

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій

«До захисту допущено»

В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО

_____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня бакалавр

зі спеціальності 122 «Комп'ютерні науки» _____,

освітньо-професійної програми «Інформаційні технології проектування»
на тему: «Інформаційна технологія машинного навчання геоінформаційної системи для семантичної сегментації цифрового зображення регіону»

Здобувача групи

ІТ-02

(шифр групи)

Рощупкін Олексій Адріанович

(прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

(підпис)

Олексій РОЩУПКІН

(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник доцент кафедри ІТ, к.т.н., доц. Юлія ПАРФЕНЕНКО

(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ)

(підпис)

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

В. о. зав. кафедри ІТ

_____ Світлана ВАЩЕНКО

«__» _____ 2024 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Рощупкіну Олексію Адріановичу

1 Тема роботи Інформаційні технології проектування»

на тему: «Інформаційна технологія машинного навчання геоінформаційної системи для семантичної сегментації цифрового зображення регіону

керівник роботи Парфененко Юлія Вікторівна, доцент _____,

затверджені наказом по університету від « 07 » травня 2024 р. №0482-VI

2 Строк подання студентом роботи « 26 » травня 2024 р.

3 Вхідні дані до роботи _____

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) _____

1) Огляд останніх досліджень і публікацій; 2) Аналіз методів семантичної сегментації та аналітичний огляд методів машинного навчання; 3) Розробка та розгляд алгоритму геоінформаційної системи семантичної сегментації 4) Програмна реалізація геоінформаційної системи семантичної сегментації; 5) Аналіз результатів

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди: 1) Титульний аркуш; 2) Актуальність; 3) Мета роботи, об'єкт та предмет дослідження; 4) Функціональна категорійна модель; 5) Вхідні дані; 6) Опис алгоритму машинного навчання; 7) Результат комп'ютерного моделювання; 8) Висновки

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Огляд останніх досліджень і публікацій	07.05-09.05.24	виконано
2	Аналіз методів семантичної сегментації та аналітичний огляд методів машинного навчання	10.05-13.05.24	виконано
3	Розробка та розгляд алгоритму геоінформаційної системи семантичної сегментації	14.05-18.05.24	виконано
4	Програмна реалізація геоінформаційної системи семантичної сегментації	19.05-21.05.24	виконано
5	Аналіз отриманих результатів	22.05-24.05.24	виконано
6	Оформлення пояснювальної записки до кваліфікаційної роботи бакалавра	24.05-26.05.24	виконано

Студент _____
(підпис)

Олексій РОЦУПКІН

Керівник роботи _____
(підпис)

доц., Юлія ПАРФЕНЕНКО

АНОТАЦІЯ

Темою кваліфікаційної роботи бакалавра є «Інформаційна технологія машинного навчання геоінформаційної системи для семантичної сегментації цифрового зображення регіону».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 26 найменувань, 3 додатків. Загальний обсяг роботи – 88 сторінок, у тому числі 45 сторінок основного тексту, сторінки списку використаних джерел, 42 сторінки додатків.

Актуальність теми дослідження визначається потребою у розвитку систем для сегментації цифрових зображень за допомогою машинного навчання. Швидкий розвиток цифрових технологій призвів до зростання обсягу цифрових зображень, що потребує ефективних методів їх аналізу та обробки.

Об'єктом дослідження є процес машинного навчання системи семантичної сегментації зображень.

Предмет дослідження є інформаційна технологія семантичної сегментації зображень з використанням методу інформаційно-екстремального машинного навчання.

Метою роботи є розроблення системи семантичної сегментації зображень цифрового кадру за допомогою інформаційно-екстремальному алгоритму машинного навчання.

У результаті дипломного проектування проведено аналіз існуючих методів семантичної сегментації. Розроблено алгоритм та програмне забезпечення геоінформаційної системи семантичної сегментації, застосовуючи інформаційно – екстремальне машинне навчання. Проведено навчання системи на реальних вхідних даних. Інформаційну систему було розроблено за допомогою мови програмування Python.

Ключові слова: інформаційно-екстремальна інтелектуальна технологія, геоінформаційна система, машинне навчання, семантична сегментація.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРОБЛЕМИ	8
1.1 Огляд останніх досліджень і публікацій	8
1.2. Аналіз програмних продуктів - аналогів	11
1.3. Постановка задачі	14
2. ОПИС МЕТОДУ ДОСЛІДЖЕННЯ	16
2.1. Вхідні дані	16
2.2. Базовий алгоритм інформаційно-екстремального машинного навчання	17
2.3 Функціональні категорійні моделі інформаційно-екстремального машинного навчання з оптимізацією контрольних допусків	21
2.4 Опис алгоритму інформаційно-екстремального машинного навчання з оптимізацією контрольних допусків на ознаки розпізнавання	26
3. МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО ДОДАТКУ СЕГМЕНТАЦІЇ ЗОБРАЖЕНЬ	32
3.1. Структурно-функціональне моделювання	32
4 ПРОГРАМНА РЕАЛІЗАЦІЯ	36
4.1. Програмна реалізація і опис програмного забезпечення	36
ВИСНОВКИ	45
СПИСОК ЛІТЕРАТУРИ	46
ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ	50
ДОДАТОК Б.	64
ДОДАТОК В. ЛІСТИНГ ПРОГРАМИ	76

ВСТУП

У роботі розглядається задача застосування методів машинного навчання для геоінформаційної системи, яка здійснює ідентифікацію кадрів цифрового зображення місцевості з метою сегментації та класифікації його ділянок. Досягнення мети здійснюється шляхом використання ідей і методів інформаційно-екстремальної інтелектуальної технології (ІЕІ-технології) аналізу даних.

В сучасному цифровому світі обробка зображень є однією з найбільш актуальних та перспективних галузей комп'ютерних наук. Системи комп'ютерного зору здійснюють розпізнавання об'єктів, що оточують нас, та аналізують цифрові зображення в різних сферах, від медицини до автомобільної промисловості. Одним із важливих етапів обробки зображень є їх сегментація – розділення на окремі частини для подальшого аналізу. На точність визначення територій інтересу в електронному зображенні регіону впливають такі ключові фактори:

- Актуальність вхідного математичного опису в геоінформаційній системі для семантичної сегментації кадрів зображень місцевості.
- Функціональна ефективність алгоритмів машинного навчання семантичної сегментації геоінформаційної системи для аналізу наземних об'єктів.

Об'єктом дослідження є процес машинного навчання системи семантичної сегментації зображень.

Предмет дослідження є інформаційна технологія семантичної сегментації зображень з використанням методу інформаційно-екстремального машинного навчання.

Метою роботи є розроблення системи семантичної сегментації зображень цифрового кадру за допомогою інформаційно-екстремальному алгоритму машинного навчання.

Задля успішного виконання мети проекту можна виділити наступні задачі:

1. Визначити актуальність роботи, дослідити її предметну область.

2. Провести аналіз останніх публікацій суміжних тем.
3. Визначити функціональні можливості програмного додатку семантичної сегментації зображень.
4. Визначити засоби реалізації для побудови програмного додатку.
5. Програмно реалізувати алгоритм машинного навчання для заданої системи.
6. Програмно реалізувати логіку семантичної сегментації зображень.
7. Дослідити отримані результати тестування.

Практичне значення роботи полягає у її застосуванні для аналізу цифрових зображень місцевості. Зараз, завдяки використанню концентратора Raspberry Pi 3 як основи, сучасна бортова система має значну потужність обчислень, а також високоякісні відеокамери та тепловізори з великою роздільною здатністю. Це створює всі необхідні технічні умови для швидкого аналізу та обробки цифрових зображень. Практичним результатом кваліфікаційної роботи бакалавра є підвищенню функціональної ефективності машинного навчання автономного БПЛА, у тому числі за рахунок впровадження сегментації зображень.

1 АНАЛІЗ ПРОБЛЕМИ

1.1 Огляд останніх досліджень і публікацій

Останні дослідження в області сегментації цифрових зображень з використанням методів машинного навчання свідчать про значний прогрес у цьому напрямку. Сегментація зображень, яка полягає в розділенні зображення на окремі сегменти або області для подальшого аналізу та обробки, є важливим етапом в обробці зображень.

Широко застосовуються згорткові нейронні мережі (CNN), які демонструють високу точність у різних задачах сегментації. Вони є потужним інструментом для автоматичного виявлення об'єктів на зображеннях та використання в різноманітних застосуваннях, таких як медична діагностика, аналіз зображень з дронів, автономні автомобілі, виробництво та багато інших.

Недавні дослідження показали, що використання методів машинного навчання у цьому процесі дозволяє досягти високої точності та швидкості обробки, що є актуальним у контексті розвитку сучасних систем обробки зображень. Деякі з найбільш успішних підходів включають у себе використання таких архітектур, як U-Net, SegNet, Mask R-CNN, PSPNet, ResNeXt та інших.

Останні дослідження в області сегментації цифрових зображень за допомогою методів машинного навчання свідчать про значний прогрес у цьому напрямку. Зокрема, в статті [1] автори застосували архітектуру U-Net для сегментації медичних зображень, де показали високу точність та ефективність порівняно з іншими методами. Ще однією важливою галуззю є сегментація зображень в умовах обмеженої кількості даних. У роботі [2] дослідники використали техніку передбачення на основі знань (knowledge distillation) для покращення сегментації в умовах недостатньої кількості даних.

Також важливим напрямком є розвиток методів для роботи з великими обсягами даних та високою швидкістю обробки. У роботі [3] було запропоновано новий підхід до сегментації зображень, який поєднує в собі ефективність

згорткових нейронних мереж з високою швидкістю роботи. Крім того, у роботах [4-7] використано глибокі мережі для сегментації медичних зображень у тривимірному просторі, показавши високу точність у сегментації структур головного мозку на основі зображень магнітно-резонансної томографії.

В дослідженні [8] пропонується архітектура DeepLab для семантичної сегментації зображень. Модель використовує глибокі згорткові нейронні мережі з атрієм (atrous convolution), що дозволяє збільшити рецептивне поле без втрати роздільної здатності. Також використовується повністю зв'язані узгоджені випадкові поля (fully connected conditional random fields), що допомагає виправляти недоліки сегментації, пов'язані з відсутністю контексту. Модель досягає вражаючих результатів на таких відомих наборах даних, як PASCAL VOC 2012 та Cityscapes.

Врох U-Net [9] є ефективною архітектурою для семантичної сегментації зображень, особливо в біомедичних дослідженнях. Модель складається з енкодера, який зменшує роздільну здатність зображення, і декодера, який відновлює сегментоване зображення до вихідного розміру. Така архітектура дозволяє моделі зберігати інформацію про контекст та використовувати її для точної сегментації об'єктів.

В дослідженні [10] пропонується глибока мережа парсингу для семантичної сегментації зображень. Мережа складається з енкодера, який виділяє важливі ознаки зображення, та декодера, який використовує ці ознаки для генерації семантичної сегментації. Застосування механізму згорткового декодування (convolutional decoder) дозволяє моделі ефективно відтворювати об'єкти на зображенні, що призводить до високої точності на різноманітних наборах даних.

В дослідженні [11] запропоновано архітектуру Encoder-Decoder для семантичної сегментації зображень. Модель використовує атріємні роздільні згортки для збільшення рецептивного поля без втрати роздільної здатності. Це дозволяє моделі отримувати більше контекстуальної інформації і досягати високої точності на відомих наборах даних.

В роботі [12] пропонується новий підхід до семантичної сегментації зображень, використовуючи глибокі нейронні мережі з короткими з'єднаннями. Модель використовує внутрішню наглядність для забезпечення більш ефективного навчання та вищої точності сегментації.

Дослідження в області сегментації цифрових зображень також включає роботу над розробкою ефективних алгоритмів для обробки великих обсягів даних та високою швидкістю обробки. У роботах [13-17] досліджено різноманітні архітектури нейронних мереж та методи для задач сегментації зображень, що показують високу ефективність та точність результатів.

Зображення відіграють важливу роль у геоінформаційних системах (ГІС), які використовуються для збору, аналізу та візуалізації геопросторової інформації. Обробка зображень для ГІС дозволяє автоматизувати аналіз геоданих, виявляти об'єкти на зображеннях, виконувати класифікацію земельних використань та визначати геопросторові характеристики об'єктів і територій. Методи машинного навчання, зокрема нейронні мережі, демонструють вражаючі результати у вирішенні завдань обробки зображень для ГІС.

Наприклад, в роботі [18] було розглянуто застосування CNN для класифікації зображень в геоінформаційних системах з високою точністю та швидкістю.

Також, у роботах [19, 20] розглядається вітчизняна технологія розробки ГІС системи розпізнавання транспортних засобів на основі інформаційно-екстремальної інтелектуальної технології аналізу даних, яка ґрунтується на максимізації інформаційної спроможності системи в процесі її машинного навчання.

Методи машинного навчання в рамках ІЕІ-технології використовують радіально базисні методи для створення гіперповерхонь, які розділяють класи для розпізнавання. Основна ідея інтелектуального аналізу даних у цьому контексті, схожа на штучні нейронні мережі полягає в тому, щоб під час машинного навчання адаптувати математичний опис даних для максимально точних рішень класифікації. Також перевагою цього методу є те, що в інформаційно-екстремальному машинному навчанні використовується підхід моделювання

когнітивних процесів, що дозволяє безпосередньо моделювати природний механізм прийняття рішень схожий на людський. Цей підхід надає гнучкість у перенавчанні системи через збільшення кількості класів розпізнавання. Побудова правил у рамках цього підходу вирішує проблему багатовимірності ознак розпізнавання, оскільки сучасні комп'ютери здатні обробляти структуровані вектори, що містять 2^{85} ознак розпізнавання.

Через це, в результаті проведеного дослідження було вибрано інформаційно-екстремальний інтелектуальний алгоритм [20] для подальшої реалізації семантичної сегментації цифрових зображень.

Обраний метод базується на сучасних підходах машинного навчання, що дозволяє досягти високої точності та ефективності у сегментації зображень.

Отримані результати [21] свідчать про успішність обраного підходу до сегментації зображень та його потенціал у практичному застосуванні. Розроблена система сегментації зображень може стати ефективним інструментом у різних галузях та сприяти подальшому розвитку сучасних технологій обробки цифрових зображень.

Загальна робота над проектом дозволила не лише розглянути та проаналізувати різноманітні методи сегментації зображень, але й реалізувати ефективну систему, яка може бути використана для подальших досліджень та практичних застосувань у різних галузях.

1.2. Аналіз програмних продуктів - аналогів

На сьогоднішній день існує значна кількість програмних продуктів, які забезпечують сегментацію цифрових зображень за допомогою методів машинного навчання. Деякі з найбільш відомих та популярних програмних продуктів включають у себе OpenCV, TensorFlow, PyTorch та Keras.

OpenCV [22] - це безкоштовна бібліотека з відкритим кодом для комп'ютерного зору та обробки зображень, розроблена Intel. Вона пропонує широкий спектр функцій, включаючи алгоритми сегментації зображень на основі

машинного навчання. OpenCV використовує такі методи для сегментації зображень, як K-Means, Watershed, GrabCut та Simple Linear Iterative Clustering (SLIC).

TensorFlow [23] - це безкоштовна бібліотека з відкритим кодом для машинного навчання, розроблена Google. Вона використовується для створення моделей машинного навчання, включаючи моделі сегментації зображень. TensorFlow використовує такі методи, як згорткові нейронні мережі (CNN) та архітектура U-Net.

PyTorch [24] - це безкоштовна бібліотека з відкритим кодом для машинного навчання, яка надає динамічний граф обчислень, що дозволяє легко створювати, навчати та використовувати моделі. Вона використовується для створення моделей машинного навчання, включаючи моделі сегментації зображень. PyTorch також використовує методи, подібні до TensorFlow, такі як згорткові нейронні мережі (CNN) та архітектура U-Net.

Keras [25] - це бібліотека з відкритим кодом з високорівневим інтерфейсом для створення моделей машинного навчання. Вона працює поверх інших бібліотек, таких як TensorFlow або PyTorch, і пропонує більш простий та зрозумілий синтаксис для побудови моделей. Keras дозволяє легко створювати моделі сегментації зображень на основі CNN або U-Net за допомогою TensorFlow або PyTorch. Крім того, він пропонує модулі для завантаження, підготовки та збільшення даних зображень, а також для навчання та оцінки моделей сегментації.

PIL [26] (Python Imaging Library) - це бібліотека для роботи з зображеннями у мові програмування Python. PIL дозволяє відкривати, редагувати та зберігати різні формати зображень. Ця бібліотека часто використовується для попередньої обробки зображень перед їх подальшою обробкою алгоритмами сегментації.

Ці програмні продукти надають широкі можливості для розробки систем сегментації зображень на основі методів машинного навчання. Вони мають активні спільноти користувачів, які постійно розробляють нові методи та алгоритми для поліпшення точності та ефективності сегментації зображень.

Сформуємо порівняльну таблицю можливостей програмних продуктів таб. 1.

Таблиця 1 - Порівняльна таблиця можливостей програмних продуктів

Критерії	OpenCV	TensorFlow	PyTorch	PIL	Keras
Методи сегментації	K-Means, Watershed, GrabCut, SLIC	CNN, U-Net, DeepLab, SegNet, Mask R-CNN	CNN, U-Net, DeepLab, SegNet, Mask R-CNN	Немає	CNN, U-Net, DeepLab, SegNet, Mask R-CNN
Відкриття зображень	Так	Немає	Немає	Так	Немає
Редагування зображень	Так	Немає	Немає	Так	Немає
Збереження зображень	Так	Немає	Немає	Так	Немає
Створення моделей	Немає	Так	Так	Немає	Так
Застосування	Комп'ютерний зір, обробка зображень	Машинне навчання, комп'ютерний зір	Машинне навчання, комп'ютерний зір	Обробка зображень	Машинне навчання, комп'ютерний зір

Оскільки реалізація семантичної сегментації ґрунтуватиметься на інформаційно-екстремальному алгоритмі машинного навчання, який не має вбудованої підтримки в жодній з доступних бібліотек і буде програмно реалізований самостійно, для нашої реалізації ми оберемо лише ті бібліотеки, які необхідні лише для обробки зображень. Використання будь-яких інших бібліотек

може лише ускладнити та уповільнити процес розробки. Таким чином, для підготовки зображень до машинного навчання було обрано бібліотеки PIL та OpenCV. Сам алгоритм буде реалізований за допомогою стандартних методів мови програмування Python.

