

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

_____ Оксана ШОВКОПЛЯС
(підпис)

12 грудня 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня магістр

зі спеціальності 122 «Комп'ютерні науки»

освітньо-професійної програми «Інформатика»

на тему: Інформаційна технологія машинного навчання автономного БПЛА для семантичної сегментації цифрового зображення регіону
здобувача групи ІН.м-33 Дзернюка Вадима Юрійовича

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

_____ Вадим ДЗЕРНЮК
(підпис)

Здобувач групи ІН.м-33

(підпис)

Вадим ДЗЕРНЮК

Доктор філософії, асистент

кафедри комп'ютерних наук

(підпис)

Микита МИРОНЕНКО

Суми - 2024

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
Оксана ШОВКОПЛЯС
(підпис)

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»
здобувача групи ІН.м-33 Дзернюка Вадима Юрійовича

1. Тема роботи: «Інформаційна технологія машинного навчання автономного БПЛА для семантичної сегментації цифрового зображення регіону»
затверджую наказом по СумДУ від «03» грудня 2024 р. №1257-VI

2. Термін задачі здобувачем кваліфікаційної роботи до 8 грудня 2024 року

3. Вхідні дані до кваліфікаційної роботи

Цифрові зображення регіону спостереження, отримані за результатами аерофотозйомки

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1) Аналіз проблеми предметної області, постановка й формування завдань дослідження.

2) Огляд технологій, що використовуються для машинного навчання автономного БПЛА для семантичної сегментації цифрового зображення регіону.

3) Розробка інтелектуальної системи для семантичної сегментації цифрового зображення регіону.

4) Аналіз результатів

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Слайди:

1) Титульний аркуш.

2) Актуальність роботи.

3) Мета та задачі її досягнення.

4) Вхідні дані

5) Опис методу машинного навчання

6) Структурна схема алгоритму машинного навчання.

7) Результати комп'ютерного моделювання.

8) Висновки.

6. Консультанти до проекту (роботи), із зазначенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання « ____ » _____ 20 ____ р.

Завдання прийняв до виконання _____ Керівник _____
 (підпис) (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми предметної області, постановка формалізованої задачі, формування завдань дослідження</i>	30.09.2024	
2	<i>Огляд технологій, що використовуються для машинного навчання автономного БПЛА для семантичної сегментації цифрового зображення регіону</i>	14.10.2024	
3	<i>Розробка інтелектуальної системи для семантичної сегментації цифрового зображення регіону</i>	01.11.2024	
4	<i>Аналіз результатів</i>	05.11.2024	
5	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>	20.11.2024	

Здобувач вищої освіти _____ Керівник _____
 (підпис) (підпис)

АНОТАЦІЯ

Записка: 61 стор., 7 рис., 4 додатка, 22 джерела.

Обґрунтування актуальності теми роботи – Тема кваліфікаційної роботи є актуальною, оскільки присвячена вирішенню важливого практичного завдання автоматичної класифікації об'єктів на цифрових зображеннях, отриманих за допомогою БПЛА, з використанням машинного навчання у рамках інформаційно-екстремальної інтелектуальної технології аналізу даних. Це сприяє покращенню точності розпізнавання об'єктів за умов обмежених ресурсів.

Мета роботи – підвищення точності машинного навчання автономного БПЛА для семантичної сегментації цифрового зображення регіону спостереження.

Об'єкт дослідження – процес машинного навчання автономного БПЛА для семантичної сегментації цифрового зображення регіону, отриманого за результатами аерофотозйомки.

Предмет дослідження – критерії оптимізації параметрів машинного навчання та алгоритм машинного навчання автономного БПЛА для семантичної сегментації зображення регіону спостереження.

Результати – розроблено інформаційну технологію та її засоби для машинного навчання автономного БПЛА для ідентифікації природних та інфраструктурних об'єктів на цифрових зображеннях регіону спостереження. За результатами функціонального тестування підтверджено високу точність машинного навчання.

ІНФОРМАЦІЙНА СИСТЕМА, МАШИННЕ НАВЧАННЯ, МОВА
ПРОГРАМУВАННЯ PYTHON, БПЛА, СЕГМЕНТАЦІЯ ЦИФРОВОГО
ЗОБРАЖЕННЯ.

ЗМІСТ

ВСТУП	7
1 АНАЛІЗ ПРОБЛЕМИ	8
1.1 Сучасний стан і перспективи розвитку та застосування автономних БПЛА	8
1.2 Аналіз сучасних інтелектуальних інформаційних технологій аналізу даних	11
1.3 Формалізована постановка задачі дослідження	15
2 ОПИС МЕТОДУ ДОСЛІДЖЕННЯ	19
2.1 Основні положення та принципи інформаційно-екстремальної інтелектуальної технології аналізу даних	19
2.2 Інформаційні критерії оптимізації параметрів машинного навчання	21
2.3 Опис базового алгоритму інформаційно-екстремального машинного навчання	24
2.4 Опис алгоритму функціонального тестування	26
3 ІНФОРМАЦІЙНЕ АЛГОРИТМІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ АУТОНОМНОГО БПЛА	29
3.1 Формування вхідної навчальної матриці	29
3.2 Алгоритм машинного навчання з оптимізацією параметрів функціонування системи розпізнавання	30
3.3 Побудова бінарного дерева даних	37
3.4 Алгоритм ієрархічного машинного навчання	38
3.5 Короткий опис програми	40
3.6 Результати комп'ютерного моделювання	42

ВИСНОВКИ	45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	47
ДОДАТОК А	50
ДОДАТОК Б	53
ДОДАТОК В	58
ДОДАТОК Г	60

ВСТУП

Актуальність. Сучасний розвиток технологій, зокрема штучного інтелекту та автономних безпілотних літальних апаратів (БПЛА), відкриває нові перспективи для вирішення складних завдань у таких галузях, як сільське господарство, екологічний захист, логістика та оборона. Одним із ключових напрямів розвитку є здатність автономного БПЛА самостійно аналізувати та обробляти дані, включно з візуальною інформацією, що значно покращує ефективність їх використання.

Об'єкт дослідження. Процес машинного навчання автономного БПЛА для семантичної сегментації цифрового зображення регіону, отриманого за результатами аерофотозйомки.

Предмет дослідження. Функціональні категорійні моделі та алгоритми інформаційно-екстремального машинного навчання та функціонального тестування.

Новизна. Розроблено метод інформаційно-екстремального машинного навчання автономного БПЛА за технологією XGBoost з використанням бінарного дерева рішень, що дозволяє підвищити точність розпізнавання об'єктів за обмежених обчислювальних ресурсів.

Структура. Дана робота складається зі вступу, аналітичного огляду, постановки задачі, обґрунтування вибору методу розв'язання поставленої задачі, опису програмного забезпечення інформаційної системи, висновків, списку використаних джерел та додатків.

1 АНАЛІЗ ПРОБЛЕМИ

1.1 Сучасний стан і перспективи розвитку та застосування автономних БПЛА

У наш час безпілотні літальні апарати швидко стають необхідними в різних сферах діяльності завдяки можливостям працювати у важкодоступних або небезпечних областях. Використання БПЛА поширене в таких сферах, як: сільське господарство, екологічний моніторинг, оборона та логістика або навіть у наданні медичних послуг в дальніх районах [1].

У сільському господарстві безпілотні літальні апарати використовуються для моніторингу стану посівів, визначення рівня вологості ґрунту та оцінки зрілості урожаю [2]. Вони також допомагають виявляти шкідливих комах та хвороб на ранньому етапі. Це допомагає фермерам оперативно вирішувати різні проблеми на полях, що значно покращує урожайність. Також безпілотники дозволяють створити детальні карти полів із точними даними, що гарантує ефективне управління аграрною діяльністю.

У галузі екологічного моніторингу безпілотні літальні апарати допомагають вивчати стан навколишнього середовища. Це дозволяє відслідковувати зміни у екосистемах, виявляти забруднення водних та повітряних ресурсів, а також проводити моніторинг лісових масивів для запобігання лісовим пожежам. Завдяки можливостям швидко отримувати актуальну інформацію, безпілотні літальні апарати є ефективними інструментами для збереження природного світу та контролю за ресурсами [3]. Наприклад, виявлення незаконного рубання дерев або спостереження за сміттям у прибережних водах можна проводити у реальному часі, що сприяє швидкому реагуванню на проблеми.

У військовій сфері безпілотні літальні апарати використовуються для здійснення розвідки та нагляду. Вони можуть працювати самостійно у небезпечних умовах, проводити розвідувальні операції без покладання ризику на

життя військових. Також вони служать для ведення обліку переміщень підозрілих об'єктів чи техніки, й також підтримки в ході бойових дій. БПЛА дозволяють оперативно отримувати інформацію про потенційні загрози тощо.

