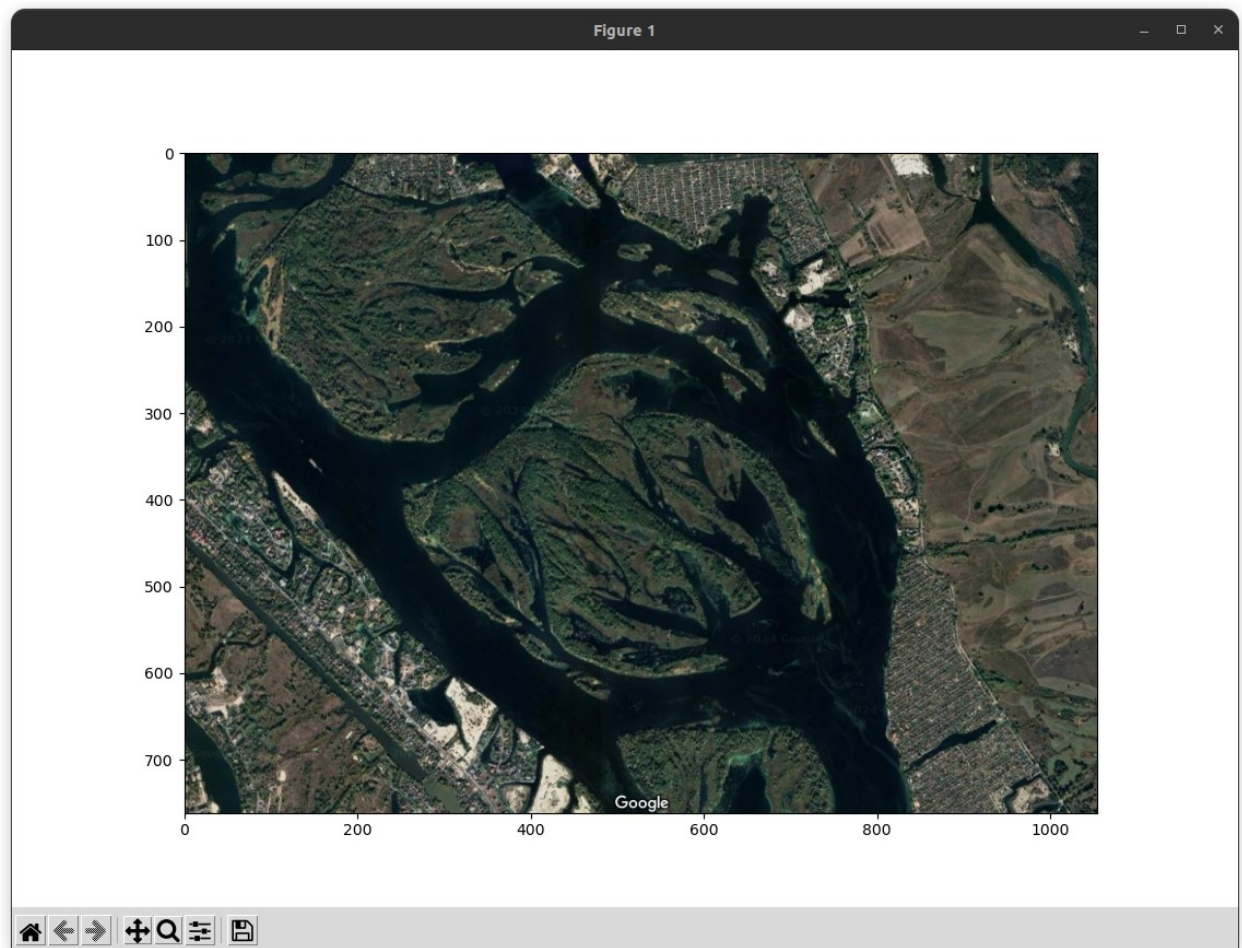


Рисунок 4 - Високоякісне зображення

Сформована на основі високоточного зображення маски:

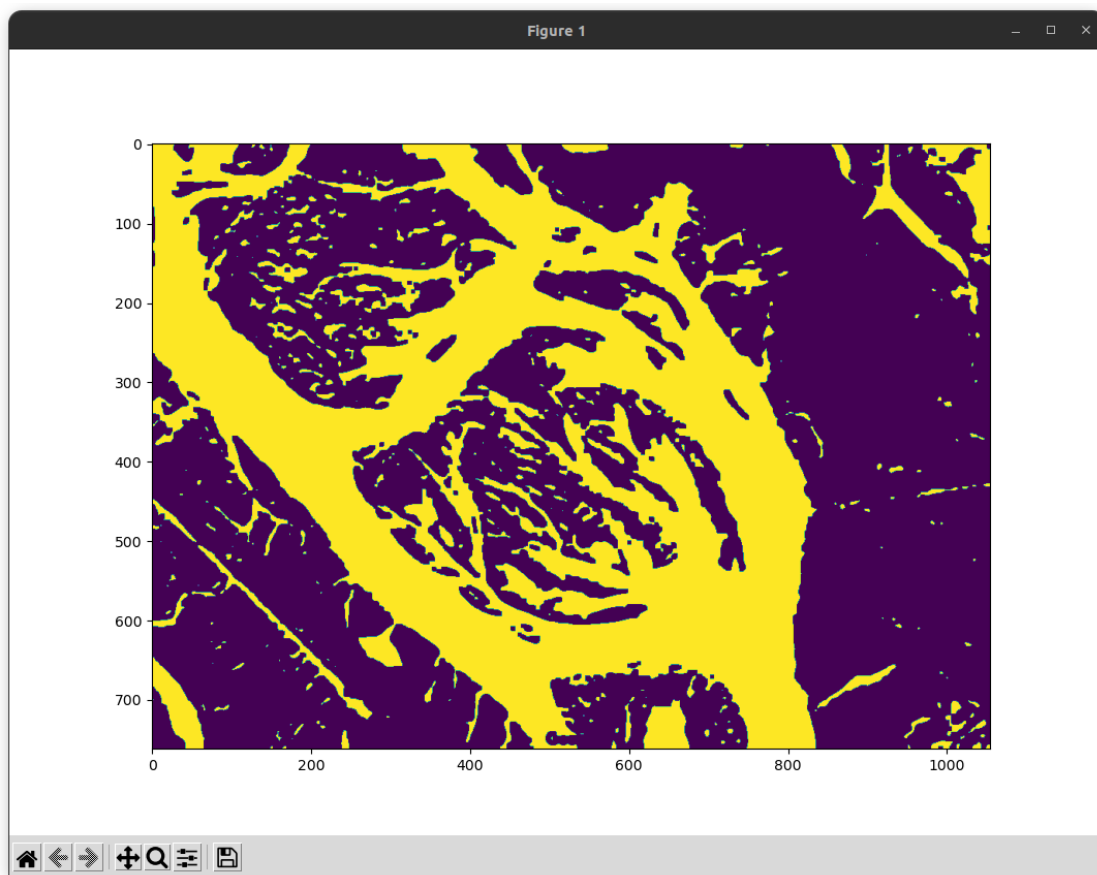


Рисунок 5 - Маска високоякісного зображення

Накладення маски на територію:



Рисунок 6 - Накладена маска на високоякісне зображення

Покриття території водоймами:

```
/usr/bin/python3.10 /home/sideshowbobgot/university/ComputerVisionSixSemester/Lab5/main.py  
Водойми займають = 38.0% кадру
```

Рисунок 7 - Результат покриття

Вхідне тестове оперативне зображення:

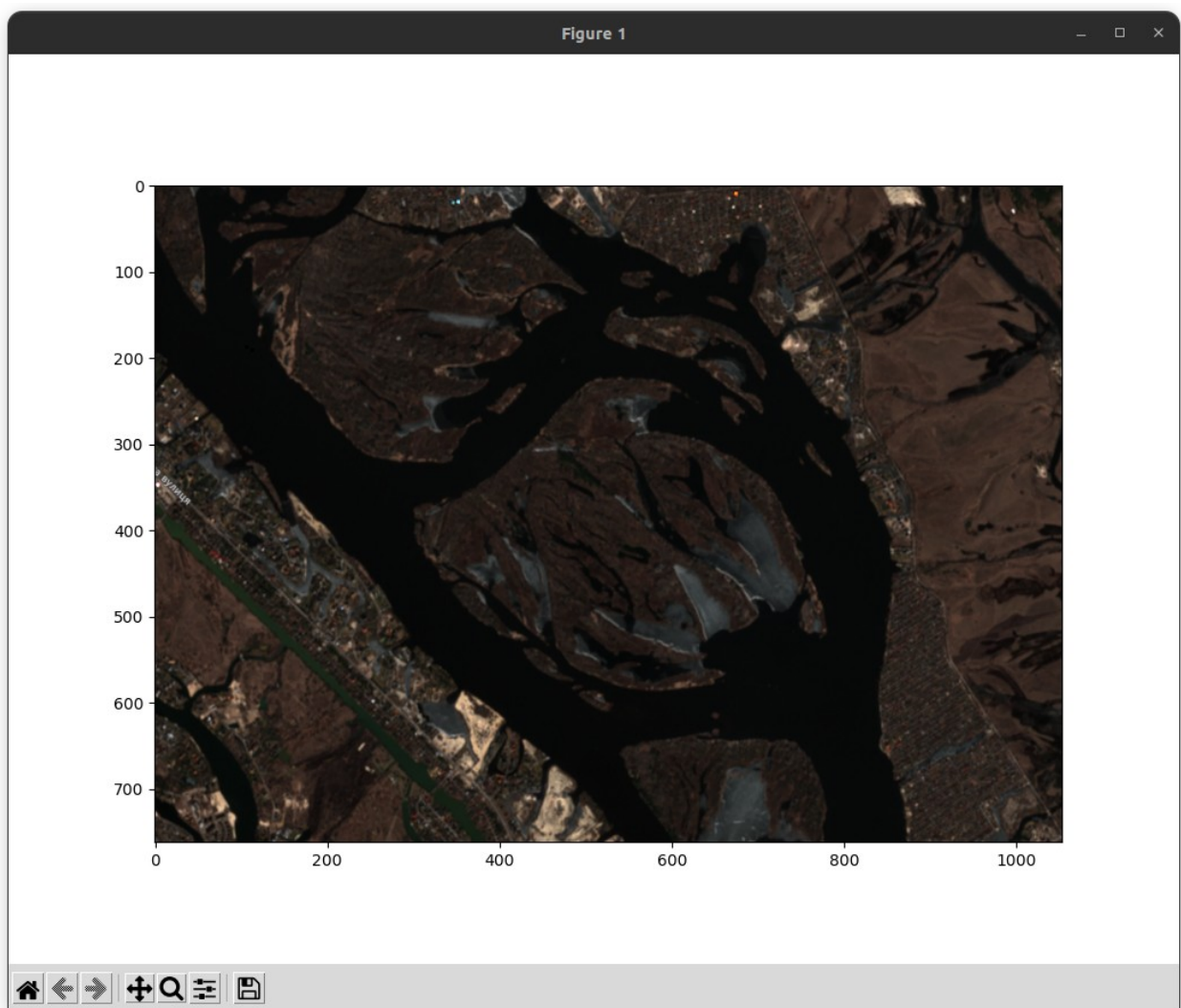


Рисунок 8 - Оперативне зображення

Сформована на основі оперативного зображення маска:

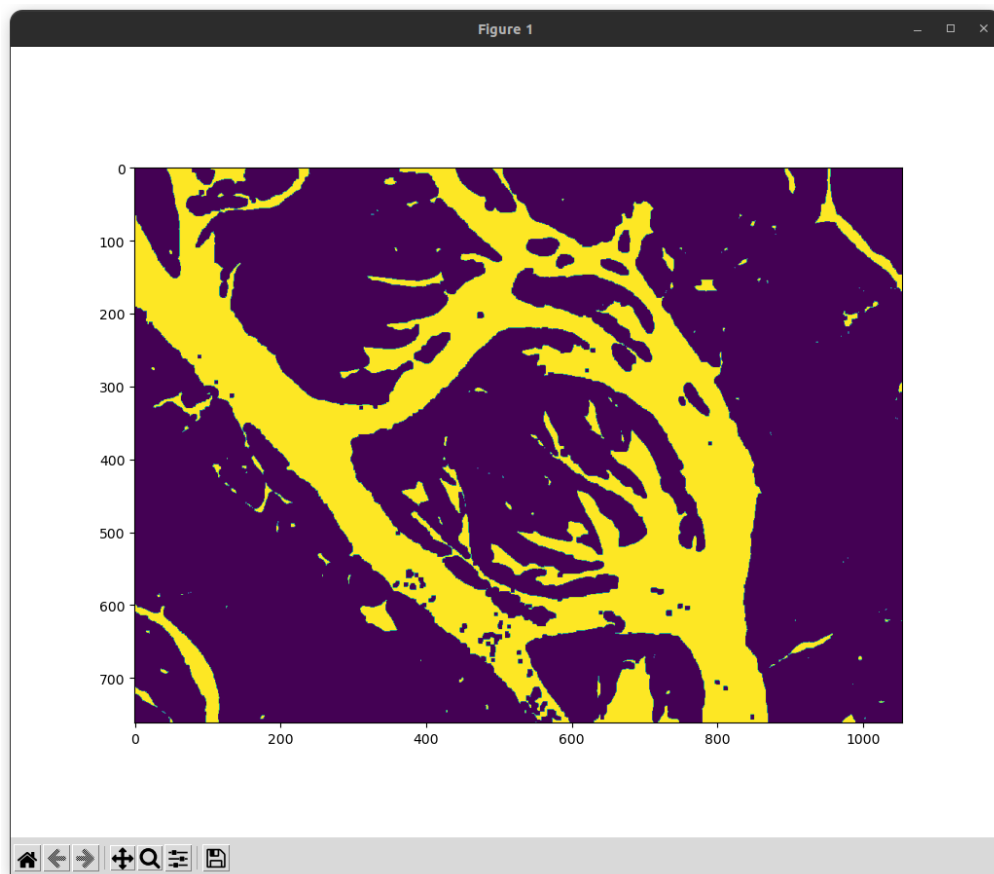


Рисунок 9 - Маска оперативного изображения

Наочний результат роботи алгоритму:

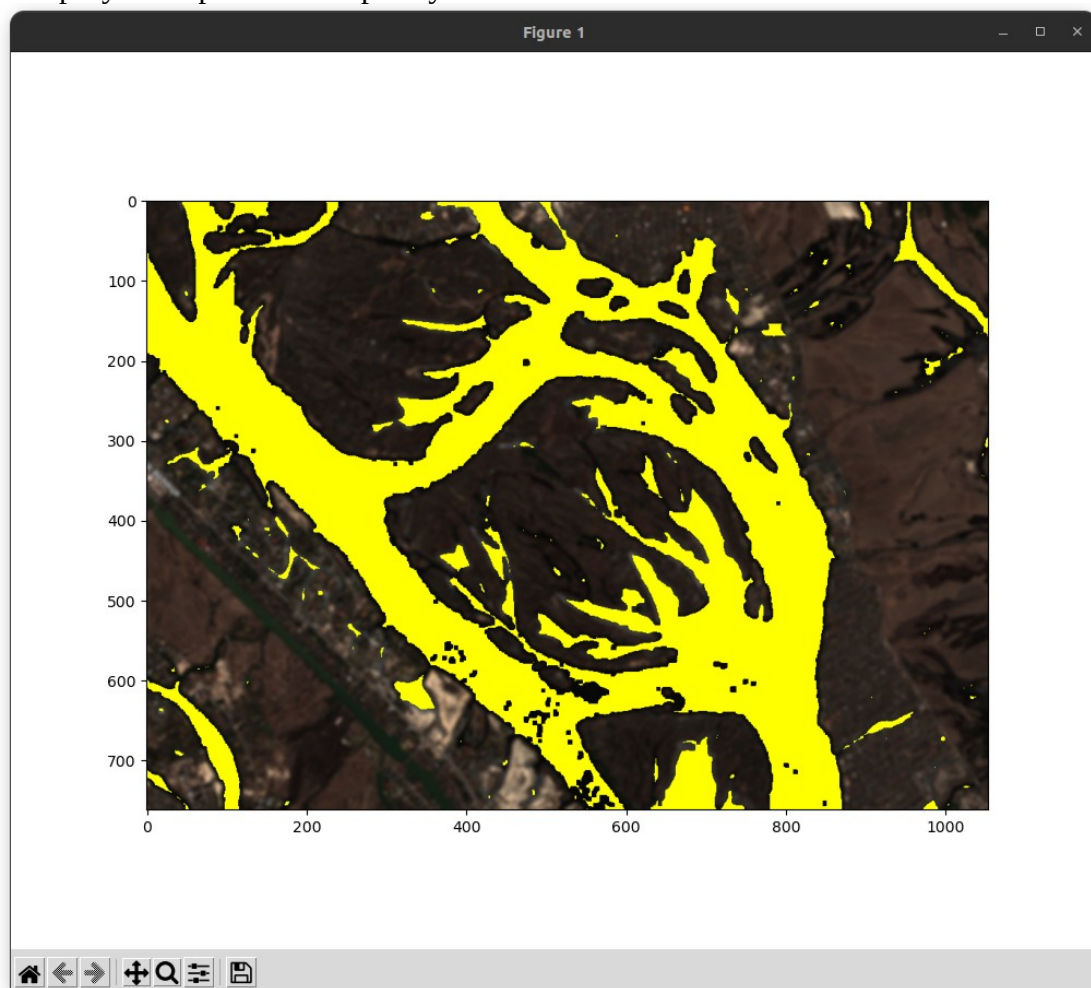


Рисунок 10 - Накладена маска на оперативне зображення

Водойми займають = 27.0% кадру

Як можна побачити, ідентифікація водойм проходить успішно. Хоча бачимо, що на високоякісному зображенні малі притоки річки детектуються краще, тому існує різниця у покритті водоймами території на двох рисунках. Представлені результати у повному обсязі відповідають завданню лабораторної роботи.

3.5. Програмний код.

Програма послідовно виконує алгоритм, представлений на рисунку 1, і призначена для отримання результатів, які ілюструються на рисунках з 4 по 10. Для спрощення коду та оптимізації обчислювального процесу було впроваджено використання функцій та підпрограм. При цьому були застосовані інструменти Python та бібліотеки такі як Matplotlib та OpenCV. Контекстні зауваження в програмі забезпечують пояснення функцій та методів, використаних у коді.

main.py

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

# зчитування та відображення зображення
def image_read(FileIm):
    img = cv2.imread(FileIm)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    plt.imshow(img)
    plt.show()
    return img

# опрацювання зображення
def image_processing(img, is_lowres=False):
    # розмиття гауса з більшим ядром
    img = cv2.GaussianBlur(img, (7, 7), 3)

    # перехід до HSV
    hsv_img = cv2.cvtColor(img, cv2.COLOR_RGB2HSV)

    # розширений діапазон кольорів для водойм
    if is_lowres:
        # для low-res зображення включаємо більш темно-сині відтінки
        lower_range = (70, 0, 0)
        upper_range = (180, 255, 100)
    else:
        # для high-res зображення залишаємо все як є
        lower_range = (0, 0, 0)
        upper_range = (180, 255, 50)

    # маска частин зображення що задовольняють потреби
    mask = cv2.inRange(hsv_img, lower_range, upper_range)

    # морфологічне закриття для видалення шумів
    kernel = np.ones((5, 5), np.uint8)
```

```

mask = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kernel)

plt.imshow(mask)
plt.show()

# змінюємо колір водойм на блакитний для кращої видимості
img[mask > 0] = (255, 255, 0)

# підрахунок водойм у кадрі
res = img.shape
water = np.sum(mask == 255)
print(f"Водойми займають = {(water / (res[0] * res[1]) * 100):.1f}% кадру")

# відображення
plt.imshow(img)
plt.show()
return

# головна частина скрипту
if __name__ == '__main__':
    # зчитування високоточного зображення
    image_entrance = image_read("highres.png")

    # його опрацювання
    image_processing(image_entrance)

    # зчитування оперативного зображення
    image_entrance = image_read("lowres.png")

    # його опрацювання з урахуванням низької роздільної здатності
    image_processing(image_entrance, is_lowres=True)

```

3.6. Аналіз результатів відлагодження та верифікації результатів роботи програми.

Результати від лагодження та тестування довели працездатність розробленого коду.

Верифікація функціоналу програмного коду, порівняння отриманих результатів з технічними умовами завдання на лабораторну роботу доводять, що усі завдання виконані у повному обсязі.

IV. Висновки.

Під час виконання лабораторної роботи було здійснено аналіз технологій сегментації та кластеризації для обробки цифрових зображень у контексті Computer Vision, використовуючи високорівневу програмну мову Python та її спеціалізовані бібліотеки. Робота включала в себе визначення площинних об'єктів, таких як водойми, на знімках з точних та оперативних джерел. Для мінімізації шумів на високодеталізованих зображеннях застосовувалося розмиття за Гаусом, а для виявлення синіх та темно-синіх кольорів використовувалась переведення зображення у HSV кольорову модель. Ідентифікація водойм здійснювалась через формування маски, яка дозволяла визначити відсоток площі водойм в обраній місцевості та візуалізувати їх на зображенні.

Виконав: студент Панченко С.В.