1.3. Постановка задачі

Метою дослідження є розробка інформаційної технології сегментації цифрових зображень на основі інформаційно-екстремального машинного навчання для геоінформаційної системи. Система має працювати в поєднанні з системою аерозйомки для розпізнавання і семантичної сегментації об'єктів на землі. Сегментація зображень є важливим етапом в обробці зображень, оскільки дозволяє автоматично розподілити зображення на окремі сегменти або області для подальшого аналізу та обробки.

Задача інформаційно-екстремального синтезу геоінформаційної системи полягає в оптимізації параметрів машинного навчання шляхом наближення до максимального значення інформаційного критерію. Водночас метою інформаційно-екстремального машинного навчання є побудова безпомилкових за навчальною матрицею вирішувальних правил, які в режимі екзамену забезпечують високу достовірність рішень і дозволяють провести семантичну сегментацію зображення за допомогою ідентифікації окремих кадрів зображення.

Основними етапами дослідження є:

1. Розробка базового алгоритму машинного навчання за допомогою інформаційно-екстремального алгоритму.
2. Реалізація алгоритму оптимізації параметрів машинного навчання що покращує результати сегментації зображення в режимі екзамену.
3. Наближення до безпомилковості за навчальною вибіркою вирішальних правил шляхом подальшого збільшення глибини машинного навчання.

4. Проведення режиму екзамену для проведення семантичної сегментації кадрів зображення на основі отриманих вирішальних правил.

Використання методів машинного навчання у цьому процесі дозволяє досягти високої точності та швидкості обробки. Порівняно з традиційними методами обробки зображень, методи машинного навчання забезпечують кращі результати в ряді завдань сегментації, зокрема в умовах складних або неоднорідних зображень. Такий підхід дозволяє автоматизувати та полегшити процес обробки зображень, що робить його більш ефективним та швидким.

Розроблена система має на меті вирішити ряд завдань:

- Підвищення точності та швидкості сегментації зображень за допомогою застосування методів машинного навчання.
- Забезпечення зручного та інтуїтивно зрозумілого інтерфейсу користувача для взаємодії з системою.
- Підтримка різноманітних форматів зображень, таких як JPEG, PNG, TIFF тощо.
- Вихідні дані повинні бути у вигляді маски класифікованих кадрів які в поєднанні створюють сегментаційну карту, де кожен кадр відповідає певному класу об'єкта.
- Висока точність сегментації, вимагається досягнення високого показника інформаційного критерію [26] для різних класів об'єктів на тестових наборах даних.
- Швидка обробка зображень з використанням оптимізованих алгоритмів.
- Можливість обробки зображень в реальному часі з врахуванням вимог до швидкості обробки.

Ця робота є актуальною та важливою у контексті розвитку сучасних систем обробки зображень та їх застосування у різних галузях, включаючи медицину, автомобільну промисловість, аналіз земельних даних та багато інших. Отримані результати в подальшому будуть впроваджені в загальну систему БСР для автономного БПЛА.

2. ОПИС МЕТОДУ ДОСЛІДЖЕННЯ

2.1. Вхідні дані

Для створення математичного опису інформаційно-екстремального навчання інформаційної технології для геоінформаційної системи було використано дані, отримані шляхом порядкового зчитування яскравості пікселів цифрового зображення регіону (рис. 2.1).

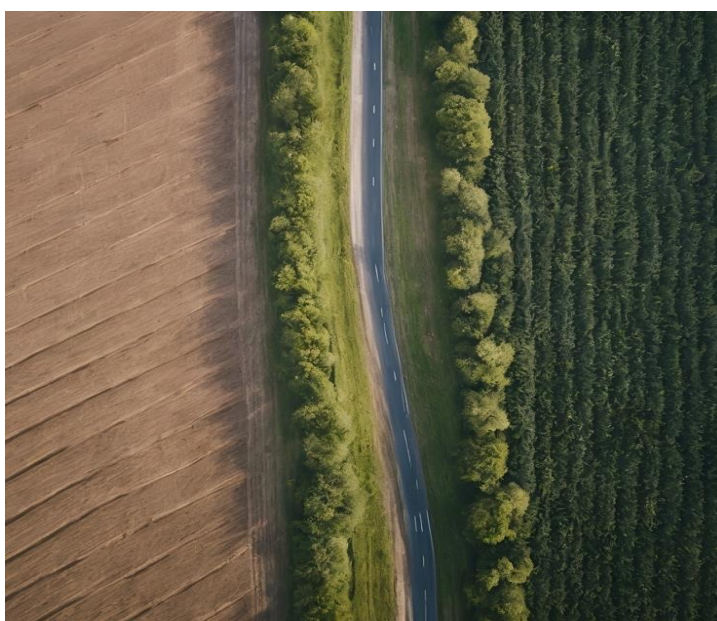


Рисунок 2.1 – Зображення регіону

Для формування алфавіту класів розпізнавання було взято чотири кадри, які характеризували різні ділянки місцевості розмірністю 50x50 пікселей.

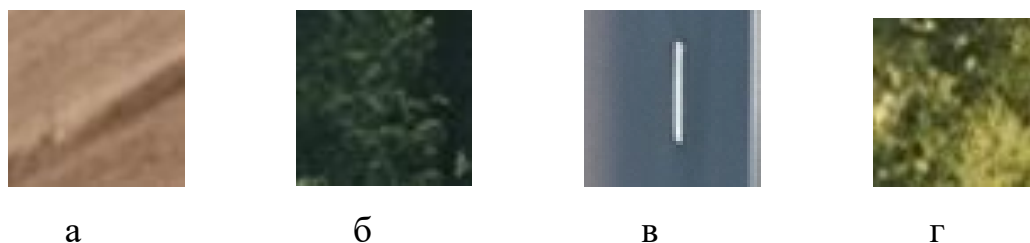


Рисунок 2.2 – Зображення кадрів: а – клас розпізнавання X^0_1 (поле); б – клас розпізнавання X^0_2 (ліс); в – клас розпізнавання X^0_3 (дорога); г – клас розпізнавання X^0_4 (сад)

На основі обраних кадрів цифрового зображення розмірністю 50x50 пікселів була сформована вхідна навчальна матриця яскравості шляхом послідовного зчитування RGB-компонент кадрів. Водночас значення яскравості кожного пікселя у форматі RGB є ознакою розпізнавання. Оскільки показані на рисунку 2.2 зображення кадрів відносяться до типу “текстура”, то оброблення кадрів здійснювалося шляхом їх порядкового сканування в декартовій системі координат. Таким чином, навчальна матриця складалася із структурованих векторів ознак розпізнавання одного рядка зображення кадра. Водночас стовпчик навчальної матриці утворює навчальну вибірку випадкових значень відповідної ознаки розпізнавання. У результаті для кожного класу розпізнавання було сформовано матрицю розмірністю 150x50, де 150 кількість ознак розпізнавання і 50 кількість структурованих векторів ознак розпізнавання.

2.2. Базовий алгоритм інформаційно-екстремального машинного навчання

Для реалізації семантичної сегментації кадрів зображення шляхом машинного навчання в рамках ІЕІ технології необхідно реалізувати базовий алгоритм машинного навчання який виконує оптимізацію кодових відстаней задля віддалення простору значень бінарної навчальної матриці для кожного класу один від одного.

Фундаментальна концепція машинного навчання за допомогою технології інформаційно-екстремального алгоритму схожа на концепцію штучних нейронних мереж, мета якої полягає в тому, щоб адаптувати математичне представлення системи розпізнавання для максимізації ймовірності правильних рішень під час класифікації. Основна відмінність між інформаційно-екстремальними методами машинного навчання та штучними нейронними структурами полягає в їх розвитку в рамках підходу до моделювання процесів мислення, властивих людині під час формування та прийняття рішень з класифікації, ефективно моделюючи механізм

природного інтелекту. Водночас процеси машинного навчання розглядаються як оптимізація параметрів системи розпізнавання, що впливає на її ефективність. Ці параметри називаються параметрами машинного навчання. У рамках методів технології ІЕІ будь-яка статистична інформація про різноманітність аналізованих об'єктів може бути використана як критерій оптимізації.

Базовий алгоритм оптимізації параметрів функціонування інформаційної системи виконується в межах внутрішнього циклу процедури машинного навчання (3), що і визначає його назву. Основне призначення алгоритму навчання полягає в:

- оптимізації параметрів контейнерів класів розпізнавання;
- обчисленні інформаційного критерію;
- пошуку максимального значення критерію в області визначення його функції, яка є допустимою для цього критерію.

Модель інформаційно-екстремального машинного навчання для інтелектуальної системи у функціональній категорійній формі розробляється через створення спрямованого графа, де відповідні множини операторів взаємодіють між собою у процесі машинного навчання. Водночас математичний опис вхідних даних системи, здатної навчатись, представляється у вигляді структури [25]:

$$I_{ex} = \langle G, T, \Omega, Z, Y, X; f_1, f_2 \rangle ,$$

де G – простір вхідних аспектів; T – множина моментів в часі отримання інформації; Ω – простір ознак розпізнавання; Z – простір можливих станів об'єкту дослідження; Y – вхідна матриця навчання; X – робоча матриця навчання в бінарній системі, яка під час інформаційно-екстремального машинного навчання наближається до максимальної ймовірності прийняття правильних рішень при класифікації; f_1 – оператор формування вхідної матриці навчання із джерела інформації, яке задається декартовим добутком $G \times T \times \Omega \times Z$; f_2 – оператор перетворення вхідної матриці навчання Y в робочу бінарну матрицю навчання X .

Функціональну категорійну модель інформаційно-екстремального машинного навчання за базовим алгоритмом показано на рисунку 2.3 [26].

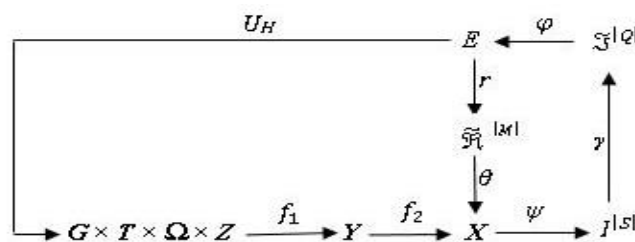


Рисунок 2.3 – Функціональна категорійна модель машинного навчання за базовим алгоритмом

На рисунку 2.3 множина E , яка складається із обчислених на кожному кроці машинного навчання значень критерію, є загальною для всіх контурів оптимізації параметрів. Оператор $r: E \rightarrow \tilde{\mathfrak{R}}^{|M|}$ під час процесу навчання системи, контейнери класів розпізнавання відтворюються у радіальному базисі бінарного простору ознак, що формують розбиття $\tilde{\mathfrak{R}}^{|M|}$. Оператор θ відображає розбиття $\tilde{\mathfrak{R}}^{|M|}$ на нечіткий розподіл класифікованих двійкових векторів ознак. Далі оператор $\psi: X \rightarrow I^{|S|}$, де $I^{|S|}$ – множина гіпотез, які перевіряють основну гіпотезу $\gamma_1: x_m^{(j)} \in X_m^o$. Оператор γ визначає множину характеристик $\mathfrak{Z}^{|Q|}$, де $Q = S^2$, а оператор ϕ обчислює набір значень критерію, який є функціоналом від характеристик точності. Контур цієї оптимізації закінчує роботу через терм-множину D , елементами якої є значення контрольних допусків на ознаки розпізнавання. Оператор u описує процес машинного навчання.

За одержаними в процесі машинного навчання параметрами класів розпізнавання будуються правила рішень, які в продукційній формі можемо подати у вигляді [26]:

$$(\forall X_k^o \in \tilde{\mathfrak{R}}^{|M|})(\forall X_l^o \in \tilde{\mathfrak{R}}^{|M|}) \left(\text{if } [(\mu_m > 0) \& (\mu_m = \max_{\{m\}} \{\mu_m\})] \right)$$

$$\text{then } x^{(j)} \in X_m^o \text{ else } x^{(j)} \notin X_m^o, \quad (13)$$

де $x^{(j)}$ – вектор, який розпізнається; μ_m – функція, яка описує належність вектора $x^{(j)}$ контейнеру класу розпізнавання X_m^o .

У виразі (14) функція належності для гіперсферичного контейнера класу розпізнавання X_m^o визначається за формулою [25]

$$\mu_m = 1 - \frac{d(x_m^* \oplus x^{(j)})}{d_m^*}, \quad (14)$$

де x_m^* – оптимальний двійковий вектор усередненого значення ознак; d_m^* – оптимальний радіус контейнера гіперсфери, для класу розпізнавання X_m^o .

Таким чином, при роботі системи в режимі екзамену (процесу сегментації) визначається за правилами рішень (13) належність реалізації класу розпізнавання до одного з класів, що і є процесом класифікації. Водночас правила рішень через малу обчислювальну трудомісткість відрізняються високою оперативністю.

Оцінка ефективності інформаційно-екстремального машинного навчання проводиться під час роботи інформаційної системи у режимі екзамену.

Модель, у вигляді орієнтованого графа, яка відображає множини, що використовуються на етапі екзамену, можна побачити на рисунку 2.4. [25].

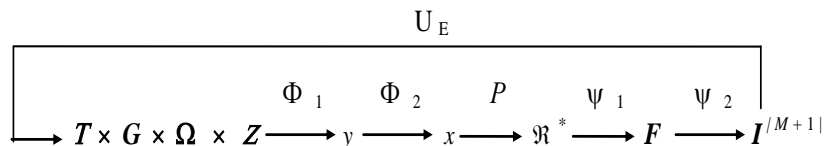


Рисунок 2.4 – Функціональна категорійна модель роботи системи розпізнавання в режимі екзамену

У категорійній моделі (рис. 2.4) оператор Φ_1 формує вектор екзамену у ознак класу, що розпізнається, тобто вхідних даних. Цей вектор схожий за структурою на

вектор з матриці навчання. Оператор Φ_2 за заданими на етапі навчання системи контрольними допусками формує вектор двійкових значень x , а оператор P відображає цей вектор на сформоване на етапі навчання оптимальне розбиття \mathcal{R}^* класів. Оператор Ψ_1 для кожного вектора ознак, що розпізнається, обчислює значення правил рішень, які були побудовані на етапі навчання і формує множину F , а оператор Ψ_2 за значенням правила рішень, що є максимальним відносить вектор x , до одного з класів $\{X_m^o\}$. У цьому разі множина гіпотез $I^{[M+1]}$ містить додаткову гіпотезу γ_{M+1} , що характеризує рішення яке не було класифіковане, тобто вектор, який розпізнається, що не відноситься до жодного класу із заданих класів. Призначенням оператора U_E є регламентація процесу екзамену.

Таким чином, побудовані в процесі машинного навчання в рамках цього підходу правила рішень є точними і відрізняються незначною складністю обчислень і тому характеризуються високою швидкістю виконання.

Розглянуті вище моделі можна розглядати як узагальнені схеми алгоритмів машинного навчання і функціонування системи розпізнавання і сегментації під час екзамену.

Для збільшення глибини машинного навчання ми можемо також реалізувати лінійний алгоритм оптимізації параметрів машинного навчання.

2.3 Функціональні категорійні моделі інформаційно-екстремального машинного навчання з оптимізацією контрольних допусків

Концепція технології інформаційно-екстремального навчання передбачає перетворення вхідної навчальної матриці Y в робочу матрицю X , представлену в бінарному просторі Хеммінга. Ця робоча матриця X , під час машинного навчання, піддається допустимим перетворенням з метою адаптації до максимальної ймовірності рішень класифікації, які будуть правильними в цьому контексті. Для визначеного рівня глибини навчання потрібно встановити множину в просторі

ознак Хеммінга $\{g_m\}$ структурованих векторів параметрів машинного навчання, які мають вплив на ефективність системи розпізнавання.

Нехай задано вектор параметрів системи для розпізнавання реалізацій, наприклад, класу розпізнавання X_m^o за паралельно-послідовною оптимізацією контрольних допусків має вигляд [25]:

$$g_m = \langle x_m, d_m, \delta, \{\delta_i \mid i = \overline{1, N}\} \rangle \quad (16)$$

де x_m – усереднений двійковий вектор ознак;

d_m – радіус контейнера класу розпізнавання X_m^o ;

δ – параметр, величина якого дорівнює половині симетричного поля контрольних допусків для всіх ознак розпізнавання;

$\{\delta_i \mid i = \overline{1, N}\}$ – множина параметрів, величина яких дорівнює половині симетричного поля контрольних допусків i -ї ознаки розпізнавання.

Згідно з працею [25] параметр δ_i дорівнює половині симетричного поля контрольних допусків i -ї ознаки розпізнавання (рис. 2.5).

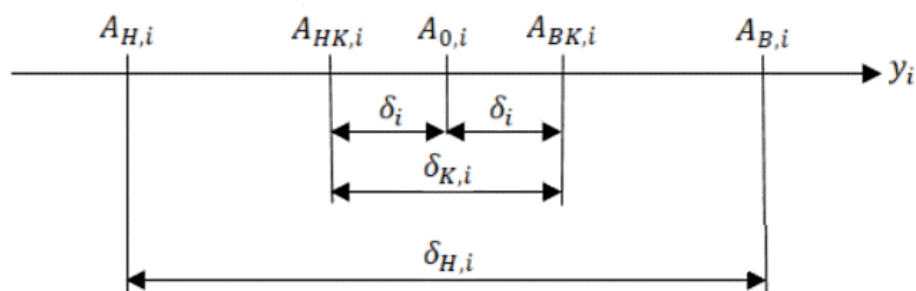


Рисунок 2.5 – Поля допусків на ознаку розпізнавання

На рисунку 2.5 визначенні такі значення: $A_{0,i}$ – номінальне значення ознаки y_i ; $A_{H,i}$ – нижнє нормоване допускне значення; $A_{B,i}$ – верхнє нормоване допускне значення; $A_{HK,i}$ – нижній контрольний допуск; $A_{BK,i}$ – верхній контрольний допуск; $\delta_{K,i}$ – поле контрольних допусків; $\delta_{H,i}$ – поле нормованих допускних значень, δ_i –

параметр, який дорівнює половині симетричного поля контрольних допусків i -ї ознаки розпізнавання.