У сфері логістики безпілотні літальні апарати можуть бути використані для швидкої доставки товарів, особливо у випадках, коли звичайні транспортні засоби є неефективними або недоступними. Наприклад, у віддалених або складних регіонах. Використання безпілотників для логістичних завдань може значно зменшити час доставки та скоротити транспортні витрати, сприяючи розвитку “останньою милею” доставки швидкого покращення структури мегаполісу. Наприклад, для екстрених ситуацій, таких як природні катастрофи, безпілотні літальні апарати можуть поставляти необхідні медикаменти, речі чи харчові продукти.

Слід приділити особливу увагу можливостям розвитку безпілотних літальних апаратів із використанням штучного інтелекту. Останні досягнення в галузі штучного інтелекту відкривають нові горизонти для розвитку безпілотників [4]. ШІ дозволяє літальним апаратам самостійно аналізувати дані, адаптуватися до змін у середовищі та приймати автономні рішення в реальному часі. Це значно розширює можливості безпілотників і робить їх необхідними для швидкого реагування на непередбачені обставини.

За допомогою ШІ безпілотні літальні апарати можуть виконувати такі складні завдання:

1. Завдяки використанню алгоритмів і технологій комп'ютерного зору та машинного навчання безпілотники можуть автоматично уникати перешкод, що підвищує безпеку і надійність їх функціонування.

2. БПЛА зі штучним інтелектом здатні розрізняти об'єкти на основі аерофотозйомки й виділяти важливі елементи – такі як дерева, дороги, будинки, транспортні засоби. Це дозволяє використовувати їх для картографування й моніторингу [5].

3. Завдяки машинному навчанню безпілотні літальні апарати можуть адаптувати свої алгоритми до змін умов оточуючого середовища (наприклад, освітлення та погодних умов), що забезпечує стабільну роботу в різних умовах.

Ці можливості відкривають нові перспективи використання безпілотних літальних апаратів у складних умовах, де необхідна автономність, точність та гнучкість у процесі прийняття рішень.

Розвиток безпілотних літальних апаратів супроводжується з безліччю викликів, пов'язаних з обробкою великих обсягів інформаційних даних, підвищенням точності класифікації об'єктів і забезпеченням стабільності алгоритмів машинного навчання.

Безпілотні літальні апарати, що використовуються для збору інформації, генерують значну кількість зображень, котрі потребують швидкої обробки та аналізу [6]. Для того щоб забезпечити самостійне функціонування БПЛА, необхідний ефективний алгоритм для обробки даних в режимі реального часу, який дозволить проводити аналіз без затримок.

Програмне забезпечення повинно надійно класифікувати об'єкти. Автономним безпілотним літальним апаратам потрібна здатність точно ідентифікувати об'єкти – це завдання складне у різноманітних або мінливих умовах. Наприклад, при польотах над густонаселеними районами або складними міськими ландшафтами алгоритм повинен вміти дуже точно розрізняти об'єкти один від одного з великою точністю. Вимоги до точності ще важливіші, оскільки неточна класифікація може призвести до збою та спричинити серйозну шкоду військовим або у державних рятувальних операціях.

Варто зазначити надійність алгоритмів машинного навчання в динамічних умовах. Безпілотні літальні апарати, що працюють в реальних умовах, піддаються впливу непередбачуваних змін у навколишньому середовищі, таким як зміна освітлення та погодних умов, а також поява нових перешкод. Алгоритми машинного навчання повинні стабільно працювати у таких обставинах, адаптуватися до нових ситуацій і підтримувати високу точність класифікації.

Тому, розвиток безпілотних літальних апаратів потребуватиме створення нових технологій обробки даних з інформаційно-екстремальною спрямованістю. Це забезпечує високу продуктивність обробки даних і надійну роботу в реальних умовах. Вирішення цих завдань може істотно підвищити автономність та функціональність безпілотників, сприяючи їх широкому застосуванню в різних галузях, де важливий швидкий і точний аналіз інформації.

1.2 Аналіз сучасних інтелектуальних інформаційних технологій аналізу даних

У сучасному світі обсяг даних постійно зростає, тому інтелектуальні інформаційні технології стають все більш важливими. Системи, засновані на штучному інтелекті та машинному навчанні, можуть аналізувати цю інформацію, виявляти приховані закономірності та робити дуже точні прогнози [7]. Серед основних методик обробки даних можна виділити машинне навчання, глибоке навчання, а також алгоритми класифікації та кластеризації. Кожна з цих методик має свої переваги та обмеження.

Машинне навчання стало ключовим інструментом для аналізу даних завдяки його здатності вчитися на прикладах та покращувати свою ефективність без чітких інструкцій [8]. Це означає, що алгоритми машинного навчання можуть самостійно аналізувати дані, виявляти закономірності та робити висновки, просто шляхом вивчення попередньо наданих інформаційних даних. Машинне навчання застосовується у сферах розпізнавання образів, текстового аналізу, прогнозування на ринку та в багатьох інших контекстах. Наприклад, в області аналізу зображень, моделі можуть бути розроблені для виявлення різних об'єктів на аерофотознімках, таких як лісові масиви, дороги або поля. Це надзвичайно важливо для операцій безперервного польоту без використання екрану.

Глибоке навчання (DL) – це метод машинного навчання, який використовує багатошарові нейронні мережі для вивчення складних та багатокомпонентних залежностей [9]. Глибокі нейронні мережі, такі як згорткові

нейронні мережі (CNN), можуть автоматично відокремлювати важливі функції від зображень великого обсягу, що робить їх ідеальними для завдань комп'ютерного зору та обробки природної мови. Наприклад, CNN може бути навчений для розпізнавання конкретних візуальних патернів на аерофотознімках БПЛА. Глибоке навчання – це дуже потужний метод, проте він вимагає великої кількості даних та значних обчислювальних ресурсів. Це може стати перешкодою для автономних систем, яким потрібно обробляти дані в режимі реального часу.

Алгоритми класифікації дозволяють автоматично розподіляти об'єкти за певними категоріями [10]. Класифікація є важливим завданням у багатьох додатках, де нам потрібно визначити тип або категорію об'єкта. Наприклад, алгоритм класифікації може бути застосований для розпізнавання типу ландшафту на зображенні і виділення кадрів, таких як лісу, поля або дороги. В даний час існує багато алгоритмів класифікації, таких як логістична регресія, метод опорних векторів, дерева прийняття рішень та ансамблеві методи, такі як випадковий ліс і градієнтний бустинг. Крім того, алгоритми мають власну ефективність та точність, що дозволяє вибрати оптимальний підхід в залежності від конкретного завдання.

Алгоритми кластеризації дозволяють об'єднувати схожі об'єкти разом без попередньо визначених міток [10]. Це може бути корисним для вивчення структури даних, де категорії можуть бути невідомі заздалегідь. Наприклад, для дослідження рослинності або ландшафтів на зображеннях місцевості, для виявлення нових зразків або виняткових моментів.

Варто відзначити, що серед сучасних алгоритмів машинного навчання особливою популярністю користуються XGBoost і дерева рішень. Ці методи часто використовуються для класифікації та аналізу даних.

XGBoost (Extreme Gradient Boosting) [11] є одним з найпотужніших та найефективніших алгоритмів для задач класифікації та регресії. Це вдосконалена версія методу градієнтного бустингу, яка використовує кілька дерев рішень

разом, кожне з яких виправляє помилки з попереднього дерева рішень. Це значно підвищує точність прогнозування, зменшує кількість помилок і дозволяє уникнути проблем з перенавчання, котрі часто виникають з простими моделями. XGBoost оптимізований для обробки великих обсягів даних завдяки своїм можливостям паралельних обчислень. Це робить його швидким і високоефективним, особливо в критичні моменти часу, наприклад у випадку автономних безпілотників.

Дерева рішень [12] – це прості та інтуїтивно зрозумілі алгоритми моделювання. Вони є у вигляді дерева з гілками для різних варіантів на основі значень ознак. Цей метод чудово підходить для завдань класифікації і може служити зрозумілою моделлю результатів класифікаційно-аналітичного процесу. Наприклад, за допомогою дерева прийняття рішень ми можемо створити шляхи вибору та пояснення до класифікаційних результатів. Наприклад, метаданих фотографій як ландшафт або поле культурного застосування ґрунту на основі конкретних ознак фотографованого об'єкта.

Умови роботи автономних БПЛА вимагають високої точності і швидкого та стабільного аналізу даних для досягнення гарних результатів. Вибір методу обробки даних суттєво залежить від специфіки завдання БПЛА та оточення, у якому він працюватиме.

Визначимо основні критерії при виборі процесу аналітики даних для безпеки польоту літального апарату:

1. Точність і надійність. Глибоке навчання та нейронні мережі забезпечують високу точність класифікації через їх здатність розділяти складні характеристики вхідних даних. Однак ці моделі мають ускладнення при перенавчанні через появу нових класів розпізнавання, час машинного навчання суттєво залежить від кількості ознак розпізнавання і вимагають великої кількості позитивних прикладів для ефективного навчання. Наприклад, XGBoost та дерева рішень не настільки ресурсомісткі, як глибоке навчання, але можуть забезпечити стабільне прогнозування з меншими обсягами даних. Це особливо

корисно для завдань, де важлива стабільність або обмежений доступ до потужних обчислювальних ресурсів.