Для зображень область значень параметрів δ і δ_i задається нерівністю

$$\delta < \delta_H / 2,$$

де δ_H – нормоване поле допусків на ознаки розпізнавання

Область значень радіуса класу розпізнавання X_m^o задається нерівністю

$$d_m < d(x_m \oplus x_c),$$

де $d(x_m \oplus x_c)$ – міжцентрова відстань класів, які є найближчими один до одного, розпізнавання X_m^o і X_c^o , яка визначається кодовою відстанню між відповідними векторами ознак усередненого значення розпізнавання x_m і x_c .

Таким чином, вектор (16) задає третій рівень глибини машинного навчання, оскільки проходять оптимізацію на першому рівні глибини два параметри: вектор середніх значень ознак розпізнавання x_m і радіус d_m контейнера класу розпізнавання X_m^o ; на другому рівні проводиться оптимізація параметра δ і на третьому рівні оптимізується параметр δ_i .

Функціональна модель включає вхідний математичний опис системи, який ми можемо представити у вигляді структури

$$I_B = \langle T, W, \Omega, K, Z, Y^{|M|}, X^{|M|}; f_1, f_2 \rangle, \quad (17)$$

де T – множина моментів в часі, коли інформація зчитується;

W – простір факторів, які впливають на зображення об'єкта на землі;

Ω – простір ознак, які розпізнаються;

K – кількість кадрів зображення;

Z – набір класів, які розпізнаються;

$Y^{|M|}$ – вхідна матриця навчання для заданого набору $\{X_m^o \mid m = 1, M\}$ класів;

$X^{|M|}$ – матриця бінарних значень навчання, задана в просторі Хеммінга ознак розпізнавання;

$f_1 : T \times W \times \Omega \times K \times Z \rightarrow Y^{|M|}$ – оператор формування вхідної матриці навчання $Y^{|M|}$, де декартовий добуток $Z \times W \times \Omega \times K \times Z$ визначає джерело отриманої інформації;

$f_2 : Y^{|M|} \rightarrow X^{|M|}$ – оператор трансформації вхідної матриці навчання $Y^{|M|}$ у робочу матрицю бінарних значень $X^{|M|}$.

На рисунку 2.6 представлена функціональна категорійна модель інформаційно-екстремального навчання третього рівня глибини. На першому рівні виконується процес оптимізації геометричних параметрів контейнерів класів розпізнавання, на другому рівні здійснюється паралельна оптимізація, а на третьому – послідовна оптимізація контрольних допусків на ознаки розпізнавання.

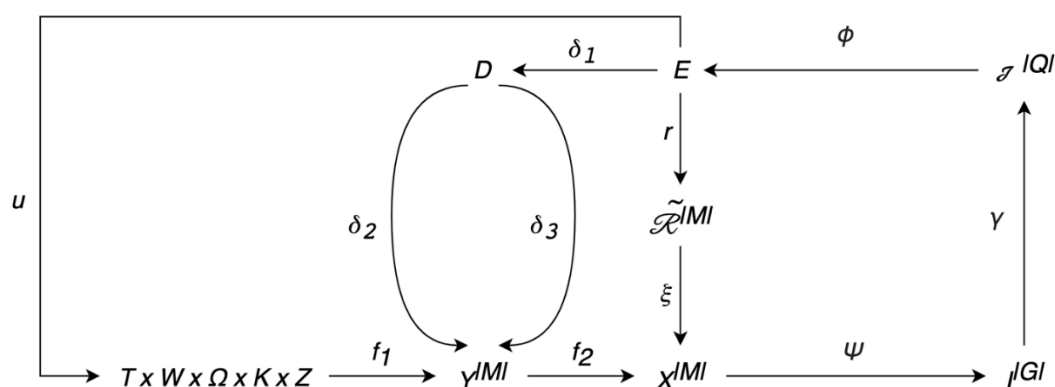


Рисунок 2.6 – Функціональна категорійна модель машинного навчання БСР із паралельно-послідовною оптимізацією контрольних допусків

На рисунку 2.6 терм-множина E значень критерію є загальною для всіх контурів оптимізації параметрів, які задані вектором (2.3). Оператор $r : E \rightarrow \tilde{\mathfrak{R}}^{|M|}$ будує на кожному кроці в загальному випадку нечітке розбиття $\tilde{\mathfrak{R}}^{|M|}$, яке відображається оператором ξ на розподіл двійкових векторів ознак розпізнавання навчальної матриці $X^{|M|}$. Далі оператор $\psi : X \rightarrow I^{|G|}$, де $I^{|G|}$ – це деяка множина G

статистичних гіпотез, які перевіряють основну гіпотезу $\gamma_1 : x_{m,i}^{(j)} \in X_m^o$. Оператор γ формує множину $\mathfrak{S}^{|Q|}$ характеристик рішень, де $Q = G^2$, а оператор φ обчислює множину значень E критерію оптимізації, яка є функціоналом від характеристик точності. Контур цієї оптимізації контрольних допусків на ознаки по яким ми можемо розпізнати об'єкт замикається через множину D , елементами якої є значення системи контрольних допусків на ознаки розпізнавання. У цьому контурі оператор δ_1 під час циклу виконання машинного навчання задає поле допусків з терм-множини D . При паралельній оптимізації оператор δ_2 змінює на кожному етапі машинного навчання допуски для всіх ознак розпізнавання, а оператор δ_3 відповідно змінює допуски на i -ту ознаку при їх послідовній оптимізації. Процес машинного навчання регламентується оператором $u : V \rightarrow G \times T \times \Omega \times K \times Z$.

Перевірка ефективності навчання проводиться під час функціонування системи в режимі екзамену. На рисунку 2.7 зображена функціональна категорійна модель у вигляді орієнтованого графа відображень множин, що використовуються при роботі системи в режимі екзамену.

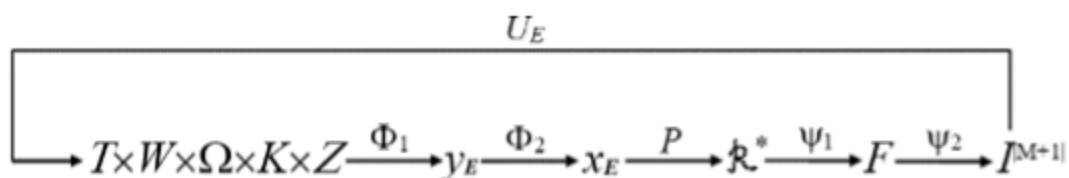


Рисунок 2.7 – Категорійна модель функціонування системи під час режиму екзамену

На рисунку 2.7 оператор Φ_1 формує з джерела інформації $T \times W \times \Omega \times K \times Z$ вектор екзамену для ознак розпізнавання y_E , який за структурою подібний векторам вхідної матриці навчання. Оператор Φ_2 за отриманими на етапі машинного навчання контрольними допусками, які є оптимальними формує вектор бінарних значень екзамену x_E , а оператор P відображає цей вектор на оптимальне чітке розбиття \mathfrak{R}^* класів розпізнавання. Оператор Ψ_1 обчислює чи належить вектор екзамену ознак до кожного класу розпізнавання, значення яких формують

терм-множину F , а оператор Ψ_2 за правилами прийняття рішень відносить вектор екзамену ознак до одного з класів $\{X_m^o\}$. Призначенням оператора U_E є опис процесу екзамену.

Таким чином, категорійні функціональні моделі, представлені на рисунках 2.6 і 2.7, можна вважати узагальненими структурними схемами алгоритмів функціонування системи як у режимі інформаційно-екстремального машинного навчання з паралельно-послідовною оптимізацією контрольних допусків на ознаки розпізнавання, так і в режимі екзамену (сегментації).

Після виконання сегментації в режимі екзамену буде сформовано основне зображення з маскою кадрів різного кольору та номера класу для отримання загальної картини семантичної сегментації.

2.4 Опис алгоритму інформаційно-екстремального машинного навчання з оптимізацією контрольних допусків на ознаки розпізнавання

Перспективним напрямком інформаційного синтезу БСР для автономних БПЛА є використання методів машинного навчання та розпізнавання образів. При цьому достовірність пошуку та розпізнавання наземних об'єктів на електронному зображенні регіону залежить від двох основних факторів:

1. релевантність вхідного математичного опису для ідентифікації кадрів цифрового зображення регіону в умовах, наближених до реальних;
2. висока ефективність методу машинного навчання, що включає достовірність та оперативність прийняття рішень класифікації системою.

Відповідно до функціональної категорійної моделі (7), загальна схема алгоритму інформаційно-екстремального машинного навчання другого рівня глибини полягає у реалізації ітераційної процедури, яка складається з двох циклів оптимізації параметра δ поля контрольних допусків на ознаки розпізнавання

шляхом пошуку максимального значення інформаційного критерію в робочій області його визначення[25].

$$\delta^* = \arg \max_{G_\delta} \left[\frac{1}{M} \sum_{m=1}^M \max_{G_{Em} \cap \{k\}} E_m^{(k)} \right], \quad (18)$$

де $E_m^{(k)}$ – значення інформаційного критерію оптимізації, обчислене на k -му кроці;

G_δ – область допустимих значень параметра поля контрольних допусків на ознаки розпізнавання;

G_{Em} – робоча область визначення критерію оптимізації параметрів машинного навчання системи для розпізнавання реалізацій класу X_m^o ;

$\{k\}$ – впорядкована множина кроків машинного навчання.

Розглянемо схему алгоритму інформаційно-екстремального машинного навчання за процедурою (18) з оптимізацією ознак паралельно для системи контрольних допусків, при якій параметр δ поля контрольних допусків змінюється одночасно для всіх ознак. При цьому вхідними даними є тривимірний масив вхідної матриці навчання $\{y[m, j, i]\}, m = \overline{1, M}, j = \overline{1, n}, i = \overline{1, N}$ для заданого алфавіту $\{X_m^o\}$ класів розпізнавання, які описують зони інтересу на зображенні, та нормоване поле δ_H допусків на ознаки розпізнавання, яке визначає область значень контрольних допусків.

Основними етапами машинного навчання є:

– вибір із заданого алфавіту базового класу розпізнавання X_1^o , відносно якого задається система контрольних допусків;

– за навчальною матрицею яскравості базового класу розпізнавання обчислюється усереднений вектор ознак \bar{y}_1 ;

– оптимізація параметрів машинного навчання для заданого рівня глибини за двохциклічною процедурою (18);

– Розробка вирішальних правил на основі оптимальних геометричних

параметрів контейнерів класу розпізнавання, отриманих у процесі машинного навчання.

– функціональне тестування з метою перевірки досягнення безпомилковості за навчальною вибіркою;

– перевірка ефективності машинного навчання в екзаменаційному режимі.

Алгоритм машинного навчання згідно з процедурою (18) реалізується за схемою:

1) приводимо лічильник класів розпізнавання до нуля: $m := 0$;

2) інкрементуємо лічильник класів розпізнавання на 1: $m := m + 1$;

3) приводимо лічильник кроків зміни параметра поля контрольних допусків до нуля: $\delta := 0$;

4) Інкрементуємо цей лічильник на 1 $\delta := \delta + 1$;

5) обчислюємо для кожної i -ї ознаки вектора \bar{y}_1 нижній $A_{HK,i}$ і верхній $A_{BK,i}$ контрольні допуски за такими формулами

$$A_{HK,i} = \bar{y}_{1,i} - \delta; A_{BK,i} = \bar{y}_{1,i} + \delta, \quad (19)$$

де $\bar{y}_{1,i}$ – значення i -ї ознаки вектора середніх значень \bar{y}_1 базового класу розпізнавання X_1^o ;

б) формуємо тривимірний масив навчальної матриці бінарних значень $\{x_{m,j,i}\}$, елементи якої обчислюються за правилом

$$x_{m,j,i} = \begin{cases} 1, & \text{if } A_{HK,i} < y_{m,j,i} < A_{BK,i}; \\ 0, & \text{if } else; \end{cases}$$

7) для масиву $\{x_{m,j,i}\}$ визначаємо вектор середніх значень \bar{x}_m ;

8) якщо $m \leq M$, то виконуємо пункт 2, інакше виконуємо пункт 12;

9) для множини векторів $\{\bar{x}_m\}$ визначаємо пари найближчих класів

розпізнавання які є сусідніми один до одного;

10) Приводимо лічильник кроків зміни радіусів контейнера класу розпізнавання до нуля $X_m^o: d_m := 0$;

11) Інкрементуємо на 1 $d_m := d_m + 1$;

12) за навчальними матрицями найближчих класів розпізнавання які є сусідніми обчислюється інформаційний критерій, наприклад, (12) оптимізації параметрів машинного навчання із відокремленням робочої області визначення його функції;

13) якщо $d_m < d(\bar{x}_m \oplus \bar{x}_c)$, де $d(\bar{x}_m \oplus \bar{x}_c)$ – міжцентрова кодова відстань класів розпізнавання X_m^o і його найближчого сусіднього класу розпізнавання X_c^o , то виконується пункт 16, інакше – пункт 14;

14) у робочій області G_E визначення функції критерію (12) обчислюється його максимальне значення;

15) визначається оптимальний радіус контейнера класу розпізнавання X_m^o :

$$d_m^* = \arg \max_{G_E \cap \{d\}} E_m(d);$$

16) якщо $t \leq M$, то виконується пункт 2, інакше – пункт 17;

17) обчислюється усереднене максимальне значення інформаційного критерію (12);

18) визначаються оптимальні значення параметра δ^* поля контрольних допусків на ознаки розпізнавання;

19) Згідно з формулами (19) обчислюються оптимальні нижні $A_{HK,i}^*$ і верхні $A_{BK,i}^*$ контрольні допуски на ознаки розпізнавання:

$$A_{HK,i}^* = \bar{y}_i - \delta^*; A_{BK,i}^* = \bar{y}_i + \delta^*;$$

20) ЗУПИН.

В результаті, в процесі машинного навчання визначається оптимальний параметр δ^* поля контрольних допусків і оптимальні параметри для всіх заданих

класів розпізнавання:

- оптимальні вектори двійкових значень $\{x_m^*\}$;
- радіуси $\{d_m^*\}$ контейнерів класів розпізнавання.

Машинне навчання з паралельною оптимізацією контрольних допусків має високий рівень ефективності, але недостатньою точно приймає правильне класифікаційне рішення. Тому для поліпшення ефективності системи є доцільним додатково впровадити алгоритм послідовної оптимізації. Одночасно, для підвищення ефективності машинного навчання, значення параметра допуску керування, що було отримане в результаті паралельної оптимізації, насправді служило вихідною точкою для алгоритму послідовної оптимізації. Алгоритм інформаційно-екстремального машинного навчання з послідовною оптимізацією контрольних допусків на ознаки розпізнавання можна схематично представити у вигляді модифікованої процедури, запропонованої автором [26]:

$$\{\delta_i^*\} = \arg \otimes_{l=1}^L \left\{ \max_{G_{\delta_i}} \left[\frac{1}{M} \sum_{m=1}^M \max_{G_{Em} \cap \{i_L\}} E_{m,i_L}^{(l)} \right] \right\}, i = \overline{1, N}, \quad (20)$$

де $E_{m,i_L}^{(l)}$ – обчислене на l -му виконанні процедури (20) значення інформаційного критерію (12);

\otimes – символ операції процесу повторення процедури (20);

G_{δ_i} – допустима область параметра δ_i поля контрольних допусків i -ї ознаки розпізнавання;

$\{i_L\}$ – множина ітерацій процесу пошуку максимального інформаційного критерію оптимізації на L повтореннях процедури (20).

На основі оптимальних геометричних параметрів контейнерів класів розпізнавання, отриманих під час машинного навчання, формуються вирішальні правила (13) для ідентифікації кадрів зображення місцевості під час роботи системи у режимах функціонального тестування та екзамену. Якщо результати

тестування підтверджують високу достовірність вирішальних правил, то вони заносяться в пам'ять системи автономного БПЛА і під час роботи.

Таким чином, ідентифікація кадрів зображення місцевості та семантична сегментація зображення здійснюються за допомогою правил рішень (13), які визначають приналежність реалізації, що розпізнається до контейнера відповідного класу розпізнавання. Водночас, через невелику складність обчислень вирішальних правил забезпечується швидкість прийняття рішень класифікації, що є важливим показником функціональної ефективності системи автономного БПЛА.

Сегментоване зображення складається з маски ідентифікованих кадрів зображення заданого розміру, який відповідає розміру кадрів словника класів.

3. МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО ДОДАТКУ СЕГМЕНТАЦІЇ ЗОБРАЖЕНЬ

3.1. Структурно-функціональне моделювання

Для опису структури реалізації проведемо моделювання програмного додатку. Створимо контекстну діаграму в нотації IDEF0, зображену на рис. 3.1.

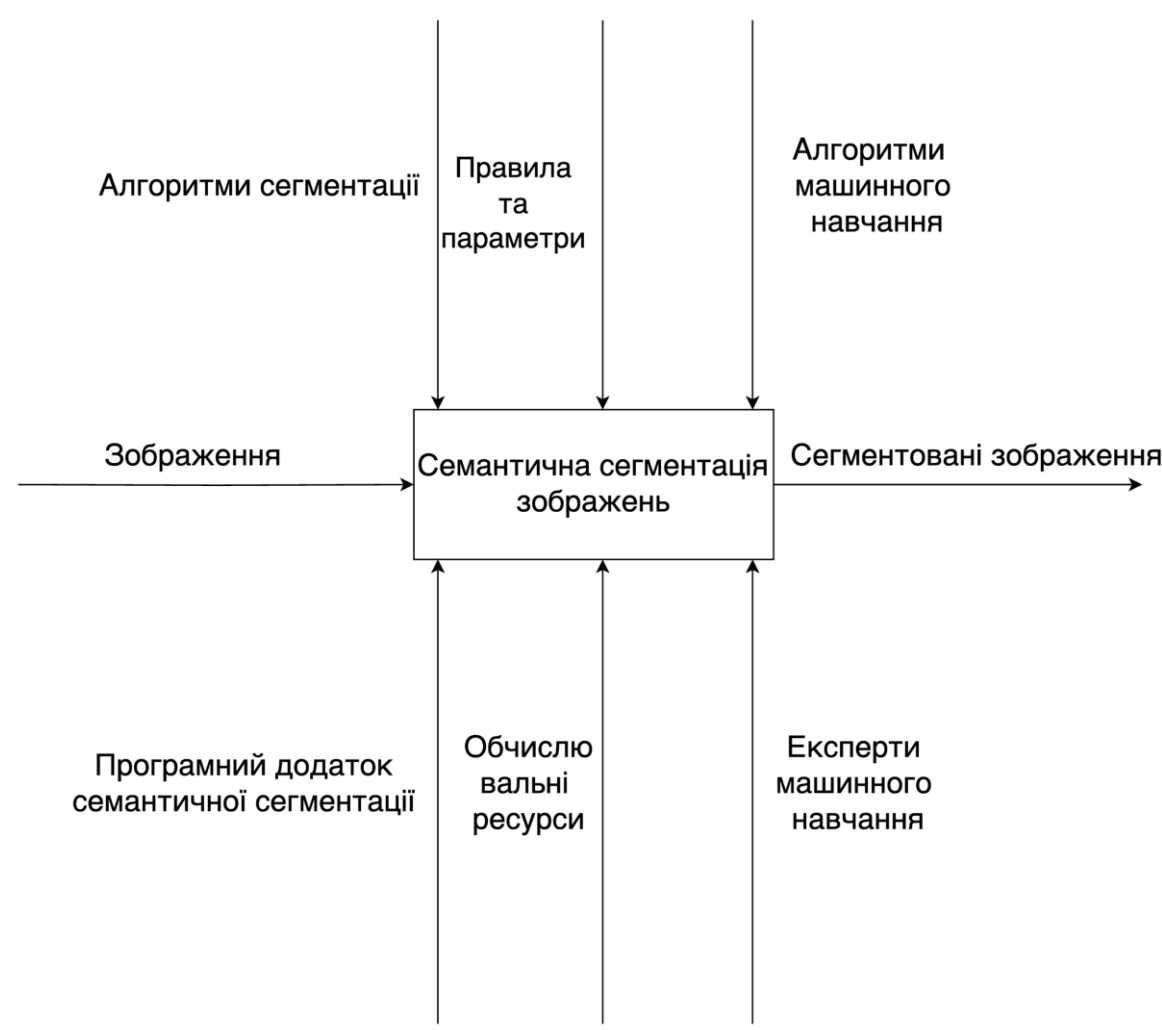


Рисунок 3.1 – Контекстна діаграма в нотації IDEF0

Тепер проведемо її декомпозицію, діаграму якої зображено на рис. 3.2.

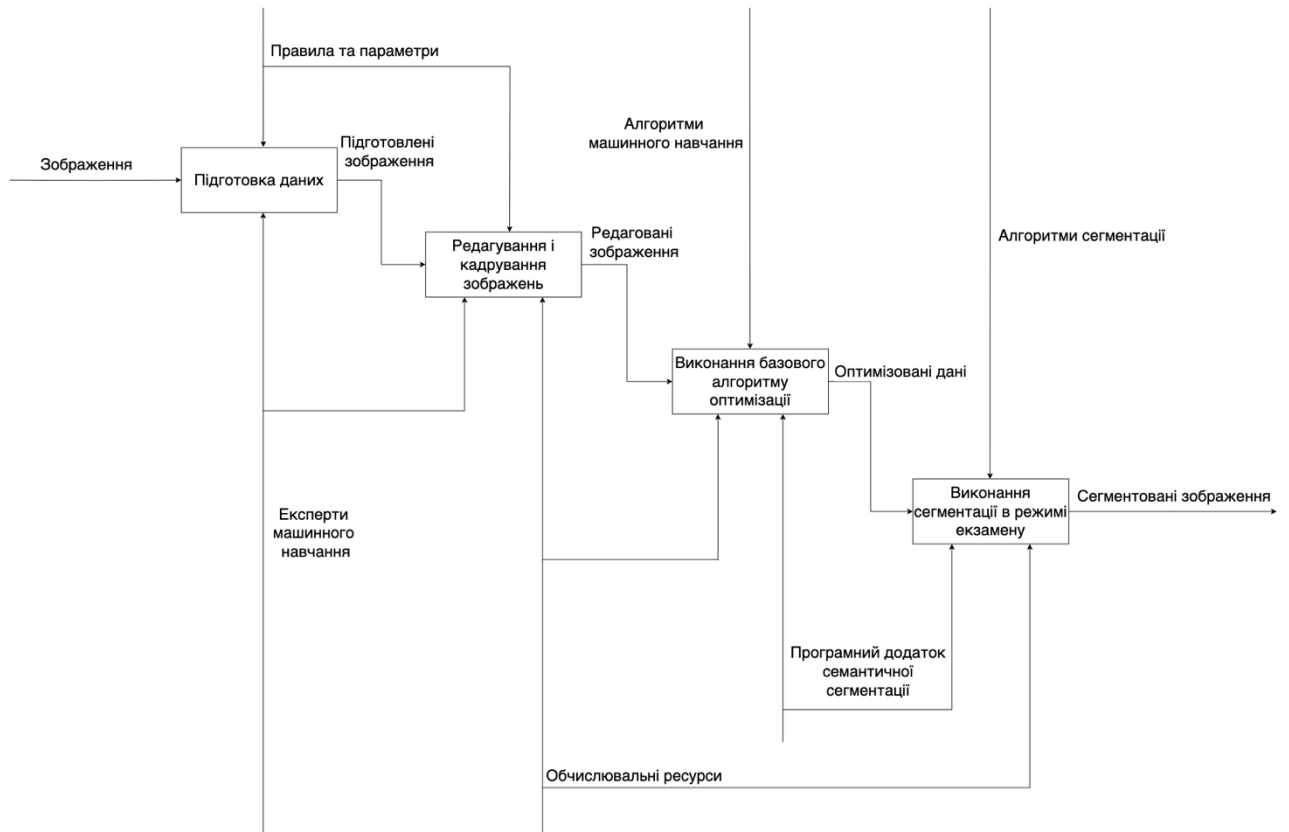


Рисунок 3.2 – Діаграма декомпозиції

Також сформуємо діаграму варіантів використання системи, для розуміння можливостей взаємодії з системою на рис. 3.3.

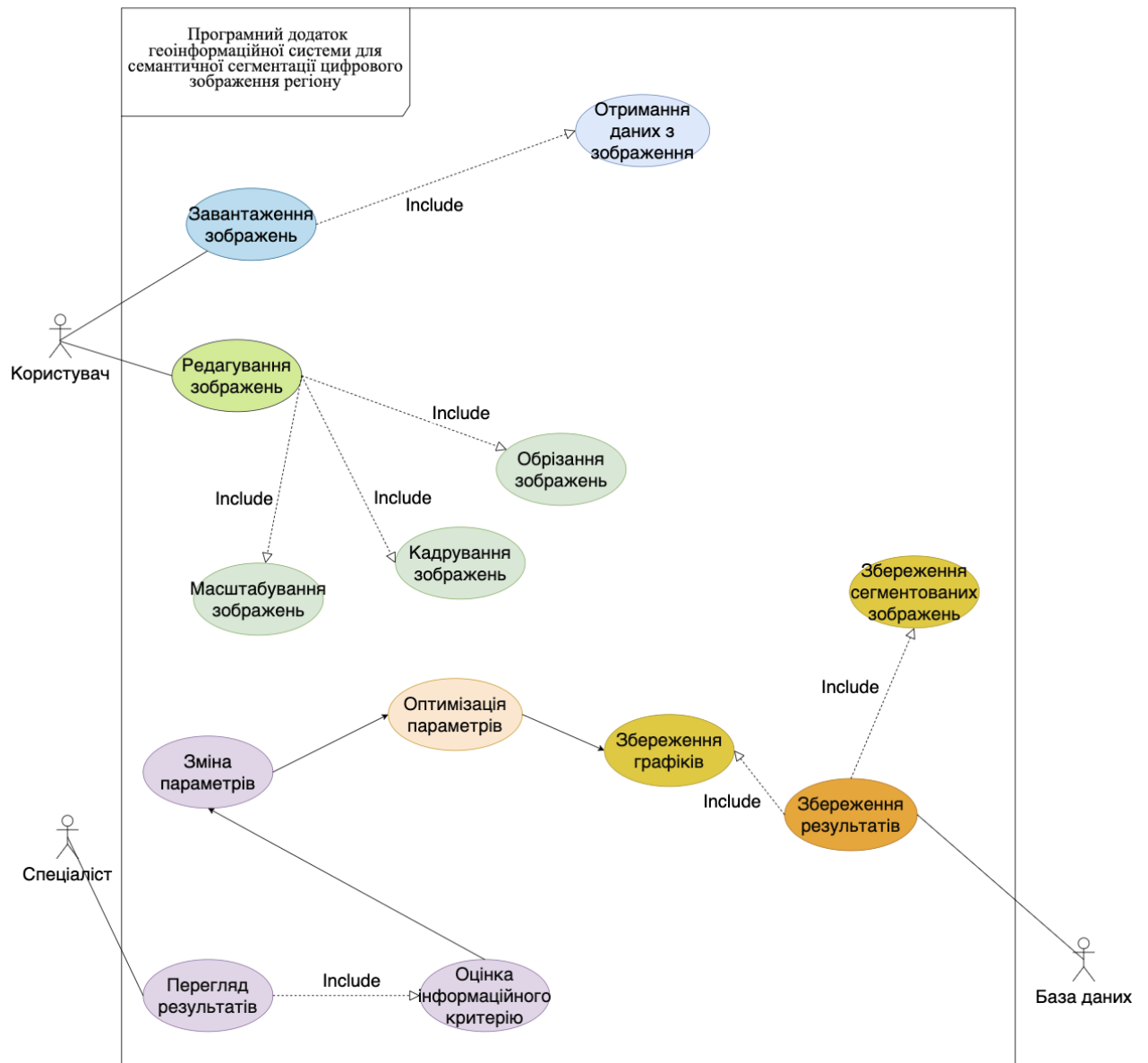


Рисунок 3.3 – Діаграма варіантів використання

Також сформуємо діаграму файлової системи, яка зображає структуру залежностей системи файлів додатку:

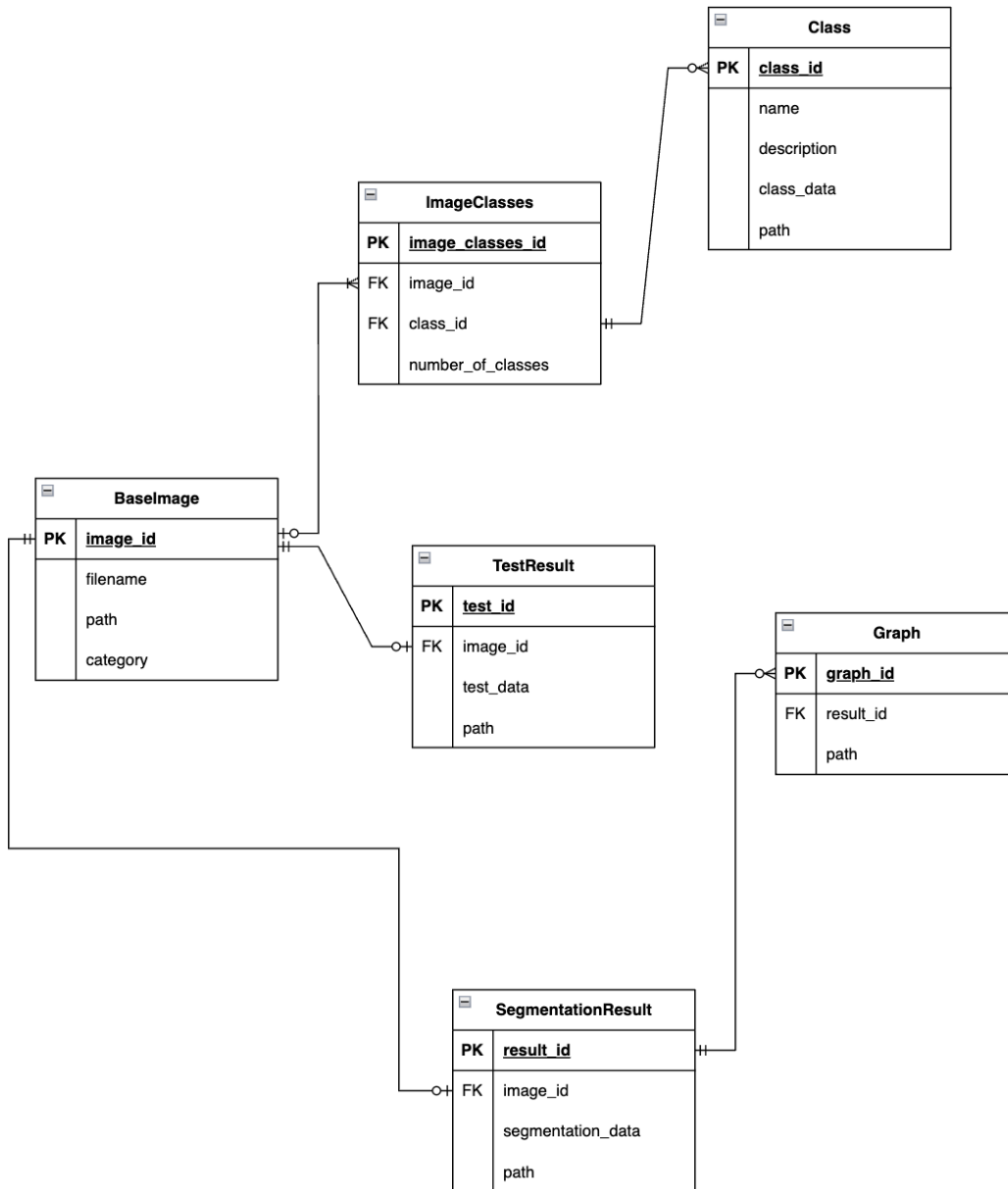


Рисунок 3.4 – ER діаграма

4 ПРОГРАМНА РЕАЛІЗАЦІЯ

4.1. Програмна реалізація і опис програмного забезпечення

Для програмної реалізації інформаційно-екстремального алгоритму машинного навчання геоінформаційної системи за лінійним алгоритмом було використано мову програмування Python.

Python є популярною мовою програмування завдяки своїй простоті, читабельності коду та широкому спектру застосувань. У веб-розробці Python використовується для створення динамічних веб-сайтів та веб-додатків. Він дозволяє швидко та ефективно розробляти проекти завдяки своїй гнучкості та простоті. У сфері науки та інженерії даних Python є ключовим інструментом для аналізу даних, проведення статистичних обчислень та створення візуалізацій, що допомагають у прийнятті рішень на основі даних.

Автоматизація і скриптинг є ще однією важливою сферою застосування Python. Мова дозволяє автоматизувати рутинні завдання, що значно підвищує продуктивність і знижує кількість помилок, пов'язаних з людським фактором. У гральній індустрії Python використовується для розробки простих ігор та прототипів завдяки своїй простоті та можливості швидкого створення робочих версій ігор.

У сфері машинного навчання Python є однією з найважливіших мов програмування. Він використовується для підготовки та аналізу даних, розробки та навчання моделей машинного навчання. Завдяки потужним засобам для роботи з даними, Python дозволяє легко очищати, маніпулювати та аналізувати великі обсяги даних. Саме тому для реалізації алгоритму було обрано саме цю мову програмування.

Для функціонування додатку системи семантичної сегментації було описано основні функції, які описані в таблиці 4.1.

Таблиця 4.1 – Функції програмного забезпечення

Функція	Призначення
read_class_image	Для зчитування даних з кадру класу
reshape_pixel_values	Для формування вхідної навчальної матриці
read_and_distribute_image	Зчитування даних зображення місцевості та розбиття зображення на кадри
show_exam_results	Відображення результатів сегментації
set_delta	Встановлення значення дельта
set_r	Встановлення розміру кадру
calc_mean_y_i	Розрахування середнього значення y_i
calc_x	Розрахунок x
calc_mean_x_i	Розрахунок середнього значення x_i
calc_dm_xm_xj	Розрахунок відстані між $mean_x_i$ та x_j
calc_dm_xm_xcj	Розрахунок відстані між $mean_x_i$ та x_{cj} , де x_{cj} - сусідній X_c до X
calc_K	Розрахунок K
calc_Ek	Розрахунок значення інформаційного критерія на кроці k
calc_Emax	Розрахунок максимального граничного значення інформаційного критерію
calc_Ek_Normal	Розрахунок нормованого значення інформаційного критерію на кроці k
calc_Mu	Розрахунок значення μ

calc_exam	Розрахунок належності кадру до одного з класів в режимі екзамену
-----------	--

Лістинг коду наведений в Додатку А.

У ході реалізації вищезгаданого алгоритму інформаційно-екстремального машинного навчання для БСР автономного БПЛА використовувалися чотири класи розпізнавання, їх зображення наведено на рисунку 2.2. Спочатку був розроблений базовий алгоритм машинного навчання, який у процедурі (18) обчислював геометричні параметри контейнерів класів розпізнавання. Початкове значення параметру дельта було встановлено на випадковому рівні, яке дорівнювало 10.

На рисунках 4.1-4.4 подано графіки залежності інформаційного критерію (12) від радіусів контейнерів класів розпізнавання, отримані під час реалізації базового алгоритму інформаційно-екстремального машинного навчання. На цих графіках виділено прямими лініями робочу (допустиму) область визначення функції інформаційного критерію (12), в якій помилки першого і другого роду не перевищують 0,5.