2. Обробка даних в реальному часі. Завдяки своїй оптимізованій структурі, XGBoost може швидко обробляти дані й ідеально підходить для використання з БПЛА, де швидкість прийняття рішень має вирішальне значення. Нейронні мережі можуть бути точними, але повільними, і це обмеження, особливо в середовищах, де важлива швидкість.

3. Ясність висновків. У більшості випадків важливо знати принципи алгоритмічного прийняття рішень, особливо коли мова йде про важливі або важливі завдання. Дерево рішень є досить чіткою моделлю, оскільки воно може відстежувати кожен крок процесу прийняття рішень. Це особливо важливо в тих областях, де необхідно точно і чітко прогнозувати поведінку алгоритму, наприклад, в операціях відновлення.

Таким чином, сучасні методи обробки даних надають широкі можливості для виконання складних завдань класифікації та аналізу зображень – це основа для розвитку автономних систем. Вибір методу залежить від ряду факторів: точності результатів, швидкодії обчислень, обмежень використання обчислювальних ресурсів і вимог до ясності отриманих результатів. Кожен метод має свої унікальні переваги і може бути адаптований до конкретних умов експлуатації безпілотних літальних апаратів, забезпечуючи надійні результати і ефективність в реальних умовах.

Безпілотні літальні апарати (БПЛА) стають все більш популярними в сучасному світі для виконання різних завдань. Деякі використовуються для моніторингу місцевості, пошуку і порятунку, а також для сільськогосподарського контролю. Однак для успішного виконання цих завдань БПЛА повинен мати можливість швидко та точно аналізувати зображення, отриманих під час польоту. Тому розробка алгоритмів автоматичної обробки візуальних даних і класифікація типів земель має вирішальне значення. Такі вимоги є актуальними у випадку використання автономних БПЛА для

семантичної сегментації цифрових зображень у зоні спостереження з метою розділення та класифікації природних та інфраструктурних об'єктів на землі.

Основними недоліками розглянутих методів для машинного навчання автономного БПЛА для семантичної сегментації цифрового зображення регіону спостереження є:

1. Чутливість до багатовимірності простору ознак розпізнавання та потужності абетки класів розпізнавання. Це призводить до збільшення часу машинного навчання і зниження його точності. Наприклад, у згорткових нейромережах час машинного навчання збільшується за експоненціальним законом.

2. Відсутність гнучкості при перепідготовці БПЛА через збільшення числа класів розпізнавання. Наприклад, нейроподібні структури та методи дерева рішень вимагають побудови нових структур.

3. Розглянуті методи машинного навчання принципово не дозволяють побудувати безпомилкові за навчальною матрицею вирішувальні правила, оскільки не враховують той факт, що у загальному випадку розбиття класів розпізнавання в просторі ознак є нечітким.

4. Забезпечення прийнятної точності вимагає великої кількості зразків зображень для машинного навчання, щоб побудувати навчальну матрицю.

Усі ці недоліки усуваються методами машинного навчання, розробленими в рамках функціонального підходу до моделювання когнітивних процесів природного інтелекту, з використанням інформаційно-екстремальної оптимізації параметрів, в основу яких покладено максимізацію інформаційних спроможностей системи розпізнавання в процесі машинного навчання [13].

1.3 Формалізована постановка задачі дослідження

Мета дослідження полягає у створенні алгоритму для автоматичної класифікації ландшафту на основі зображень від безпілотних літальних апаратів (БПЛА), використовуючи інформаційно-екстремальні критерії та оптимізацію

параметрів машинного навчання [13]. Використовуючи критерії перевантаження інформацією, ми можемо підвищити точність класифікації та зменшити ризик помилок, встановивши оптимальні параметри для розпізнавання різних об'єктів на зображенні. Наприклад, ви можете покращити чіткість між класами (наприклад, ліси та поля) та декомутувати класифікацію, оптимізувавши параметри яскравості та контрасту.

Розроблений алгоритм повинен працювати в режимі реального часу для обробки значних обсягів даних під час польоту безпілотного літального апарату. Це означає, що алгоритм має бути ефективним та мати низькі вимоги до обчислювальних ресурсів у зв'язку з обмеженими можливостями обчислювальної потужності на борту БПЛА. Таким чином, основною метою цієї роботи є не тільки створення класифікаційного алгоритму, а й оптимізація його параметрів для досягнення швидкості, точності та ефективності в умовах обмежень.

Для досягнення мети дослідження потрібно виконати кілька завдань, кожне з яких спрямоване на певний аспект розробки алгоритму та його оптимізації [14]. Розглянемо ці завдання детальніше:

1. Створення навчальної матриці є важливим кроком у вивченні моделі машинного навчання. Ця матриця містить властивості зображення (такі як значення кольору RGB, яскравість та текстури), що служать орієнтирами для моделі, а також мітки класів (ліси, поля та дороги). Вибір правильних характеристик для кожного класу ландшафту, попередня обробка зображення для видалення шуму й поліпшення якості даних – основне завдання при створенні такої матриці. Тому, до цього процесу слід підходити уважно. Побудова значущої навчальної матриці є важливим кроком, який має вирішальний вплив на точність алгоритму розпізнавання різних типів місцевості.

2. Налаштування та перевірка критеріїв класифікації. Щоб класифікувати місцевості, необхідно уточнити критерії точності, що визначають, до якого класу належить кожна окремий кадр зображення. Критерій Кульбака та інші

інформаційно-екстремальні критерії дозволяють оцінити якість розпізнавання шляхом вимірювання інформаційної різниці між класами [15]. Це підтверджує, що алгоритм налаштований таким чином, щоб максимізувати розпізнавальну здатність між класами, наприклад, навіть в складних ситуаціях, де є відокремлення лісу від поля або дороги.

3. Ієрархічне машинне навчання. У випадку, коли класів для класифікації дуже багато або вони можуть бути розподілені на підкласи, ієрархічне машинне навчання дозволяє структурувати процес розпізнавання як багаторівневе дерево. Наприклад, алгоритм може спочатку визначити, чи це природний об'єкт чи штучний об'єкт, і якщо це природний об'єкт, то чи це ліс чи поле. Використання ієрархічного підходу полегшує точну та ефективну класифікацію, оскільки алгоритм акцентується на конкретному завданні класифікації на кожному рівні. Це особливо корисно у випадках, коли різні типи місцевості мають схожі особливості. Груба класифікація на ранніх етапах допомагає алгоритму робити більш точні прогнози на подальших етапах.

4. Налагодження параметрів допусків. Під час роботи алгоритму дуже важливо контролювати допуски – це значення, що показують ступінь впевненості у процесі прийняття рішень [16]. Наприклад, якщо яскравість або контрастність зображення змінюються через зміни умов освітлення, алгоритм повинен бути готовий адаптувати свою роботу для забезпечення стабільності в класифікаційному процесі. Налаштування параметрів допусків сприяє уникненню помилкових класифікацій та допомагає подолати ситуації, коли дані є неоднозначні або нечіткими. Це означає, що ефективне налаштування алгоритму для правильного розпізнавання об'єктів можливе за наявності перешкод чи недосконалих умов у зображенні.

5. Оцінка та аналіз роботи алгоритму. Завершальним етапом є оцінка ефективності алгоритму та аналіз отриманих результатів. Це включає перевірку точності класифікації на різних типах місцевості, а також оцінку стабільності алгоритму в умовах змінної якості зображень. Результати аналізу дозволяють

зробити висновки про ефективність роботи алгоритму в завданнях класифікації та необхідність подальшого вдосконалення.

Отже, сформульоване завдання дослідження включає в себе всі важливі етапи розробки і вдосконалення алгоритму автоматичної класифікації природних об'єктів. Кожен із цих кроків є важливим для досягнення основної мети – створення точного та ефективного інструменту, здатного до класифікації зображень у реальному часі за обмежених ресурсів. Це дослідження має значний вплив на розвиток автономних БПЛА, надаючи їм новий потенціал для точного аналізу природи та ефективного прийняття рішень на основі візуальних даних.

2 ОПИС МЕТОДУ ДОСЛІДЖЕННЯ

2.1 Основні положення та принципи інформаційно-екстремальної інтелектуальної технології аналізу даних

Сучасні підходи до аналізу даних постійно розвиваються, і одним з нових та перспективних напрямків є інформаційно-екстремальна інтелектуальна технологія аналізу даних [13]. На відміну від сучасних методів машинного навчання з переваженням даних, методи інформаційно-екстремального розробляються в рамках функціонального підходу до моделювання когнітивних процесів природного інтелекту при формуванні та прийнятті класифікаційних рішень. Ця технологія заснована на принципах теорії інформації і використовує концепцію максимальної інформаційної цінності моделі, щоб зробити процес машинного навчання моделей більш ефективним і точним. Основна концепція полягає в тому, що для досягнення найкращих результатів навчання нам потрібно зосередитись на отриманні найбільш корисної інформації з кожного фрагмента даних. Завдяки цьому, модель може швидше вчитися та досягати високої точності, використовуючи менше ресурсів.