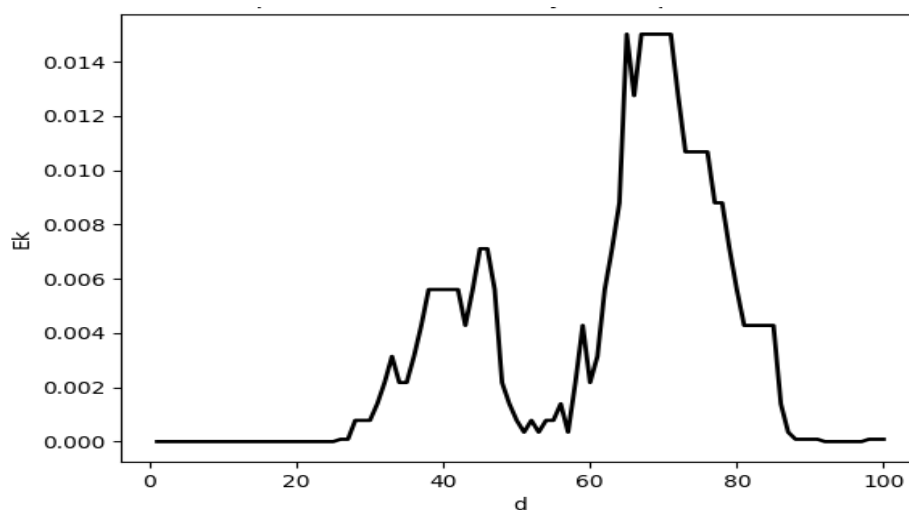


Рисунок 4.1 – Графік нормованого інформаційного критерію для класу field

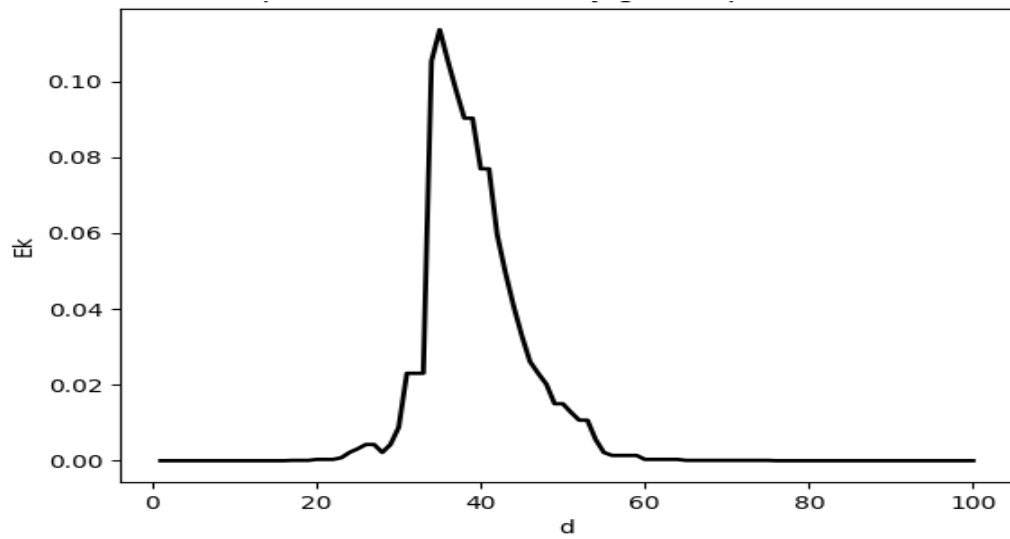


Рисунок 4.2 – Графік нормованого інформаційного критерію для класу *grass*

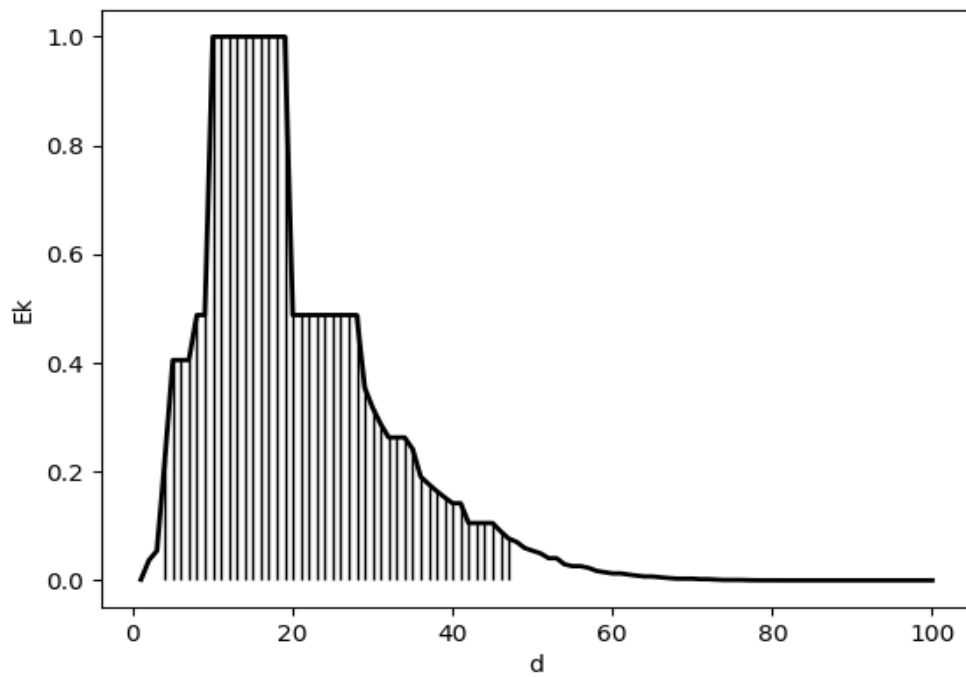


Рисунок 4.3 – Графік нормованого інформаційного критерію для класу *road*

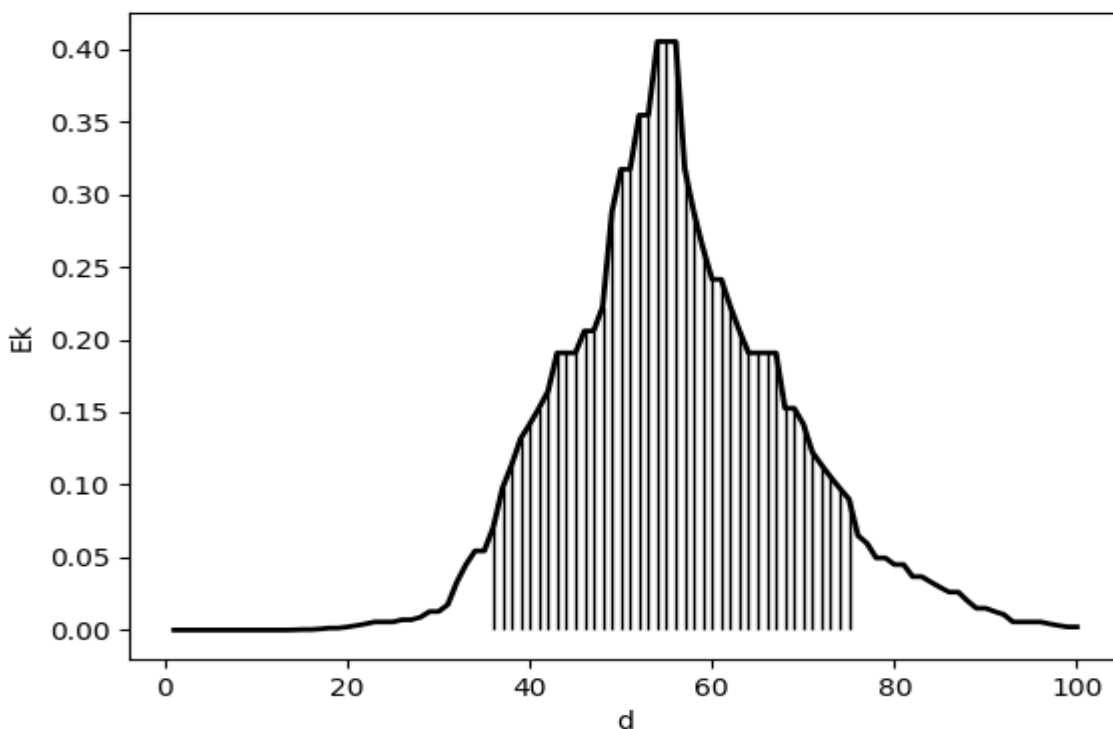


Рисунок 4.4 – Графік нормованого інформаційного критерію для класу tree

За результатами аналізу цих графіків маємо середнє значення нормованого інформаційного критерію $\bar{E}^* = 0,36$, що є невисоким через неоптимальні контрольні допуски на ознаки розпізнавання.

Для збільшення значення інформаційного критерію було реалізовано алгоритм за ітераційною процедурою (18), що дозволило оптимізувати контрольні допуски на ознаки розпізнавання.

На рисунку 4.5 показано графік залежності інформаційного критерію середніх значень критеріїв всіх класів (12) від параметра дельта поля контрольних допусків на ознаки розпізнавання, отриманий при інформаційно-екстремальному машинному навчанні з паралельною оптимізацією контрольних допусків.

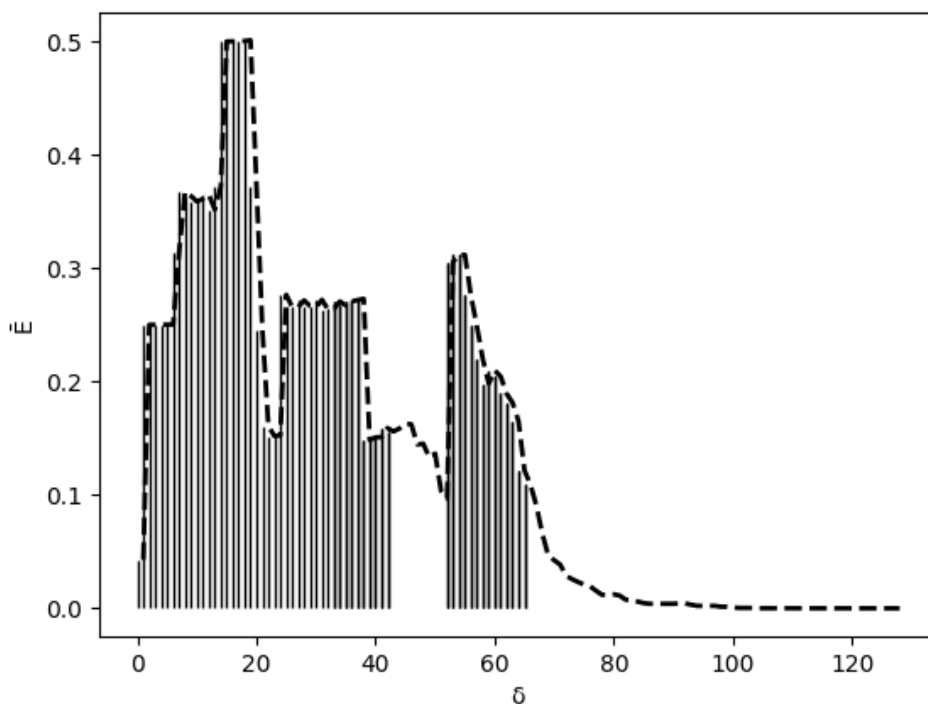


Рисунок 4.5 – Графік залежності інформаційного критерію від параметра дельта поля контрольних допусків

Аналіз рисунка 4.5 говорить нам про те, що оптимальне значення параметра дельта дорівнює $\delta^* = 19$ (у градаціях яскравості) при максимальному значенні усередненого інформаційного критерію $\bar{E}^* = 0,50$, що є покращенням в порівнянні зі значенням, отриманим при неоптимальній системі контрольних допусків на ознаки розпізнавання.

Для формування вирішальних правил (13) необхідно мати інформацію про радіуси гіперсферичних контейнерів класів розпізнавання. З цією метою на рисунках 4.6 – 4.9 наведено графіки залежності інформаційного критерію (12) від радіусів контейнерів класів розпізнавання, які були отримані в процесі інформаційно-екстремального машинного навчання БСР при оптимальному параметрі $\delta^* = 19$.

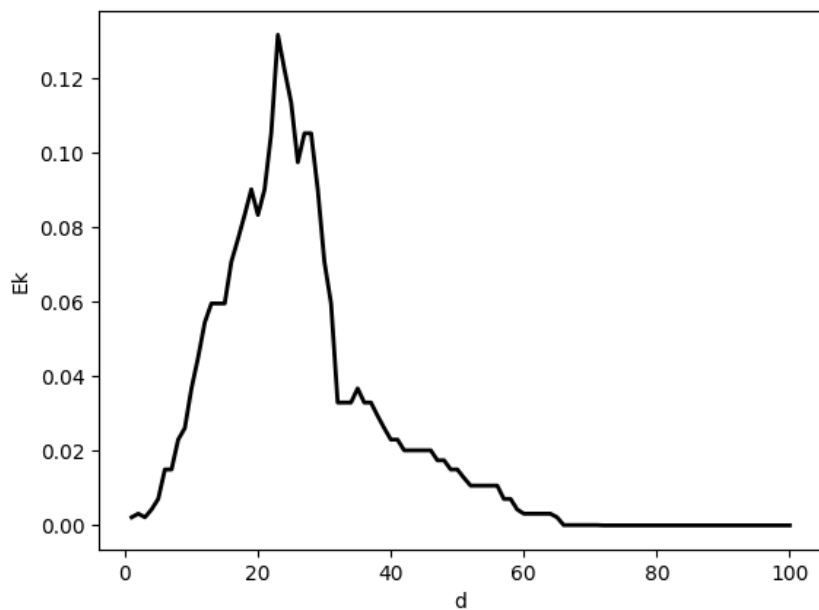


Рисунок 4.6 – Графік нормованого інформаційного критерію класу field при оптимальному дельта

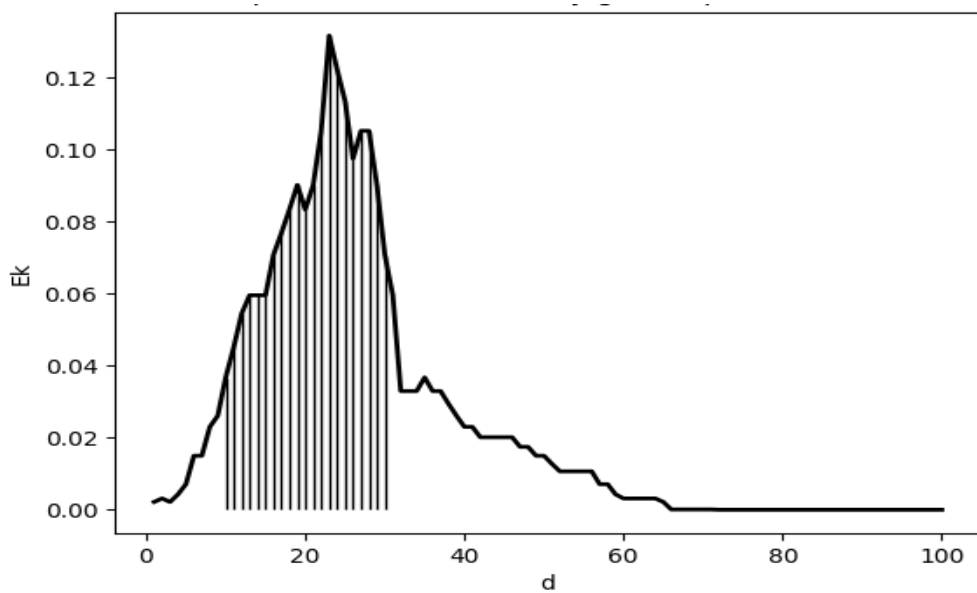


Рисунок 4.7 – Графік нормованого інформаційного критерію класу grass при оптимальному дельта

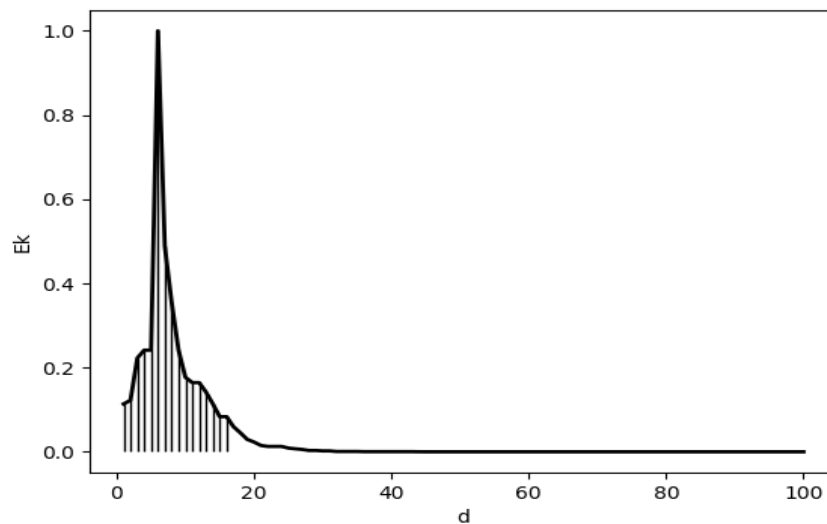


Рисунок 4.8 – Графік нормованого інформаційного критерію класу road при оптимальному дельта

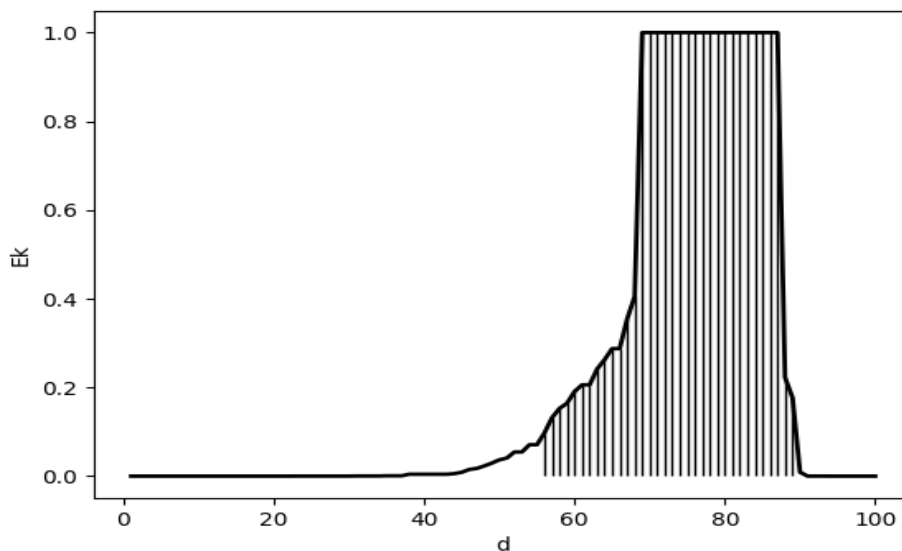


Рисунок 4.9 – Графік нормованого інформаційного критерію класу tree при оптимальному дельта

Аналіз рисунків 4.6–4.9 показує, що оптимальні радіуси контейнерів дорівнюють: $d_1^* = 24$ для класу розпізнавання X_1^o , $d_2^* = 22$ для класу розпізнавання X_2^o , $d_3^* = 9$ для класу розпізнавання X_3^o і $d_4^* = 70$ для класу розпізнавання X_4^o .