Інформаційно-екстремальна технологія була створена для оптимізації процесу машинного навчання шляхом застосування інформаційних критеріїв. Традиційні методи машинного навчання спрямовані на зменшенні похибки. Модель налаштовується таким чином, щоб мінімізувати різницю між прогнозованими та фактичними результатами. Однак інформаційно-екстремальна технологія підходить до цієї проблеми з іншого боку: замість простої мінімізації похибки вона зосереджується на максимізації корисної інформації з кожного нового екземпляра.

Цей підхід ґрунтується на критеріях максимальної інформаційної цінності для оцінки того, у якому ступені нова інформація, що потрапляє в систему, може покращити здатність моделі розрізняти класи. Наприклад, у завданні класифікації ландшафту (лісу, поля, дороги) інформаційно-екстремальна

технологія буде збирати таку інформацію, яка дозволить моделі найточніше відрізнити один клас від іншого. Це означає, що дані з найвищою інформаційною цінністю для розпізнавання класів будуть мати пріоритет у процесі навчання.

Цей підхід не тільки робить модель більш точною, але й дозволяє ефективніше використовувати час та ресурси для навчання, оскільки знання максимально розширюються з кожним новим набором даних. Це особливо важливо для безпілотних літальних апаратів (БПЛА), які мають обмежену обчислювальну потужність і вимагають швидкої обробки даних у реальному часі.

Оптимізація параметрів моделі є важливим аспектом інформаційно-екстремальної технології, котра дозволяє налаштувати модель так, щоб вона досягала максимальної точності, мінімізуючи помилки при класифікації. У випадку з класифікацією об'єктів на зображеннях від БПЛА, оптимізація параметрів може включати налаштування яскравості, контрасту, фільтрів для виділення об'єктів або інших параметрів, що впливають на сприйняття моделі.

Принцип оптимізації параметрів заснований на ідеї контрольних меж, котрі встановлюють параметри моделі розпізнавання зображень у певному допустимому діапазоні, щоб поліпшити здатність моделі точно розрізнити класифікаційні групи. Наприклад, інтенсивність світла може бути змінена в заданих межах, щоб забезпечити найкращі умови для точного розрізнення класів. Оптимізуючи параметри, модель адаптується до різних умов, таких як освітлення, що може впливати на якість зображень.

Завдяки оптимізації параметрів інформаційно-екстремальна технологія дозволяє моделі досягати високої точності класифікації навіть у складних умовах. Це особливо корисно для безпілотних літальних апаратів (БПЛА), які працюють в реальних умовах і вимагають здатності точно розпізнавати об'єкти незалежно від освітлення або перешкод.

Отже, інформаційно-екстремальна технологія є привабливим методом для навчання моделей з більш високою точністю класифікації та оптимізованим

використанням обчислювальних ресурсів. Це робить її дуже перспективною для використання в автономних системах, таких як БПЛА. У таких системах кожен фрагмент чи кадр інформації повинен бути максимально корисним, а процес навчання – ефективним і адаптивним до різних умов.

2.2 Інформаційні критерії оптимізації параметрів машинного навчання

Машинне навчання з використанням інформаційно-екстремальних методів використовує певні інформаційні критерії для підвищення якості класифікації даних. Ці критерії визначають, наскільки добре модель може розподілити класи, і визначають оптимальні параметри для отримання найбільш точних результатів. Одним з найбільш поширених критеріїв для цих потреб є критерій Кульбака-Лейблера, який оцінює неоднорідність між класами. Давайте розглянемо, як працює цей критерій та як оптимізація параметрів (таких як параметри δ і радіус контейнерів) може покращити точність нашої моделі.

Критерій Кульбака-Лейблера (KL divergence або KL-дивергенція) [17] є одним із ключових інструментів для вимірювання розбіжності між двома ймовірнісними розподілами. У нашому випадку цей критерій застосовується для оцінки того, наскільки розрізняються розподіли характеристик для різних класів (наприклад, лісу та поля) у зображеннях під час класифікації.

Формула KL-дивергенції виглядає так [17]:

$$D_{KL}(P \parallel Q) = \sum_i P(i) \cdot \log \frac{Q(i)}{P(i)} \quad (2.1)$$

де $P(i)$ – це ймовірність того, що об'єкт належить до класу i згідно з реальним розподілом;

$Q(i)$ – ймовірність того, що об'єкт належить до класу i згідно з розподілом, який передбачає модель.

Якщо враховувати формулу та додати оцінку різниці між двома розподілами D_{KL} , то чим більше це значення – тим більше розрізняються розподіли, що полегшує класифікацію класів. У разі класифікації зображень за допомогою безпілотного літального апарата (БПЛА) велика KL-дивергенція між розподілами ознак для різних класів свідчить про чітке розділення класів, що дає змогу моделі ефективно їх розпізнавати.

Критерій Кульбака використовується для вибору найбільш інформативних параметрів і ознак у моделі машинного навчання. Наприклад, якщо певні ознаки (такі як яскравість чи колір) добре розрізняють класи, модель буде налаштована так, щоб ці ознаки мали пріоритет у процесі класифікації, що значно покращує точність прогнозування.

Крім того, важливо врахувати налаштування параметра δ (дельта) і радіуса контейнера, щоб забезпечити максимальну точність в розрізненні класів об'єктів. Щоб досягти більшої ефективності в класифікаційних завданнях, нам необхідно не тільки використовувати критерій Кульбака, але й зосередитися на оптимізації параметрів моделей, таких як δ та розмір контейнерів, окремо для кожного класу. Оптимізація цих параметрів також допомагає встановити найкращі межі для розрізнення класів, підвищуючи точність та стабільність алгоритму.

Параметр δ визначається для встановлення допустимого значення ознаки, за яким модель приймає рішення про класифікацію об'єкта до певного класу. Це означає, наскільки може бути зміщення від середніх значень ознак класу, щоб об'єкт залишався в цьому класі.

Наприклад, якщо зображується ліс, можна очікувати певних відтінків зеленого кольору та текстур. Якщо значення δ надто маленьке, невелике відхилення в кольорі чи текстурі листя може призвести до помилкового розподілу цього об'єкта до іншого класу. З іншого боку, при надто великому значенні δ модель може сприймати як схожими об'єкти, які насправді належать до різних класів.

Оптимізація параметра δ виконується таким чином, щоб модель могла точно розрізняти класи та залишатися стійкою до незначних змін у даних. Щоб визначити оптимальне значення δ , аналізують KL-дивергенцію для різних значень параметра та обирають те значення, при якому розділення між класами є найефективнішим.

Також важливо враховувати радіус контейнерів. Радіус контейнера визначає область навколо середніх значень ознак кожного класу [18]. Ця область вказує на те, до якого класу належать всі об'єкти, що потрапляють у межі контейнера. Наприклад, для класу “ліс” можна визначити середні значення ознак (таких як зелений колір і текстура) і встановити радіус, який вказуватиме, наскільки далеко можуть відхилитися інші значення від цього центру, щоб об'єкт все ще належав до класу “ліс”.

Якщо радіус контейнера буде занадто малий, може статися так, що модель виключатиме об'єкти з цього класу через невеликі відхилення в характеристиках. При надто великому радіусі контейнера модель може почати включати до класу об'єкти, які насправді належать до інших класів. Тому дуже важливо знаходити оптимальний радіус для кожного класу, щоб досягти максимальної точності класифікації.

Оптимальний радіус контейнерів можна визначити шляхом побудови графіка залежності критерію Кульбака від радіусу для значень у межах від 10 до 100 пікселів. Для кожного радіуса обчислюють KL-дивергенцію між класами та представляють результат у графічній формі для знаходження точки максимального розділення між класами, після чого обирають цей радіус як оптимальний.

Таким чином, використання критерію Кульбака дозволяє оцінити чіткість розрізнення класів на основі моделі. Налаштування параметра δ і радіуса контейнерів оптимізує модель, щоб вона могла найкраще розрізняти класи, навіть у складних умовах. Це призводить до високоточних визначень про класифікацію і підвищує стабільність моделі в різних умовах, що, в свою чергу,

дозволяє їй краще працювати в режимі реального часу на БПЛА, навіть при різних умовах освітлення і напрямках зйомки.

2.3 Опис базового алгоритму інформаційно-екстремального машинного навчання

Інформаційно-екстремальне машинне навчання використовує чіткий алгоритм класифікації, оптимізований на основі інформаційних критеріїв, для отримання високоточних результатів при розподілі класифікаційних даних, таких як XGBoost.

XGBoost (Екстремальне градієнтне підсилення) – один з популярних методів градієнтного бустингу, який об'єднує кілька дерев рішень для підвищення точності прогнозування. XGBoost особливо ефективний для завдань класифікації в умовах складних структур даних і необхідності в гнучких моделях. Якщо дані мають складну структуру і потребують гнучкості, XGBoost може бути найкращим вибором. Однією з причин вибору цього алгоритму є те, що обробляти великі обсяги даних легко та має стабільність у завданнях класифікації.