Для підвищення точності було реалізовано алгоритм інформаційно-екстремального машинного навчання з послідовною оптимізацією контрольних допусків на ознаки розпізнавання за процедурою (20).

На рисунку 4.10 показано графік зміни інформаційного критерію (12) у процесі послідовної оптимізації контрольних допусків.

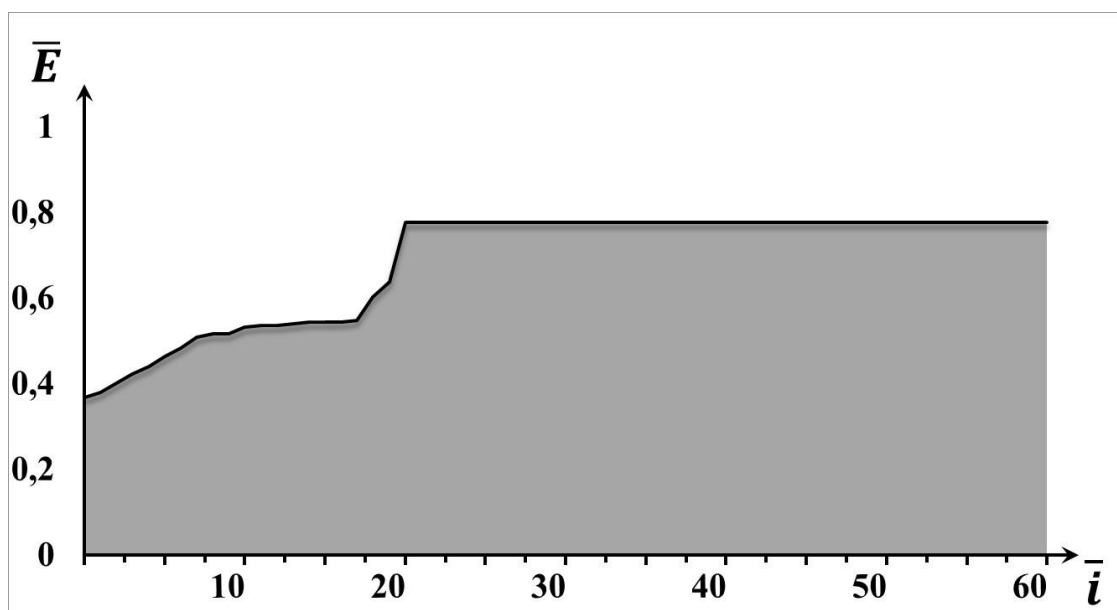


Рисунок 4.10 – Графік зміни інформаційного критерію в процесі паралельно-послідовної оптимізації контрольних допусків

Аналіз рисунку 4.10 показує, що в процесі послідовної оптимізації максимальне значення усередненого інформаційного критерію збільшилося до 0,8 вже на першому прогоні процедури (20).

По результатам аналізу процесу оптимізації ми можемо бачити збільшену ефективність семантичної сегментації за рахунок наближення до максимуму інформаційного критерію.

Для подальшого підвищення точності машинного навчання необхідно збільшити його глибину шляхом оптимізації додаткових параметрів функціонування БСР, включаючи параметри оптимізації вхідного математичного опису, що буде предметом подальших досліджень.

ВИСНОВКИ

В процесі виконання проекту було проведено аналіз предметної області семантичної сегментації цифрових зображень і огляд останніх досліджень і публікацій. За результатами аналізу методів було обрано оптимальне рішення для задачі семантичної сегментації шляхом впровадження методологій ІЕІ технології.

Результатом виконання роботи є розроблений програмний додаток який виконує задачі семантичної сегментації зображень, використовуючи методи машинного навчання.

В процесі виконання було розроблено базовий алгоритм інформаційно-екстремального машинного навчання, також розроблено алгоритм оптимізації параметрів і отримано вирішальні правила з використанням яких була проведена семантична сегментація зображень.

За результатами комп'ютерного моделювання підтверджено надійність розробленого програмного комплексу і високу точність машинного навчання для заданого алфавіту класів розпізнавання.

Проаналізувавши отримані результати було визначено доцільність збільшення глибини машинного навчання.

СПИСОК ЛІТЕРАТУРИ

1. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 234-241. URL: <https://arxiv.org/abs/1505.04597> (дата доступу: 03.05.2024).
2. Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the Knowledge in a Neural Network. *NIPS*, 2015. URL: <https://arxiv.org/abs/1503.02531> (дата доступу: 03.05.2024).
3. Chen, L.C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv preprint arXiv:1706.05587*. URL: <https://arxiv.org/abs/1706.05587> (дата доступу: 03.05.2024).
4. Dong, H., Yang, G., Liu, F., Mo, Y., & Guo, Y. (2018). Automatic brain tumor detection and segmentation using U-Net based fully convolutional networks. In *2017 International Conference on Bioinformatics and Biomedical Engineering (iCBBE)* (pp. 76-80). URL: <https://arxiv.org/abs/1705.03820> (дата доступу: 03.05.2024).
5. Isensee, F., Jaeger, P. F., Kohl, S. A., Petersen, J., & Maier-Hein, K. H. (2021). nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2), 203-211. URL: <https://pubmed.ncbi.nlm.nih.gov/33288961/> (дата доступу: 03.05.2024).
6. Kamnitsas, K., Ledig, C., Newcombe, V. F., Simpson, J. P., Kane, A. D., Menon, D. K., ... & Rueckert, D. (2017). Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Medical image analysis*, 36, 61-78. URL: <https://www.sciencedirect.com/science/article/pii/S1361841516301839> (дата доступу: 03.05.2024).
7. Chen, H., Dou, Q., Yu, L., Qin, J., & Heng, P. A. (2018). VoxResNet: Deep voxelwise residual networks for brain segmentation from 3D MR images. *NeuroImage*, 170, 446-455. Посилання

8. Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, Alan L. Yuille. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, 2018, pp. 834-848. URL: <https://pubmed.ncbi.nlm.nih.gov/28445774/> (дата доступу: 03.05.2024).
9. Olaf Ronneberger, Philipp Fischer, Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation." *International Conference on Medical image computing and computer-assisted intervention*, Springer, Cham, 2015, pp. 234-241. URL: https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28 (дата доступу: 03.05.2024).
10. Zhi Liu, Xiaoxiao Li, Ping Luo, Chen Change Loy, Xiaoou Tang. "Semantic Image Segmentation via Deep Parsing Network." *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1377-1385. URL: <https://arxiv.org/abs/1509.02634> (дата доступу: 03.05.2024).
11. Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, Hartwig Adam. "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation." *European Conference on Computer Vision*, Springer, Cham, 2018, pp. 801-818. URL: https://openaccess.thecvf.com/content_ECCV_2018/papers/Liang-Chieh_Chen_Encoder-Decoder_with_Atrous_ECCV_2018_paper.pdf (дата доступу: 03.05.2024).
12. Lihe Zhang, Xiangyu Zeng, Jianchao Yang, Zhijun Li, Yichen Wei. "Deeply Supervised Salient Object Detection with Short Connections." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3203-3212.
13. Milletari, F., Navab, N., & Ahmadi, S. A. (2016). V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)* (pp. 565-571). IEEE. URL: <https://ieeexplore.ieee.org/document/7785132> (дата доступу: 03.05.2024).

14. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., ... & Torr, P. H. (2015). Conditional random fields as recurrent neural networks. In Proceedings of the IEEE international conference on computer vision (pp. 1529-1537). URL: <https://arxiv.org/abs/1502.03240> (дата доступу: 03.05.2024).
15. Shelhamer, E., Long, J., & Darrell, T. (2017). Fully convolutional networks for semantic segmentation. IEEE transactions on pattern analysis and machine intelligence, 39(4), 640-651. URL: <https://arxiv.org/pdf/1605.06211> (дата доступу: 03.05.2024).
16. Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1492-1500). URL: <https://arxiv.org/abs/1611.05431> (дата доступу: 03.05.2024).
17. Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid scene parsing network. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2881-2890). URL: <https://arxiv.org/abs/1612.01105> (дата доступу: 03.05.2024).
18. Zhao, H., & Wang, J. (2019). Unsupervised feature learning via non-parametric instance discrimination. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3733-3742). URL: <https://arxiv.org/abs/1805.01978> (дата доступу: 03.05.2024).
19. А. С. Довбиш, А. В. Васильєв та В. О. Любчак, *Інтелектуальні інформаційні технології в електронному навчанні*, Суми: СумДУ, 201
20. Anatoliy Dovbysh, Ihor Shelehov, Anatolii Romaniuk, Roman Moskalenko and Taras Savchenko, “Decision-making support system for diagnosis of oncopathologies by histological images”, *Jurnal of Pathology Informatics*, 100193, 2023. <https://doi.org/10.1016/j.jpi.2023.100193>
21. Рощупкін О.А., Парфененко Ю.В. (2024). Тези доповіді на конференції ІМА-2024. URL: <https://files.fm/f/yawded3kqc> (дата доступу: 03.05.2024).
22. Intel Corporation. OpenCV (Version 4.6.0) [Computer software]. URL: <https://opencv.org/> (дата доступу: 03.05.2024).

23. Google LLC. TensorFlow (Version 2.9.0) [Computer software]. URL: <https://www.tensorflow.org/> (дата доступа: 03.05.2024).
24. PyTorch Contributors. PyTorch (Version 1.10.0) [Computer software]. URL: <https://pytorch.org/> (дата доступа: 03.05.2024).
25. Chollet, F., et al. Keras (Version 2.8.0) [Computer software]. URL: <https://keras.io/> (дата доступа: 03.05.2024).
26. Python Imaging Library. Python Imaging Library (PIL Fork) (Version 9.0.0) [Computer software]. URL: <https://python-pillow.org/> (дата доступа: 03.05.2024).

ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ**ТЕХНІЧНЕ ЗАВДАННЯ
на розробку інформаційної системи
«Система семантичної сегментації цифрового зображення регіону»****ПОГОДЖЕНО:**

Доцент кафедри інформаційних технологій

_____ Парфененко Ю.В.

Студент групи ІТ-02

Рощупкін О.А.

1. Призначення й мета створення системи

1.1 Призначення системи

Геоінформаційна система семантичної сегментації зображень призначена для автоматизованої обробки та сегментації великого обсягу зображень отриманих під час зйомки місцевості з висоти на основі їх характеристик та властивостей.

1.2 Мета створення системи

Метою створення інформаційної системи є спрощення процесу аналізу та сегментації зображень шляхом використання інформаційно-екстремальних алгоритмів машинного навчання та обробки зображень. Система має надавати користувачам можливість автоматичної або напівавтоматичної сегментації зображень за певними критеріями.

1.3 Цільова аудиторія системи

Цільовою аудиторією інформаційної системи є користувачі, які потребують швидкого та ефективного способу організації та сегментації за класами великого обсягу зображень. Це можуть бути фахівці у галузі комп'ютерного зору, науковці, фотографи, медичні працівники та інші спеціалісти, що працюють з великим обсягом графічних даних.

2 Вимоги до системи

2.1 Вимоги до системи в цілому

2.1.1 Вимоги до структури й функціонування системи

- Система повинна мати інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам ефективно завантажувати, обробляти та сегментувати зображення.
- Має бути реалізована можливість автоматичної та напівавтоматичної сегментації зображень на основі попередньо заданих критеріїв.
- Система повинна бути гнучкою та легко розширюватися для впровадження нових функцій та алгоритмів обробки зображень.
- Можливість ручної корекції сегментації зображень користувачем.

2.1.2 Вимоги до розробника

- Розробник повинен мати досвід у галузі машинного навчання та обробки зображень.
- Розробник повинен мати знання програмування, зокрема у мовах, які підтримують обробку зображень, таких як Python.

2.1.3 Вимоги до збереження інформації

- Система повинна забезпечувати безпечне зберігання зображень та пов'язаних з ними даних.
- Потрібно забезпечити резервне копіювання даних для запобігання втраті інформації у випадку виникнення непередбачуваних ситуацій.

2.1.4 Вимоги до розмежування доступу

- Система повинна мати два основних рівні доступу: режим сегментації і режим навчання.
- Режим сегментації буде загальнодоступним для всіх користувачів, де вони можуть використовувати систему для сегментації зображень за наявними моделями.
- Режим навчання має бути обмеженим та доступним лише для адміністратора або спеціалістів з дозволом доступом. В цьому режимі користувачі можуть навчати нові моделі машинного навчання за допомогою навчальних даних.
- Для переходу в режим навчання користувачу потрібно буде мати спеціальні права адміністратора або отримати дозвіл від адміністратора системи.
- Система повинна забезпечувати безпеку даних у режимі навчання, забезпечуючи конфіденційність та цілісність навчальних даних.

2.2 Структура системи

2.2.1 Загальна інформація про структуру системи

Система складається з таких основних компонентів:

- Модуль завантаження: призначений для завантаження зображень у систему з різних джерел, таких як локальний комп'ютер, хмарні сховища або відеострімки.
- Модуль обробки: відповідає за обробку зображень, включаючи їхнє масштабування, обрізання та попередню обробку перед сегментацією.
- Модуль сегментації: використовує навчені моделі машинного навчання для семантичної сегментації зображень за певними класами.
- Модуль адміністрування: надає можливість керувати користувачами, правами доступу, навчальними даними та іншими налаштуваннями системи.

2.2.2 Навігація

Навігація в системі здійснюється за допомогою інтуїтивно зрозумілих інтерфейсів користувача, які дозволяють переходити між різними модулями та функціональними можливостями. Меню навігації може містити такі пункти, як "Завантажити зображення", "Сегментація", "Навчити модель", "Керувати користувачами" та інші.

2.2.3 Наповнення системи (контент)

Наповнення системи передбачає завантаження зображень та навчання моделей машинного навчання. Користувачі можуть додавати свої власні зображення для сегментації або використовувати наявні навчальні набори даних для навчання моделей.

2.2.4 Функціональна архітектура системи

1. Модуль завантаження зображень:

- Забезпечує можливість користувачам завантажувати зображення у систему.
- Перевіряє та обробляє вхідні дані для подальшої обробки.

2. Модуль обробки зображень:

- Відповідає за масштабування, обрізання та попередню обробку зображень перед сегментацією.
- Перетворює зображення у формат, придатний для подальшої обробки.

3. Модуль сегментації зображень:

- Використовує навчені моделі машинного навчання для сегментації зображень за їх характеристиками та властивостями.
- Надає результати сегментації у зручному для подальшого аналізу форматі.

4. Модуль ручної корекції сегментації:

- Забезпечує можливість користувачам вручну коригувати результати сегментації у випадках потреби.

5. Модуль управління навчанням моделей:

- Надає можливість навчати нові моделі машинного навчання на основі навчальних даних для покращення результатів сегментації.

2.2.5 Система навігації (карта системи)

Карта системи зображена на рисунку А.2.

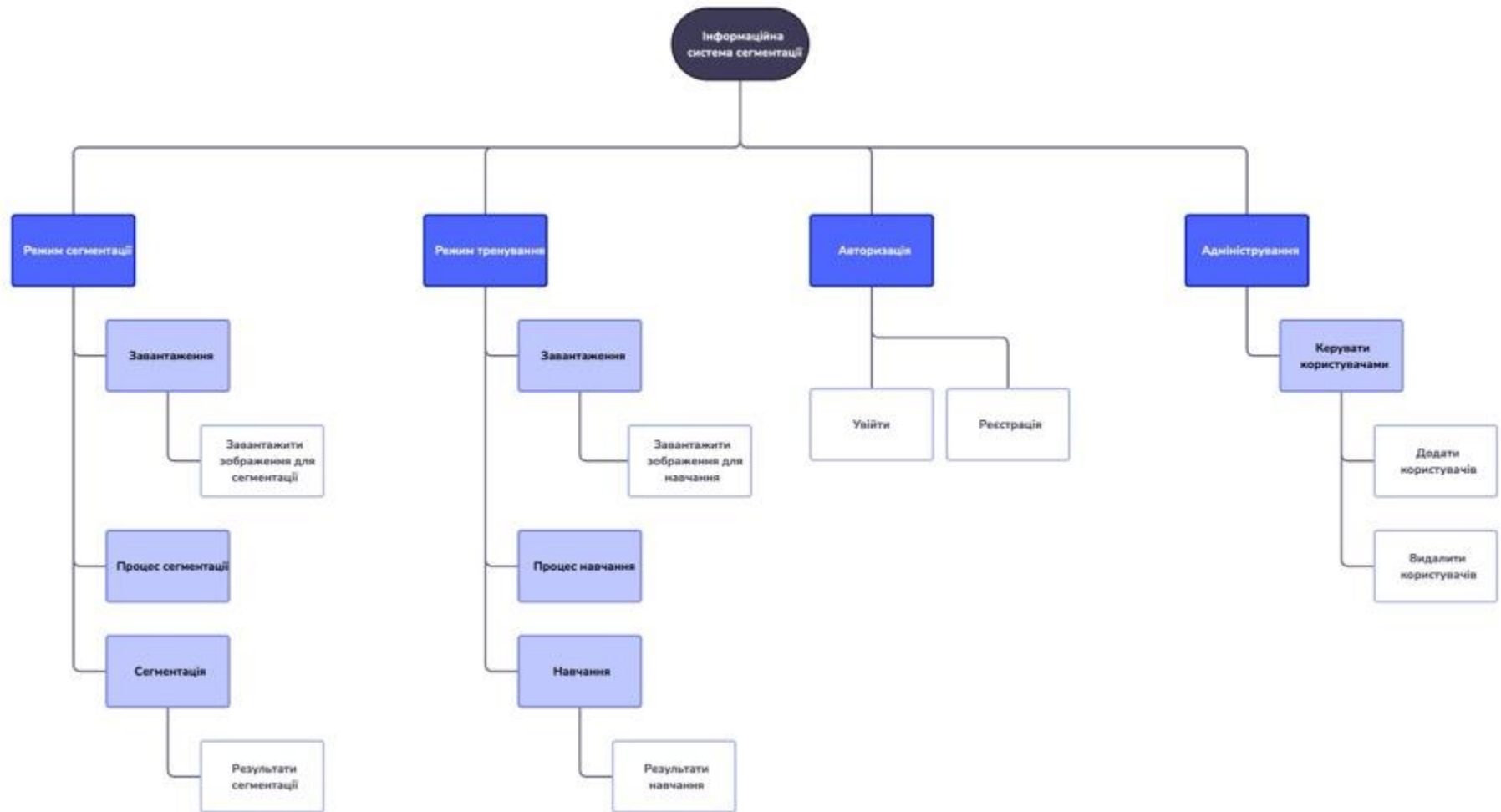


Рисунок А.2 – Карта додатку

2.3 Вимоги до функціонування системи

2.3.1 Потреби користувача

Потреби користувача представлені у таблиці А.1.