Алгоритм XGBoost працює, створюючи серію послідовних моделей (дерев рішень), в яких кожне нове дерево спрямовується на виправлення помилок з попередніх дерев. На кожному етапі алгоритм намагається мінімізувати функцію втрат L для покращення точності передбачень. Загальний вигляд функції [11]:

$$L = \sum_{i=1}^N l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \quad (2.2)$$

де y_i – це справжня мітка класу для об'єкта i ;

$\hat{y}_i^{(t-1)}$ – прогноз, зроблений на попередньому кроці;

$f_t(x_i)$ – поточне дерево рішень, що коригує попередній прогноз;

$\Omega(f_t)$ – регуляризаційний член, який запобігає перенавчанню моделі.

Для використання XGBoost у класифікації зображень кожен об'єкт (кадр зображення) представляється певними числовими атрибутами, які описують колірні особливості (наприклад, середні значення червоного, зеленого та синього (RGB) каналів). Для кожного кадру обчислюється середнє значення RGB як вхідний атрибут для моделі, що дозволяє розпізнавати колірну специфіку, характерну для конкретного класу (лісу, поля чи дороги).

Роль XGBoost у цьому процесі полягає у пошуку оптимальних меж значень RGB. Це дозволяє виділяти класи. Наприклад, якщо середнє значення зеленого каналу високе, а червоного і синього каналів низьке, модель може припустити, що це ліс. XGBoost автоматично шукає найкраще правило, додаючи нове дерево для покращення класифікації.

Параметр δ є важливою складовою для інформаційно-екстремального підходу. Він визначає, наскільки чутливою буде модель до змін у значеннях ознак для кожного класу даних. Цей параметр діє як свого роду "буфер", який дозволяє моделі розрізняти класи, незважаючи на можливість зміни вихідних даних. Наприклад, при зміні освітлення чи кута зйомки значення RGB можуть трохи змінюватися, але об'єкт залишатиметься стабільним. Параметр δ запобігає помилковій класифікації в таких ситуаціях.

Оптимізація параметра δ проводиться за допомогою аналізу графіків залежності точності від значення δ . Наприклад, графік може показати, як змінюються результати розпізнавання при різних значеннях δ і допомогти обрати оптимальне значення для моделі. При дуже малому значенні δ модель стає надто чутливою до незначних змін і може неправильно класифікувати об'єкти; натомість при надто великому значенні δ модель може нехтувати важливими відмінностями між класами.

Оптимальне значення δ обирається так, щоб мінімізувати функцію втрат при класифікації [11]:

$$\delta^* = \arg \min_{\delta} \sum_{i=1}^N l(y_i, \hat{y}_i(\delta)) \quad (2.3)$$

де $\hat{y}_i(\delta)$ – прогноз для об'єкта i , зроблений при значенні параметра δ ;
 $l(y_i, \hat{y}_i(\delta))$ – функція втрат, яка вимірює, наскільки сильно прогноз відрізняється від справжньої мітки.

На графіках залежності значення критерію Кульбака від параметра δ видно, як відрізняються класи при різних значеннях цього параметра. Це допомагає визначити оптимальне значення δ для того, щоб модель найкраще визначала класи та була стійкою до незначних змін у даних.

Використовуючи XGBoost з оптимальним параметром δ , ми можемо створити модель, яка ефективно розподіляє різні місцевості за їх кольоровими властивостями. Такий підхід дозволяє досягти високої точності і навіть в динамічних умовах забезпечити стабільну роботу моделі. Це робить модель придатною для використання в безпілотних літальних апаратах (БПЛА), де точність класифікації має першорядне значення.

2.4 Опис алгоритму функціонального тестування

Функціональні тести є невід'ємною частиною оцінки ефективності алгоритмів класифікаційних систем. Це особливо важливо у випадках застосування класифікацій для ідентифікації ландшафтних класів на зображеннях – таких як ліси, поля або дороги. Основна мета функціональних тестів полягає у перевірці точності та надійності алгоритму у розпізнаванні об'єктів у різних умовах і виявленні можливих помилок у його функціонуванні. Тести допомагають визначити межі придатності алгоритму та забезпечити стабільність результатів у відповідь на очікування.

Процес функціонального тестування зображень починається з вибору тестових зразків. Для перевірки роботи алгоритму готують набір зображень різних типів ландшафтів, таких як лісовий масив, поля та дороги. Це допомагає

оцінити ефективність алгоритму в реальних умовах класифікації об'єктів на зображеннях. Для покращення якості тестування вибирають як чітко виражені зразки кожного типу ландшафту, так і складніші кадри, де межі між класами можуть бути нечіткими або змішаними.

Для кожного зображення у тестовому наборі даних алгоритм використовує свою систему класифікаційних правил на основі середніх значень кольорових характеристик (наприклад, RGB). На цьому етапі модель оцінює ймовірність того, що кожне зображення належить до певного класу. Наприклад, якщо на зображенні переважають зелені відтінки та певна текстура, алгоритм може припустити, що це ліс. Якщо ж домінують жовтуваті тони, з великою ймовірністю це буде поле чи луг.

Згідно з формальним підходом, для кожного зображення розраховуються значення ймовірностей для кожного класу:

$$P(y = k | X) \quad (2.4)$$

де y – клас зображення (ліс, поле, дорога);

k – індекс конкретного класу;

X – вектор ознак для зображення, який включає середні значення RGB, текстуру тощо.

Алгоритм обирає клас із найвищою ймовірністю як кінцевий результат класифікації для кожного кадру. Наприклад, якщо для конкретного зображення ймовірність $P(y = \text{ліс} | X) = 0.7$, $P(y = \text{поле} | X) = 0.2$, а $P(y = \text{дорога} | X) = 0.1$, то алгоритм класифікує це зображення як "ліс".

Для забезпечення достовірності класифікації важливо перевірити відповідність результатів моделі реальності [19]. У нашому випадку це означає, що алгоритм повинен правильно визначати тип ландшафту на кожному кадрі. Однак у реальних умовах завжди є фактори, які можуть впливати на значення ознак зображення, такі як зміни освітлення, погодні умови, кут зйомки тощо. Для врахування цих варіацій під час тестування вводять поріг толерантності –

значення, яке вказує на допустиме відхилення від ідеальних значень ознак, не втрачаючи при цьому точності.

Поріг толерантності встановлюється як допустиме відхилення у значеннях ймовірностей для кожного класу. Наприклад, якщо алгоритм визначає кадр як належний до класу «ліс» з ймовірністю 0,7 але через певні перешкоди чи шуми ця ймовірність падає до 0,6 поріг толерантності дозволяє алгоритму все ще вважати цей кадр «лісом». Такий підхід допомагає зменшити ризик помилкових класифікацій, спричинених незначними відхиленнями.

Для офіційної перевірки правильності з урахуванням певного рівня толерантності встановлюють припустимий поріг ϵ – найнижчу ймовірність для обраного класу, за якої передбачення вважається коректною класифікацією. До прикладу, якщо $\epsilon = 0.2$, то алгоритм вважає прогнозування точним, але коли ймовірність для передбаченого класу дорівнює $1 - \epsilon = 0.8$.

Процес перевірки виглядає так:

1. Алгоритм розраховує ймовірність приналежності до кожного класу для кожного кадру зображення.
2. Якщо ймовірність обраної категорії дорівнює значенню $1 - \epsilon$, передбачення вважається достовірним; інакше класифікація може розглядатися як сумнівна або потребує додаткової перевірки.

Використання порогу толерантності також дозволяє оцінити стійкість алгоритму при зміні вхідних даних. Графічно цей процес можна показати побудовою кривих залежності точності від порогу толерантності. Наприклад, якщо точність залишається на високому рівні при збільшенні порогу, це свідчить про стабільність алгоритму при зміні даних.

Таким чином, метод перевірки з порогом толерантності допомагає підтримувати високоякісні результати, навіть коли зображення не є ідеальними. Це особливо важливо в автономних системах, таких як безпілотні літальні апарати (БПЛА), де якість вхідних зображень може змінюватися, але точність класифікації все ще є критичною.

3 ІНФОРМАЦІЙНЕ АЛГОРИТМІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ АВТОНОМНОГО БПЛА

Інформаційне, алгоритмічне та програмне забезпечення автономних БПЛА спрямоване на автоматичний аналіз зображень, отриманих у процесі аерофотозйомки. Це включає збір, обробку та класифікацію даних про природні або інфраструктурні об'єкти з використанням методів машинного навчання. Основою для ефективної роботи системи є правильне формування навчальної вибірки, побудова ефективних моделей і їхня оптимізація.

3.1 Формування вхідної навчальної матриці

Підготовка навчальної матриці розпочинається з завантаження і оброблення набору зображень. Кожен кадр має своє середнє значення кольорів (RGB) та класифікацію місцевості: ліс, поле або дорога. Для цього процесу використовують код із функціями для класифікації кадрів і створення навчального набору даних за RGB-значеннями та відповідними класами.