Таблиця А.1 – Потреби користувача

ID	Потреби користувача	Джерело
UN-01	Завантаження зображень для сегментації	Користувач
UN-02	Перегляд результатів сегментації зображень	Користувач
UN-03	Можливість ручної корекції сегментації зображень	Користувач
UN-04	Запуск автоматичної сегментації зображень	Користувач
UN-05	Перегляд статистики та звітів про сегментацію зображень	Адміністратор
UN-06	Доступ до детальної інформації про сегментаційні моделі	Адміністратор
UN-07	Можливість керування користувачами та їх правами доступу	Адміністратор

2.3.2 Функціональні вимоги

На основі потреб користувача, визначених для інформаційної системи сегментації зображень, сформульовані такі функціональні вимоги:

1. Завантаження зображень: Система повинна дозволяти користувачам завантажувати великі обсяги зображень для подальшої обробки та сегментації.

2. Обробка зображень: Система має автоматизовано обробляти завантажені зображення шляхом їх масштабування, обрізання та іншої попередньої обробки.
3. Сегментація зображень: Система повинна використовувати характеристики та властивості зображень для автоматичної сегментації на основі навчених моделей машинного навчання.
4. Ручна корекція сегментації: Адміністраторам має бути надана можливість ручної корекції результатів сегментації у випадках потреби.
5. Візуалізація результатів: Система повинна відображати результати сегментації зображень у зручному для користувача форматі, що дозволяє їм аналізувати та здійснювати прийняття рішень.
6. Управління навчальним процесом: Для покращення результатів сегментації, система може надавати можливість адміністраторам навчати нові моделі машинного навчання на основі навчальних даних.
7. Моніторинг та аналітика: Система має забезпечувати можливість моніторингу та аналізу результатів сегментації зображень для виявлення тенденцій та покращення ефективності системи.

2.3.3 Системні вимоги

Даний розділ визначає, розподіляє та вказує на системні вимоги, визначені розробником. Їх перелік наведений в таблиці А.2.

Таблиця А.2 – Системні вимоги

ID	Системні вимоги	Пріоритет	Опис
SR-01	Наявність модуля завантаження зображень	M	Забезпечує можливість завантаження зображень користувачем
SR-02	Модуль обробки зображень	M	Виконує автоматизовану обробку зображень перед сегментацією
SR-03	Модуль сегментації зображень	M	Забезпечує автоматичну сегментацію зображень за їхніми характеристиками
SR-04	Модуль ручної корекції сегментації	S	Надає можливість користувачеві вручну коригувати результати сегментації
SR-05	Модуль управління навчанням моделей	S	Забезпечує можливість навчання нових моделей на основі навчальних даних

SR-06	Модуль відображення результатів	М	Відображає результати сегментації зображень у зручному форматі
SR-07	Модуль моніторингу та аналітики	С	Забезпечує моніторинг та аналіз результатів сегментації для покращення ефективності системи

Умовні позначення в таблиці А.2:

Must have (M) – вимоги, які повинні бути реалізовані в системі;

Should have (S) – вимоги, які мають бути виконані, але вони можуть почекати своєї черги;

Could have (C) – вимоги, які можуть бути реалізовані, але вони не є центральною ціллю проекту.

2.4 Вимоги до видів забезпечення

2.4.1 Вимоги до інформаційного забезпечення

- Система повинна використовувати базу даних для зберігання та організації інформації про зображення. Рекомендується використання реляційних баз даних, таких як PostgreSQL або MongoDB, для ефективного зберігання та оптимізації доступу до даних.
- Для обробки зображень система може використовувати бібліотеки обробки зображень у Python, такі як OpenCV або PIL, для масштабування, обрізання та іншої попередньої обробки.
- Для реалізації сегментації зображень система використовуватиме інформаційно-екстремальний алгоритм машинного навчання.
- Для забезпечення інтерфейсу користувача та візуалізації результатів система може використовувати веб-фреймворк Django або Flask для реалізації веб-додатку, а також JavaScript та бібліотеку Plotly для відображення результатів сегментації у зручному форматі.

2.4.2 Вимоги до лінгвістичного забезпечення

- Наявність інтерфейсу користувача, що підтримує різні мови для зручності користувачів з різних країн та культур.
- Забезпечення можливості перекладу інтерфейсу та документації системи на різні мови для міжнародного використання.

2.4.3 Вимоги до програмного забезпечення

- Вимоги до програмного забезпечення включають в себе використання інформаційно-екстремальних алгоритмів машинного навчання для сегментації зображень.
- Наявність системи управління базами даних для зберігання та обробки інформації про зображення.

- Забезпечення платформонезалежності та сумісності з різними операційними системами для максимальної доступності системи користувачам.

3 Склад і зміст робіт зі створення системи

Докладний опис етапів роботи зі створення системи наведено в таблиці А.3.

Таблиця А.3 – Етапи створення системи

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Аналіз вимог та специфікація вимог	2
2	Проектування архітектури системи	2
3	Розробка інфраструктури системи	1
4	Розробка модуля завантаження зображень	2
5	Розробка модуля обробки зображень	2
6	Розробка модуля сегментації зображень	4
7	Розробка модуля ручної корекції сегментації	2
8	Розробка модуля управління навчанням моделей	2
9	Тестування та відлагодження системи	2
10	Розгортання та підтримка системи	2
	Загальна тривалість розробки	21

4 Вимоги до складу й змісту робіт із введення в експлуатацію

Розгортання системи на сервері:

- Налаштування серверного середовища для запуску інформаційної системи сегментації зображень.
- Встановлення необхідних бібліотек та залежностей для роботи системи.

Імпорт та налаштування даних:

- Завантаження та імпорт вихідних даних для подальшої сегментації.
- Налаштування бази даних для ефективного зберігання та доступу до зображень.

Підготовка моделей сегментації:

- Навчання моделей машинного навчання на основі вихідних даних.
- Оцінка та вибір найкращих даних для включення в систему.

Тестування та налагодження:

- Проведення тестів для перевірки точності та надійності сегментації.
- Виявлення та виправлення помилок та недоліків у роботі системи.

ДОДАТОК Б.

Деталізація мети проекту методом SMART. Продуктом дипломного проекту є система для семантичної сегментації цифрового зображення.

Результати деталізації методом SMART розміщені у табл. Б.1.

Таблиця Б.1 – Деталізація мети методом SMART

Критерії SMART	Опис
Специфічна (Specific)	Розробка системи для сегментації цифрових зображень на основі методів машинного навчання.
Вимірювана (Measurable)	Реалізація алгоритму, який забезпечить точність сегментації зображень.
Досяжна (Achievable)	Використання інформаційно-екстремального інтелектуального алгоритму для досягнення поставленої мети.
Релевантна (Relevant)	Враховуючи широкий спектр застосувань сегментації зображень, розробка цієї системи є актуальною та корисною.
Часові обмеження (Time-bound)	Розробка та впровадження системи сегментації зображень повинно бути завершено протягом 6 місяців.

Планування змісту структури робіт. Основним інструментом для планування змісту структури робіт служить WBS діаграма – графічне подання згрупованих елементів проекту у вигляді пакета робіт, які ієрархічно пов'язані з продуктом проекту. Побудуємо структуру WBS, у якій детально опишемо роботи, які потрібно виконати на кожному етапі створення проекту. Виконаємо декомпозицію робіт для даного проекту. Діаграма WBS зображена на рис. Б.1.

Планування структури організації, для впровадження готового проекту (OBS). Після побудови WBS розробимо організаційну структуру виконавців OBS. Організаційна структура

проекту стосується тільки внутрішньої організаційної структури проекту і не стосується відносин проектних груп чи учасників з батьківськими організаціями. Діаграма OBS зображена на рис. Б.2. Список виконавців, що функціонують в проекті знаходиться в табл. Б.2.

Таблиця Б.2 – Виконавці проекту

Роль	Ім'я	Проектна роль
Розробник	Роцупкін О. А.	Виконує розробку основного функціоналу проекту та
Проектувальник	Роцупкін О. А.	Проектує алгоритм інформаційно-екстремальної інтелектуальної системи.
Тестувальник	Роцупкін О. А.	Відповідає за тестування функціоналу та перевірку моделі на адекватність.
Косультант проекту	Довбиш А. С.	Формує завдання на розробку проекту, технологію алгоритму системи.
Менеджер проекту	Парфененко Ю. В.	Відповідає за виконання термінів, розподіл ресурсів та завдань між учасниками. Виконує збір та аналіз даних.

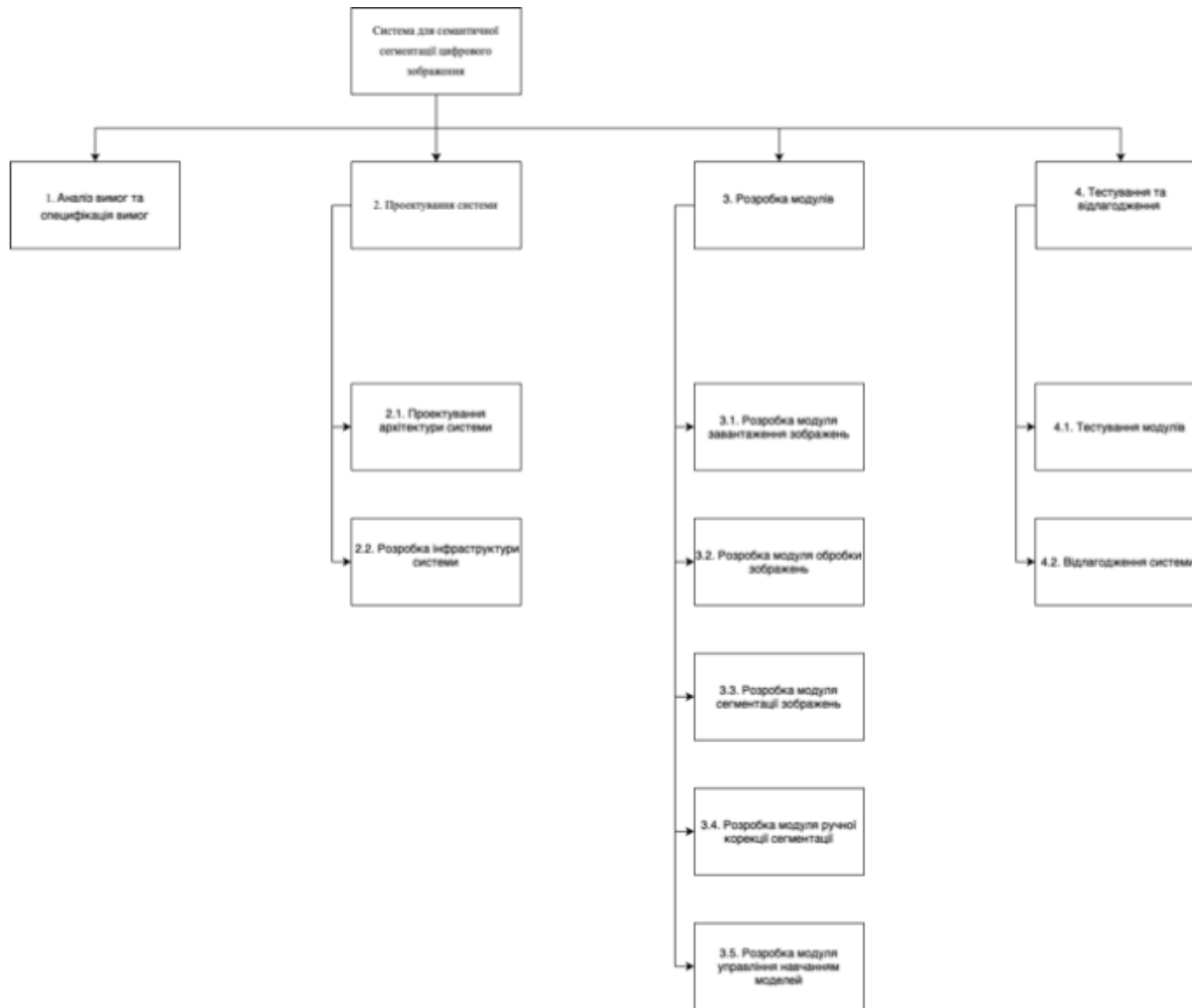


Рисунок Б.1 – WBS. Структура робіт проекту

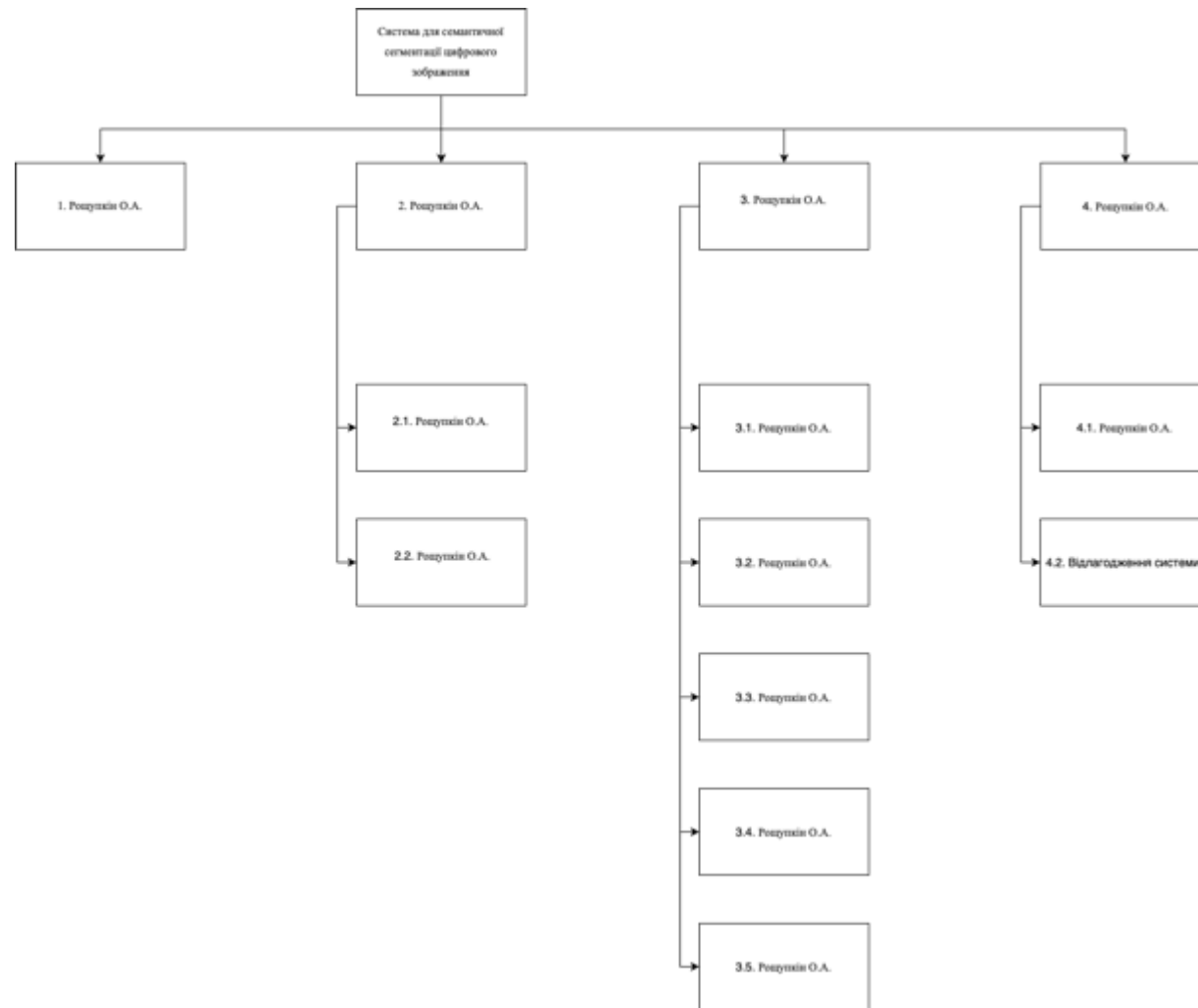


Рисунок Б.2 – Організаційна структура проекту (OBS)

Діаграма Ганта. Далі побудуємо календарний план виконання дипломного проекту. Найпоширеніший формат графіка в будь-якій галузі — діаграма Ганта. Цей графік дозволяє менеджерам проекту і всій команді розробників візуалізувати графіки часу і взаємозв'язок між окремими завданнями та етапами роботи над проектом. Тривалість виконання робіт зазначена в днях. Для того щоб мати реальне уявлення про тривалість виконання робіт з урахуванням обмеженості у використанні ресурсів, з урахуванням вихідних та святкових днів, побудовано календарний графік. Діаграма Ганта та список робіт діаграми Ганта зображені на рис. Б.3, Б.4.

ID	Name	Start Date	End Date	Duration
1	▼ Аналіз вимог та специфікація вимог	Apr 10, 2024	Apr 29, 2024	14 days
2	▼ Проектування системи	Apr 10, 2024	Apr 11, 2024	2 days
3	Проектування архітектури системи	Apr 10, 2024	Apr 10, 2024	1 day
4	Розробка інфраструктури системи	Apr 11, 2024	Apr 11, 2024	1 day
5	▼ Розробка модулів	Apr 12, 2024	Apr 29, 2024	12 days
6	Розробка модуля завантаження зображ...	Apr 12, 2024	Apr 15, 2024	2 days
7	Розробка модуля обробки зображень	Apr 15, 2024	Apr 16, 2024	2 days
8	Розробка модуля сегментації зображень	Apr 16, 2024	Apr 19, 2024	4 days
9	Розробка модуля ручної корекції сегме...	Apr 22, 2024	Apr 23, 2024	2 days
10	▼ Розробка модуля управління навчанням...	Apr 24, 2024	Apr 29, 2024	4 days
11	Тестування та відлагодження	Apr 24, 2024	Apr 25, 2024	2 days
12	Тестування модулів	Apr 24, 2024	Apr 24, 2024	1 day
13	▼ Відлагодження системи	Apr 26, 2024	Apr 29, 2024	2 days
14	Розгортання та підтримка системи	Apr 26, 2024	Apr 29, 2024	2 days

Рисунок Б.3 – Список робіт для побудови діаграми Ганта

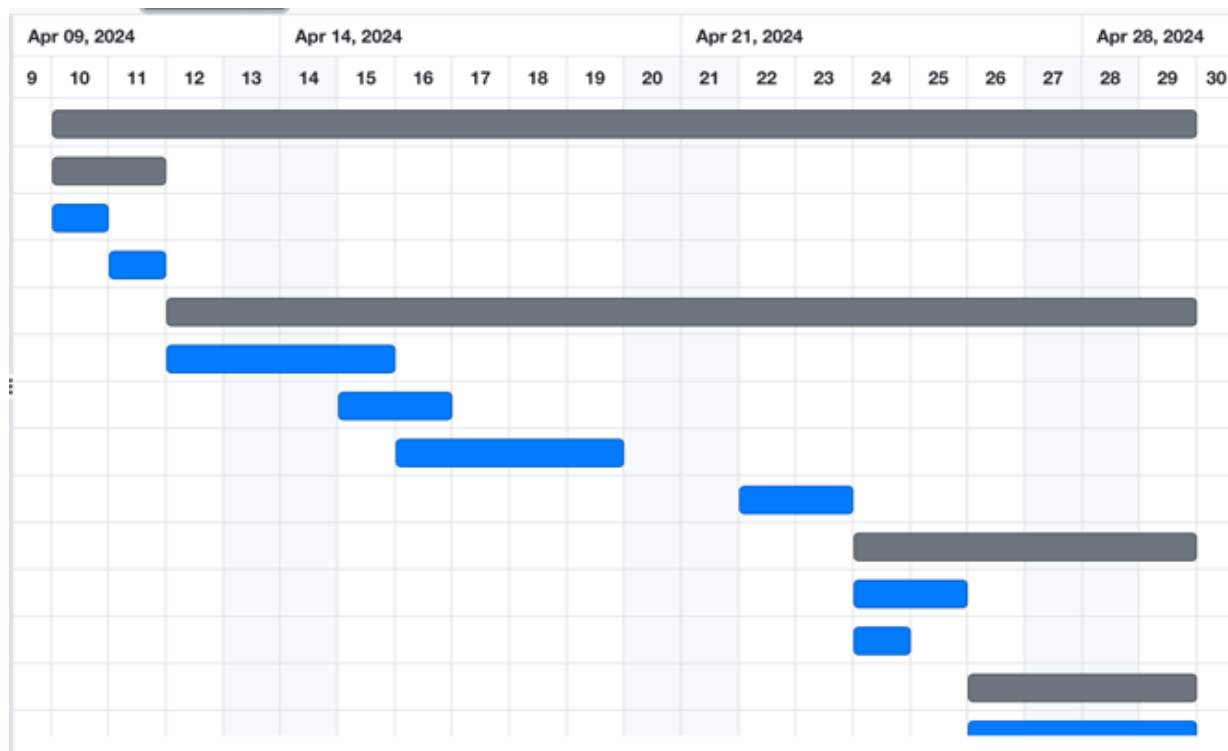


Рисунок Б.4 – Діаграма Ганта

Аналіз ризиків. Виконаємо якісну і кількісну оцінку ризиків роботи. При якісній оцінці визначимо ризики, що потребують швидкого реагування. Така оцінка визначить ступінь важливості ризику і дозволить вибрати спосіб реагування. Кількісна оцінка ризиків буде виконана для більш повної ідентифікації ризиків та ступеня їхнього впливу на виконання проекту. Кількісна і якісна оцінка ризиків можуть використовуватися окремо або разом, залежно від наявного часу і бюджету, необхідності в кількісній або якісній оцінці ризиків. У табл. Б.5 знаходиться класифікація ризиків за показниками ймовірності виникнення ризику та величині втрат.

Далі виконаємо планування реагування на ризики — це розробка методів і технологій зниження негативного впливу ризиків на проект. Визначимо ефективність розробки реагування на проект, визначимо чи будуть наслідки впливу ризику на проект позитивними або негативним. Оцінюємо ризики за показниками, що

знаходяться в табл. Б.3. На основі оцінки будуємо матрицю ймовірності виникнення ризиків та впливу ризику, що зображена на рис. Б.5.

Таблиця Б.3 – Шкала оцінювання ймовірності виникнення та впливу ризику на виконання проекту

Оцінка	Ймовірність виникнення	Вплив ризику
1	Низька	Низький
2	Середня	Середній
3	Висока	Високий

Ймовірність виникнення	3	RS_2	RS_3,	RS_9
	2	RS_1	RS_3, RS_7	-
	1	RS_4, RS_10	RS_5	RS_8
		1	2	3
		Вплив ризику		

Рисунок Б.5 – Матриця ймовірності виникнення ризиків та впливу ризику

- зелений колір – прийнятні ризики;
- жовтий колір – виправданні ризики;
- червоний колір – недопустимі ризики.

На підставі отриманого значення індексу ризику класифікують: за рівнем ризику, що знаходиться в табл. А.4.

Таблиця Б.4 – Шкала оцінювання за рівнем ризику

№	Назва	Межі	Ризики, які входять(номера)
1	Прийнятні	$1 \leq R \leq 2$	1, 4, 5, 10
2	Виправдані	$3 \leq R \leq 4$	2, 3, 7, 8
3	Недопустимі	$6 \leq R \leq 9$	3, 9

Таблиця Б.5 – Оцінка ймовірності виникнення, величини витрат та індексу ризику

D	I	Ст	Опис	Ймов	Вп	Р	План	Тип	План
			ризик	ірність	лив	анг	А	стратегії	Б
				виникненн	ризик	ризику		реагування	
				я					
S-1	Від критий	Від критий	Втрати даних через некоректне завантаження зображень	Низька	Високий	2	Регулярне резервне копіювання даних	Перенос	Використання стороннього сервісу для автоматичної верифікації зображень
S-2	Від критий	Від критий	Помилки під час навчання моделей машинного навчання	Середня	Високий	3	Розробка механізмів перевірки та валідації даних перед навчанням моделей	Пом'якшення	Залучення додаткових експертів для аналізу та корекції помилок
S-3	Від критий	Від критий	Непередбачені зміни вимог	Висока	Високий	5	Проведення регулярних	Перенос	Розробка модульної

		до системи під час розробки				зустрічей з замовником для уточнення вимог		системи з можливістю легкої модифікації
S-4	Від критий	Втрати доступу до серверів через технічні проблеми	Низька	Середній	1	Проведення регулярних технічних оглядів та профілактичних робіт	Прийняття	Використання хмарних технологій для автоматичного перенаправлення трафіку
S-5	Від критий	Низька продуктивність моделей машинного навчання	Середня	Високий	4	Оптимізація алгоритмів та параметрів навчання	Пом'якшення	Залучення додаткових експертів для аналізу та вдосконалення моделей
S-6	Від критий	Втрати ключового члена команди	Низька	Високий	6	Розробка плану перерозподілу обов'язків	Пом'якшення	Розробка плану навчання та підготовки

								іншого члена команди
S-7	Від критий	Затримки в поставках обладнання та програмного забезпечення	Середня	Середній	3	Розробка плану альтернативних джерел поставок	Пом'якшення	Збільшення запасів обладнання та програмного забезпечення
S-8	Від критий	Погана комунікація в команді проекту	Середня	Середній	3	Запровадження системи регулярних звітів та зустрічей	Прийняття	Призначення координатора комунікацій
S-9	Від критий	Втрата доступу до даних через збої в мережі	Висока	Середній	7	Розробка плану резервного забезпечення мережі	Попередження	Підписання додаткових угод з постачальниками мережевого обладнання
S-10	Від критий	Неправильна	Низький	Високий	2	Постійний	Попередження	Залучення

		оцінка обсягу робіт та термінів виконання проекту				моніторинг прогресу робіт та вчасна корекція планів		незалежних експертів для оцінки обсягу робіт та термінів
--	--	---	--	--	--	--	--	--

ДОДАТОК В. ЛІСТИНГ ПРОГРАМИ

ImageClass:

```
import PIL as PIL

from PIL import Image, ImageDraw, ImageFont

import numpy as np

import math

from datetime import datetime

import pandas as pd

class ImageClass:

    def __init__(self, class_name):

        self.class_name = class_name

        self.class_image_data = None

        self.class_brightness_by_pixel = None

        self.rmax = None

        self.channels = 3

    @staticmethod

    def convert_to_polar(pixel_values, width, height, rmax, channels):

        center_x = width // 2

        center_y = height // 2

        polar_pixel_values = []
```

```

for r in range(rmax):
    for angle in np.linspace(0, 2 * math.pi, 360):
        x = int(center_x + r * math.cos(angle))
        y = int(center_y + r * math.sin(angle))

        if x < 0 or x >= width or y < 0 or y >= height:
            polar_pixel_values.append(0)
            polar_pixel_values.append(0)
            polar_pixel_values.append(0)
        else:
            pixel_index = y * width + x
            pixel = pixel_values[pixel_index]
            polar_pixel_values.append(pixel[0])
            polar_pixel_values.append(pixel[1])
            polar_pixel_values.append(pixel[2])

image_data = np.array(polar_pixel_values).reshape((rmax, 360 * channels))
image_data = image_data.transpose()

return image_data

def read_class_image(self, polar=False, test=False):

```

```
if test:

    class_image_path = f"images/test/{self.class_name}.png"

else:

    class_image_path = f"images/classes/{self.class_name}.png"

try:

    image = Image.open(class_image_path, "r")

except IOError:

    print(f"Cannot open image for class {self.class_name}")

    return

width, height = image.size

if width == height:

    self.rmax = int(width)

pixel_values = list(image.getdata())

if image.mode == "RGB":

    self.channels = 3

elif image.mode == "RGBA":

    self.channels = 3

elif image.mode == "L":

    self.channels = 1
```

```

else:

    print("Unknown mode: %s" % image.mode)

    return

if polar:

    self.class_image_data = self.convert_to_polar(pixel_values, width, height,
self.rmax, self.channels)

else:

    self.class_image_data = self.reshape_pixel_values(pixel_values, self.rmax,
self.rmax, self.channels)

@staticmethod

def generate_excel_chart(classes, m):

    # Create a DataFrame for the data

    df = pd.DataFrame({

        'dm_k': classes[m].dm_k,

        'Ek_Normal': classes[m].Ek_Normal,

        'D1': classes[m].D1,

        'D2': classes[m].D2,

        'a': classes[m].a,

        'b': classes[m].b

    })

    # Initialize Excel writer

```

```
excel_path = 'images/result/graphs/' + 'Нормоване_Ek_для_класу_' +  
classes[m].name + '.xlsx'
```

```
with pd.ExcelWriter(excel_path, engine='xlsxwriter') as writer:
```

```
df.to_excel(writer, sheet_name='Sheet1', index=False)
```

```
workbook = writer.book
```

```
worksheet = writer.sheets['Sheet1']
```

```
chart = workbook.add_chart({'type': 'line'})
```

```
chart.add_series({
```

```
    'name': 'Ek_Normal',
```

```
    'categories': ['Sheet1', 1, 0, len(classes[m].dm_k), 0],
```

```
    # (sheetname, first_row, first_col, last_row, last_col)
```

```
    'values': ['Sheet1', 1, 1, len(classes[m].dm_k), 1],
```

```
    'line': {'color': 'black'}
```

```
})
```

```
for i in range(len(classes[m].dm_k)):
```

```
    if classes[m].D1[i] > 0.5 and classes[m].a[i] < 0.5 and classes[m].D2[i] > 0.5 and  
    classes[m].b[
```

```
        i] < 0.5:
```

```
        worksheet.write_formula(i + 1, len(df.columns),
```

```
                                '=IF(AND(C{0}>0.5,E{0}<0.5,D{0}>0.5,F{0}<0.5), 1, "")'.format(i  
+ 2))
```



```

chart.add_series({
    'name': 'Condition',
    'categories': ['Sheet1', 1, 0, len(classes[m].dm_k), 0],
    'values': ['Sheet1', 1, len(df.columns), len(classes[m].dm_k),
len(df.columns)],
    'line': {'none': True},
    'marker': {'type': 'circle', 'size': 7, 'fill': {'color': 'red'}}
})

```

```

chart.set_title(
    {'name': 'Нормоване Ек для класу ' + classes[m].name + ' при delta(' +
str(classes[m].delta) + ')'})

```

```

chart.set_x_axis({'name': 'δ'})

```

```

chart.set_y_axis({'name': 'E'})

```

```

worksheet.insert_chart('D2', chart)

```

```

@staticmethod

```

```

def reshape_pixel_values(pixel_values, width, height, channels):

```

```

    reshaped_pixel_values = []

```

```

    for i in range(width):

```

```

        for j in range(height):

```

```

            pixel_index = i * width + j

```

```

            pixel = pixel_values[pixel_index]

```

```
    reshaped_pixel_values.append(pixel[0]) # Значення червоного кольору
    reshaped_pixel_values.append(pixel[1]) # Значення зеленого кольору
    reshaped_pixel_values.append(pixel[2]) # Значення синього кольору

image_data = np.array(reshaped_pixel_values).reshape((width, height * channels))
image_data = image_data.transpose()

return image_data
```

```
@staticmethod
```

```
def read_and_distribute_image(image_path, block_size, polar=False):
```

```
    # Read the image
```

```
    image = Image.open(image_path)
```

```
    height, width = image.size
```

```
    channels = 3 # RGB
```

```
    block_width = block_size
```

```
    block_height = block_size
```

```
    num_blocks = (width // block_width) * (height // block_height)
```

```
    blocks = []
```

```
    for i in range(num_blocks):
```

```
        x = (i % (width // block_width)) * block_width
```

```
y = (i // (width // block_width)) * block_height
```

```
block = image.crop((x, y, x + block_width, y + block_height))
```

```
blocks.append(block)
```

```
reshaped_blocks = []
```

```
for block in blocks:
```

```
    pixel_values = list(block.getdata())
```

```
    if polar:
```

```
        reshaped_blocks.append(ImageClass.convert_to_polar(pixel_values,  
block_width, block_height, block_size, channels))
```

```
    else:
```

```
        reshaped_blocks.append(ImageClass.reshape_pixel_values(pixel_values,  
block_width, block_height, channels))
```

```
return reshaped_blocks, image
```

```
@staticmethod
```

```
def show_exam_results(blocks_classes, blocks, block_size, classes_names,  
padded_img, M):
```

```
    mask = Image.new("RGBA", padded_img.size, (0, 0, 0, 0))
```

```
    draw = ImageDraw.Draw(mask)
```

```
font_size = 14
```

```

font = ImageFont.load_default(font_size)

num_classes = M

colors = [tuple(np.random.randint(0, 256, 3)) for _ in range(num_classes)]
colors.append((0, 0, 0))

for i, (block, block_class) in enumerate(zip(blocks, blocks_classes)):

    y = (i // (padded_img.width // block_size)) * block_size
    x = (i % (padded_img.width // block_size)) * block_size

    if block_class == -1:
        color = (0, 0, 0, 70)
    else:
        color = colors[block_class] + (100,)

    draw.rectangle((x, y, x + block_size, y + block_size), fill=color)

    text = str(block_class)

    text_width, text_height = font.getbbox(text)[2], font.getbbox(text)[3]
    text_position = (
        (x + (block_size - text_width) // 2),
        (y + (block_size - text_height) // 2),
    )

    draw.text(text_position, text, fill=(255, 255, 255, 255), font=font)

```

```
result = Image.alpha_composite(padded_img.convert("RGBA"), mask)

result.show()

result.save('images/result/' + 'result_' + datetime.now().strftime("%Y_%m_%d") +
'.png')
```

Main:

```
## Визначимо алфавіт класів розпізнавання і вирахуємо яскравість навчальної
матриці зображень
```

```
classes_names = ['tree', 'grass', 'field', 'road']
M = classes_names.__len__()
```

```
classes = []
rmax = 0
```

```
for class_name in classes_names:
    image = ImageClass(class_name)
    image.read_class_image()
    rmax = image.rmax
    i_class = ClassifierClass(y=image.class_image_data, name=class_name)

    classes.append(i_class)
```

```
## Визначимо і виконаємо обчислення значень  $u_m$ ,  $x$ ,  $x_m$ , кодових відстаней між  $x_m$ 
та  $x_j$ 
```

```
m = 0 # лічильник класів розпізнавання
```

```
while m < M:
```

```
    classes[m].set_r(rmax)
```

```
    classes[m].calc_mean_y_i() # обчислення середнього значення у
```

```
    classes[m].set_delta(19) # встановлення базового значення дельта
```

```
    classes[m].calc_x() # обчислення значення x
```

```

classes[m].calc_mean_x_i() # обчислення середнього значення x
classes[m].calc_dm_xm_xj() # обчислення значення dm_xm_xj

m = m + 1

## Розбиття множини еталонних векторів на пари найближчих "сусідів"

R = []

dm_M = [[0] * M for _ in range(M)]

for i in range(M):
    for j in range(M):
        if i != j:
            dm_M[i][j] = sum(classes[i].mean_x_i[k] ^ classes[j].mean_x_i[k] for k in
range(classes[j].mean_x_i.__len__()))
        else:
            dm_M[i][j] = classes[j].mean_x_i.__len__() + 1

    R.append((i, (dm_M[i].index(min(dm_M[i])))))

## Визначимо і виконаємо базовий алгоритм, визначимо кодову відстань xm та xc(j)

m = 0 # лічильник класів розпізнавання

h = 1 # крок збільшення радіусу класу

while m < M:
    k = 0 # лічильник класів розпізнавання

    # обчислення значення d(xm ^ xc(j))

```

```
classes[m].calc_dm_xm_xcj(classes[R[m]][1]).x)
```

```
while True:
```

```
    classes[m].insert_dm_k(k, k + h)
```

```
    classes[m].calc_K(k)
```

```
    classes[m].calc_workspace(k)
```

```
    classes[m].calc_Ek(k)
```

```
    classes[m].calc_Emax()
```

```
    classes[m].calc_Ek_Normal(k) # обчислення нормалізованого значення Ek
```

```
    k = k + 1
```

```
    if k >= 100:
```

```
        break
```

```
m = m + 1
```