Для позначення кожного кадру застосовувалася функція `label_image_quick()` (Додаток А), яка показувала кожен кадр зображення і дозволяла користувачеві вручну вибрати його клас, натискаючи відповідну клавішу: 0 для лісу, 1 для поля та 2 для дороги. Код забезпечував перехід між кадрами та збереження позначень класів у форматі матриці.

Після вибору класу для кожного кадру функція зберігала мітки у масив `labels`, де кожен елемент вказує на клас конкретного кадру. Ця функція створює навчальну матрицю, що містить RGB-значення та класи для кожного кадру, що дозволяє алгоритму машинного навчання використовувати ці дані для навчання.

Останній крок – зберегти отримані дані у форматі CSV-файлу. Кожне нове зображення обробляється та додається до загального CSV-файлу для зручності подальшого навчання моделі. Функція `add_image_to_dataset()` (Додаток А) перевіряє, чи існує CSV-файл; якщо існує, додає нові дані до нього, інакше створює новий файл. Функція автоматизує весь процес:

1. Процес завантаження фотографії.
2. Виклик функції для позначення кадрів у ручному форматі.
3. Обчислення середнього значення червоного, зеленого та синього кольорів і додавання для них міток класів (0, 1 чи 2).
4. Додавання нових даних з кожного зображення до загального вихідного CSV-файлу.

Цей підхід створює ефективну навчальну матрицю для класифікації місцевості з використанням кольорових характеристик (RGB) і класифікації місцевостей за класом для кожного кадру. Згенерована матриця зберігається у форматі CSV, що значно спрощує подальший процес навчання моделі машинного навчання для автономних безпілотних літальних апаратів.

3.2 Алгоритм машинного навчання з оптимізацією параметрів функціонування системи розпізнавання

Алгоритми машинного навчання повинні ретельно оптимізувати параметри для досягнення високої точності та чіткості класифікаційних категорій зображень, особливо в контексті класифікації зображень. Наприклад, для розпізнавання типів місцевості (ліс, поле, дорога) на зображеннях, отриманих з БПЛА. Цей процес заснований на застосуванні інформаційно-екстремальних критеріїв, які дозволяють обрати оптимальне значення порогового допуску (Δ) і радіуса контейнера кожного класу для класифікації.

Критерій Кульбака-Лейблера (KL-дивергенція) є основою інформаційно-екстремальних критеріїв і заснований на теорії інформації, спрямованої на досягнення максимального розрізнення класифікаційних результатів [11]. Критерій Кульбака вимірює відмінності між двома ймовірнісними розподілами P і Q :

$$D_{KL}(P \parallel Q) = \sum_i P(i) \log \frac{Q(i)}{P(i)} \quad (3.1)$$

де $P(i)$ та $Q(i)$ – ймовірності для класу i відповідно в розподілах P і Q .

Даний критерій дозволяє зрозуміти, наскільки чітко розділені класи. У нашому алгоритмі критерій Кульбака допомагає знайти параметри, для яких клас буде максимально відокремлений.

Графік розподілу впевненості у класах дозволяє перевірити, наскільки впевнено модель присвоює кожен кадр певному класу (рис.1). Для цього використовується поріг толерантності δ , який визначає мінімальний прийнятний рівень впевненості. Наприклад, якщо впевненість класифікації кадру нижча за поріг $1-\delta$, модель розглядає ймовірності інших класів, щоб зменшити ризик помилкової класифікації.

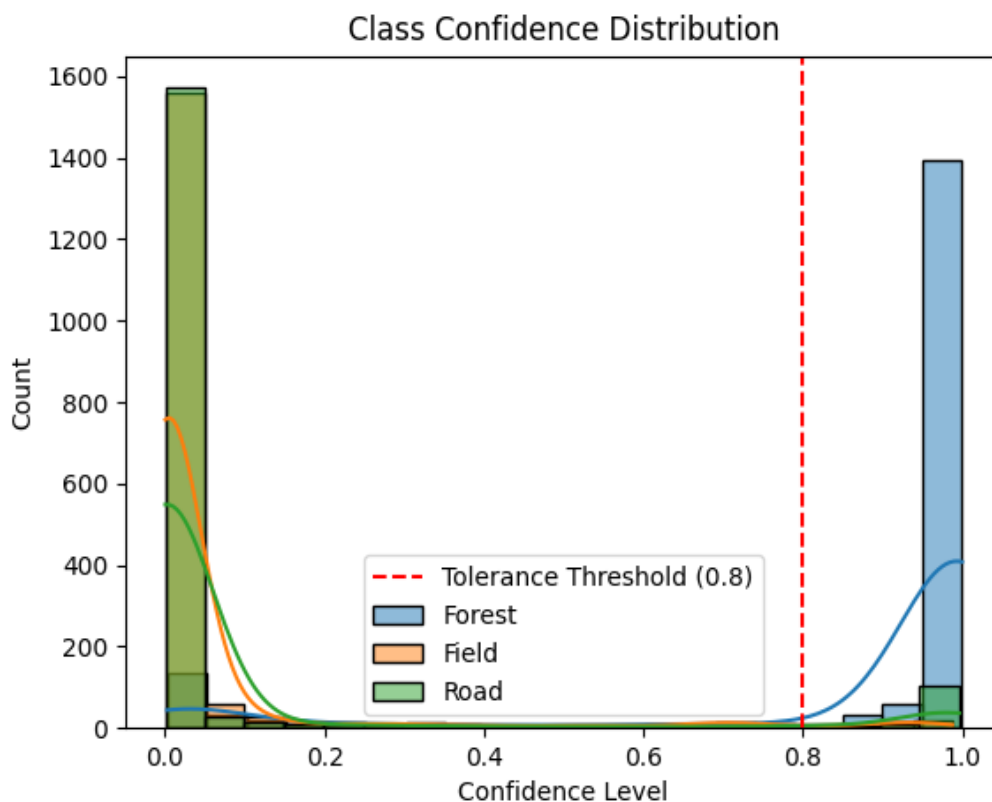


Рисунок 1 – Графік розподілу впевненості класів

Дивлячись на графік (рис.1), можна стверджувати, якщо ймовірність, з якою модель відносить кадр до певного класу, нижча за поріг, модель розглядає ймовірність для інших класів, щоб знизити ризик помилкової класифікації.

В нашому алгоритмі поріг толерантності позначається як δ . Вибір значення δ дозволяє контролювати рівень "впевненості" для кожного кадру. Якщо ймовірність, що кадр належить до певного класу, менша за $1 - \delta$, то алгоритм перевіряє інші класи.

На графіку зображено розподіл ймовірностей для трьох класів:

1. Forest (ліс) – зображено синім кольором.
2. Field (поле) – зображено помаранчевим кольором.
3. Road (дорога) – зображено зеленим кольором.

4. Червона вертикальна лінія на графіку показує поріг толерантності (наприклад, 0.8). Якщо впевненість моделі менша за цей поріг, модель вважає, що класифікація кадру недостатньо надійна і розглядає ймовірність інших класів.

Високі стовпчики з обох боків (біля значень 0 та 1) вказують, що модель має високу впевненість у своїй класифікації для багатьох кадрів.

Далі представлено графік залежності критерію Кульбака від параметра Delta (рис.2).

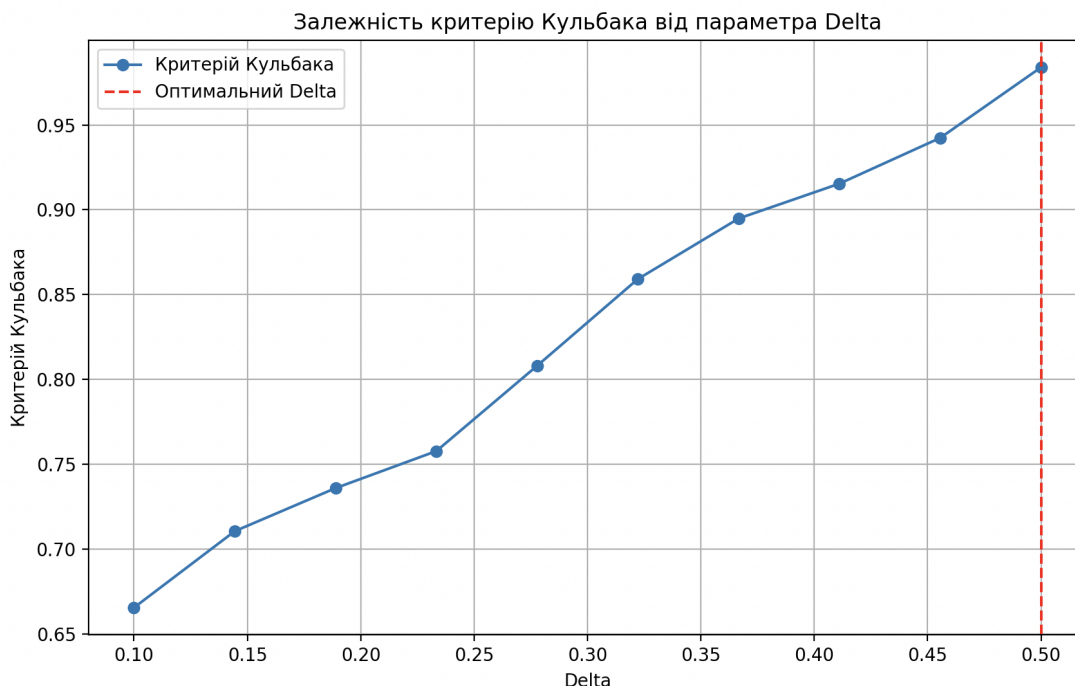


Рисунок 2 – Графік залежності критерію Кульбака від параметра Delta

На цьому графіку показана залежність критерію Кульбака від параметра Delta, використовуваного в якості порога толерантності.

Параметр Delta важливий, оскільки він контролює толерантність моделі до вразливостей. Оптимальне значення Delta допомагає максимально розрізнити класи і забезпечувати точність класифікації. Щоб оцінити якість класифікації, був розрахований середній критерій Кульбака-Лейблера для всіх кадрів і визначена ймовірність того, що кадри потрапить в межі заданої Delta.

Залежність критерію Кульбака від параметра Delta можна представити так:

$$K(\delta) = \frac{1}{N} \sum_{i=1}^N D_{KL}(P_i \parallel Q_i) \text{ для } P_i \geq (1 - \delta) \quad (3.2)$$

де $K(\delta)$ – середнє значення критерію Кульбака для всіх кадрів;

N – загальна кількість кадрів;

P_i – ймовірність приналежності кадру до правильного класу;

Q_i – розподіл ймовірностей для всіх класів.

На цьому графіку (рис. 2) показано взаємозв'язок між критерієм Кульбака і параметром Delta. Оптимальне значення Delta може бути використано для отримання максимальної ідентифікації класу. Показник якості класифікації можна визначити як середній критерій Кульбака для всіх кадрів.

Зі збільшенням Delta значення критерію Кульбака зростає, і розділення класів стає чіткішим. Проте після певного значення Delta (приблизно 0.5) цей розвиток припиняється, що вказує на оптимальне значення. При оптимальному Delta модель досягає максимальної точності у розрізненні між класами.

Не менш важливим є радіус контейнера, який визначає область навколо центру класу, в межах якої кадри можна вважати належними до цього класу. Оптимізація радіуса контейнерів дозволяє зменшити помилкові класифікації за рахунок зменшення ймовірності перекривання кадрів різних класів.

Критерій Кульбака для радіуса r можна обчислити так:

$$K(r) = - \sum_j p_j \log(p_j) \quad (3.3)$$

де p_j – ймовірність кадрів класу j , які знаходяться в межах радіуса r від центру класу. Радіус обирається таким, щоб значення $K(r)$ було максимальним.

Для більш детального аналізу, були створені графіки залежності критерію Кульбака від радіуса контейнера для кожного класу.

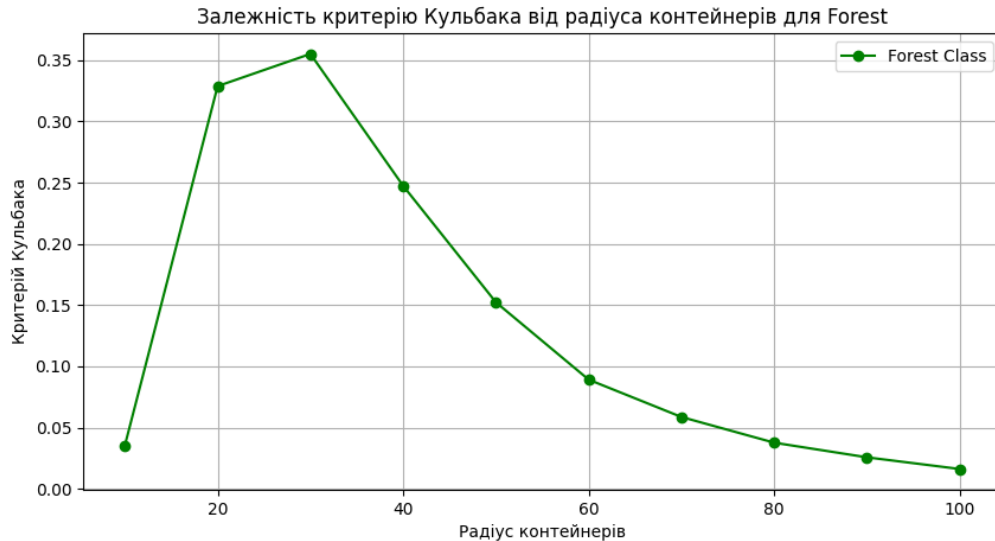


Рисунок 3 – Графік залежності критерію Кульбака від радіуса контейнерів для класу Forest

Згідно з графіком (рис. 3) можна визначити таке: коли радіус дуже малий, значення критерію Кульбака низьке, що свідчить про велике перекриття між класом “Forest” та іншими класами даних категорій. Це пояснюється тим фактом, що при таких невеликих розмірах контейнера об’єкти класу “Forest” не можуть бути ефективно розділені один від одного. Максимальне розділення спостерігається в інтервалі радіусу 20–40: саме тоді характеристика Кульбака досягає свого максимуму. Цей факт показує, що при цьому розмірі контейнера у класу “Forest” виявляється найбільше розділення з іншими категоріями. Цей радіус є оптимальним для максимального відокремлення об’єктів цього класу. Якщо радіус контейнера перевищує 40, це призводить до зниження показника Кульбака. Це означає, що об’єкти інших класів включаються в контейнер через надто великий радіус, що зменшує точність для класу “Forest”.

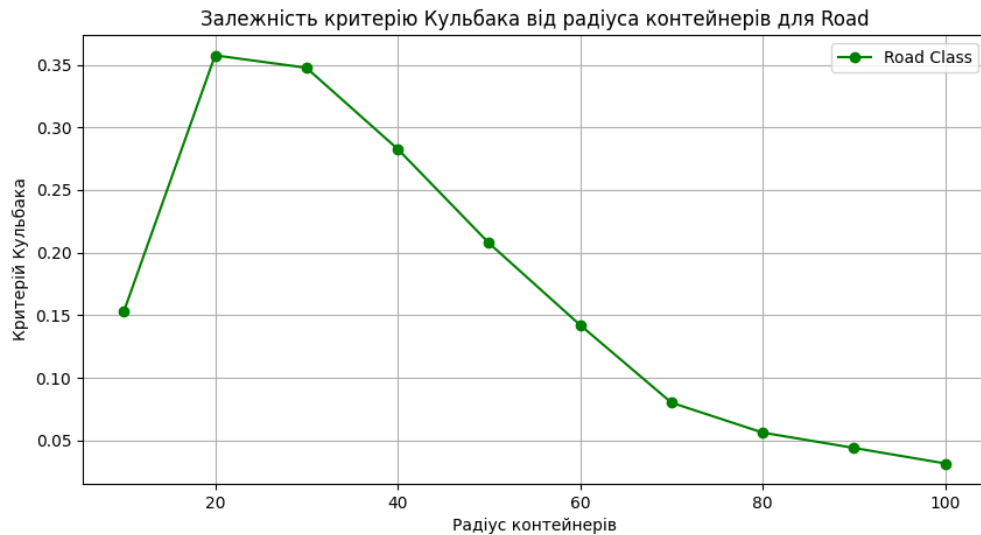


Рисунок 4 – Графік залежності критерію Кульбака від радіуса контейнерів для класу Road

За графіком (рис. 4) можна визначити, що спостерігаються схожі закономірності: коли радіус невеликий, значення критерію Кульбака низьке, що свідчить про значне перетинання класу “Road” з іншими класами. Невеликий контейнер не може ефективно розділити об’єкти класу “Road” від об’єктів інших класів. Оптимальний радіус для класу “Road” знаходиться у діапазоні 20–30, де критерій Кульбака досягає пікового значення. Це означає найменше перетинання з іншими класами при даному радіусі, забезпечуючи найкращу розрізнявальність об’єктів класу “Road”. Збільшення радіуса помітно знижувало значення критерію Кульбака, що вказувало на те, що клас “Road” перекривався з іншими класами через включення об’єктів інших класів у межах контейнера.

Після аналізу графіків для класів “Forest” та “Road” можна прийти до висновку про наявність ідеального діапазону радіусів контейнерів (приблизно 20–40), який забезпечуватиме найбільшу чіткість між класами. Це значення гарантуватиме максимальне уникнення перекриття між об’єктами різних класів та точну класифікацію. Занадто малий радіус не забезпечить достатнього виокремлення об’єктів одного класу від іншого; у той же час занадто великий

радіус охоплюватиме об'єкти з інших класів та понизить ефективність розпізнавання.

Оптимізація параметрів Delta та радіуса контейнерів має вирішальне значення для підвищення точності класифікації місцевостей на зображеннях із БПЛА. Використання критерію Кульбака-Лейблера дозволяє визначити оптимальні значення цих параметрів для досягнення найкращих результатів у розрізненні між класами.

3.3 Побудова бінарного дерева даних

Коли базовий класифікатор із застосуванням методів машинного навчання не досягає необхідної точності, доцільно використовувати ієрархічну класифікацію для покращення розрізненості між категоріями. Один із ефективних методів – побудова рекурсивного бінарного дерева (дерева рішень), яке дозволяє поетапно ділити вхідну інформацію на підкласи. Це значно підвищує точність класифікації, особливо для випадків коли у нас складні або невизначені дані.

Ієрархічна класифікація з використанням бінарного дерева рішень передбачає створення впорядкованого рішення, де на кожному вузлі дані поділяються на два підкласи (або гілки) на основі таких характеристик, як середнє значення кольору та яскравості.

Розглянемо метод побудови бінарного дерева рішень:

1. На початкових етапах алгоритм використовує прості функції, такі як середні значення RGB, для розподілу даних на два основні типи.
2. Кожен вузол у дереві використовує задані критерії, щоб визначити, який із двох підкласів обирати.
3. Розгалуження припиняється, коли модель досягає бажаного рівня точності на певному рівні в дереві рішень. Якщо точність недостатня, дерево продовжить поглиблюватися, щоб досягти кращих результатів.

Для застосування цього підходу можна використовувати алгоритм `DecisionTreeClassifier` із бібліотеки `sklearn`. Метод класифікації рекурсивного

бінарного дерева (Додаток Б) будує дерево на основі вхідних даних, налаштовуючи багаторівневий поділ даних. Параметр `max_depth` у коді визначає максимальну кількість рівнів дерева, що запобігає перенавчанню та підтримує стабільність результатів.

Бінарне дерево прийняття рішень, що використовується для ієрархічної класифікації, дозволяє поетапно розбивати дані на підкласи, забезпечують максимальну чіткість на всіх рівнях. Це досягається шляхом обчислення оптимального розподілу на основі вибраних характеристик, таких як середні значення RGB.

Якщо базовий класифікатор (наприклад, XGBoost) не забезпечує достатньої точності, бінарні дерева можуть допомогти в подальшій обробці складних сценаріїв і зменшити кількість помилок класифікації. Наприклад, на першому рівні дерево може розділити дані на “зони зеленого кольору” та “зони не зеленого кольору”. На наступних рівнях деталізація підкласів дозволить ще більше покращити точність класифікації.

Використання дерева рішень із рекурсивною структурою підвищує точність алгоритму за рахунок ієрархічного поділу даних на підгрупи, що особливо ефективно у випадках, коли базові методи класифікації не досягають потрібної точності результатів.

3.4 Алгоритм ієрархічного машинного навчання

Використання ієрархічного підходу в машинному навчанні дозволяє підвищити точність класифікації шляхом послідовного поділу даних на підкласи на кожному рівні ієрархії. У нашому випадку це означає, що кожен рівень виконує певні аспекти класифікації, створюючи багаторівневу модель, яка поступово обмежує можливу класифікацію.

Ієрархічний підхід дозволяє систематизувати процес класифікації, щоб кожен рівень моделі міг виконувати певні завдання розпізнавання. На першому рівні існує грубий поділ на основні категорії (наприклад, «зелена зона» і «нежива

зона»). Потім, на нижчих рівнях, дані поступово розбиваються на менші підкласи, щоб отримати більш детальну класифікацію.

У нашій моделі ми використовуємо класифікатор "Дерево рішень", який оптимізує критерії розподілу для кожного вузла. Серед найпоширеніших критеріїв – це індекс Джині та ентропія, які допомагають зменшити на кожному підрівні невизначеність інформації та забезпечують більшу точність.

Формула для індексу Джині [22]:

$$G(D) = 1 - \sum_{i=1}^T p_i^2 \quad (3.4)$$

де p_i – частка кадрів, що належать до класу i .

Формула для ентропії [22]:

$$H = - \sum_{i=1}^T p_i \log_2 p_i \quad (3.5)$$

Метрики інформаційності, такі як ентропія, дозволяють визначити рівень хаосу в наших даних. Це є важливий критерій для вибору найкращих розбиттів у вузлах дерева рішень.

Для кожного рівня в ієрархічному дереві ми оптимізували параметри для досягнення максимально можливої точності кожного підкласу. Це включає налаштування таких параметрів:

1. Максимальна глибина дерева (`max_depth`): обмежує кількість рівнів у дереві, щоб уникнути надмірного перенавчання. Чим більша глибина, тим складніші залежності може охоплювати модель, але це може призвести до перенавчання.

2. Мінімальна кількість прикладів для розділення вузлів (`min_samples_split`): визначає мінімальну кількість точок даних, необхідних для розділення вузла. Це запобігає надмірному розділенню в вузлах з невеликим обсягом даних.

3. Критерій розбиття (наприклад, індекс Джині або ентропія): на кожному рівні дерево вибирає розбиття, який максимально зменшує значення критеріїв, забезпечуючи більш точну класифікацію.

У нашому коді функція `recursive_binary_tree_classification` (Додаток Б) здійснює цю процедуру шляхом налаштування параметрів дерева рішень. Вона створює дерево з обмеженням глибини (`max_depth=3`), щоб уникнути перенавчання і забезпечити достатню деталізацію класифікації. Параметр `max_depth` можна змінювати залежно від потреб у точності та обсягу даних.

Ієрархічна структура класифікації підвищує точність результатів, коли проста модель не забезпечує достатнього поділу класів. Такий підхід дозволяє поступово знижувати невизначеність і досягати максимальної ефективності класифікації. Це дозволяє глибше класифікувати дані та зменшує кількість помилок у складних сценаріях.

3.5 Короткий опис програми

Програма складається з декількох основних складових частин: кожна з них використовується для різних етапів обробки даних: підготовки навчального набору для моделювання, тренування моделі, прогнозування результатів та візуалізації. Основна мета цієї програми полягає в автоматичному класифікуванні природних об'єктів (такі як лісові масиви, поля чи дороги) на основі зображень за допомогою методів машинного навчання, зокрема алгоритму XGBoost. Далі надано розширений опис ключових структурних компонентів програми.

На початковому етапі програма обробляє зображення та готує дані для навчання за допомогою функціоналу `add_image_to_dataset` (Додаток А), який автоматизує процес збору даних, розбиваючи зображення на дрібні кадри та призначаючи їм класи.

Функція `label_image_quick` (Додаток А) дозволяє користувачам вручну призначати класи для кожного кадру шляхом натискання клавіш "0", "1" або "2"

для позначення класів “ліс”, “поле” або “дорога”. Потім отримані дані зберігаються у форматі CSV для подальшого використання в процесі навчання.

У процесі навчання використовується функція `train_classifier` (Додаток Б), яка навчає модель на основі RGB-ознак із навчальної вибірки.

Для передбачення класу місцевості програма використовує функцію `predictTerrainWithTolerance` (Додаток Б). Ця функція класифікує кожен кадр зображення на основі середніх значень кольору RGB і враховує значення порогу толерантності (`tolerance`). Якщо впевненість у передбаченні низька, модель шукатиме найближчий клас у межах заданого порогу та коригуватиме результат.

Для ілюстраційного відображення результатів програма використовує функцію `overlay_predictions` (Додаток Б), яка накладає передбачені класи на зображення та відображає клас для кожного кадру числовим маркуванням. Це дозволяє користувачам побачити, як модель класифікує кожен частину зображення.

Для перевірки точності та стабільності прогнозів програма будує графіки розподілу впевненості класифікатора для кожного класу, а також графіки критерію Кульбака-Лейблера. Функція `plot_class_confidences` (Додаток Б) дає змогу графічно визначити рівень впевненості в прогнозах для кожного класу та корисна для налаштування порогу толерантності.

Функції `plot_kullback_vs_delta` і `plot_kullback_vs_radius` (Додаток Б) створюють графіки, що зображують залежність критерію Кульбака від `delta` й радіуса контейнерів відносно кожного класу. Це сприяє оптимізації параметрів моделі для покращення точності класифікаційних результатів.

З цього можна сказати, якщо базова класифікація не досягає бажаного рівня точності, то програма використовує додатковий алгоритм класифікації на основі рекурсивного бінарного дерева. Ця функція (Додаток Б) дозволяє створити ієрархічну структуру для прийняття рішень. Завдяки цій структурі, підвищується ефективність класифікаційного процесу. Й також це дає

можливість класифікатору поступово розподіляти дані на підкатегорії для більш точного визначення об'єктів на зображеннях.

Узагальнюючи, надана програма є повноцінним рішенням для автоматичного визначення територій на зображеннях, отриманих з БПЛА. Програма охоплює обробку даних, навчання моделей, прогнозування та візуалізацію. Використання алгоритмів XGBoost і бінарних дерев дозволяє досягти високої точності класифікації навіть для зображень, що є низькоякісними чи складними.

3.6 Результати комп'ютерного моделювання

Основна мета полягала в перевірці розробленого алгоритму на реальних зображеннях і їх поділі на сегменти. На доданих зображеннях видно результати класифікаційного процесу. Алгоритм ефективно визначив три типи місцевості: лісовий масив, поле та дорогу.

На прикріплених зображеннях (рис 4, 5, 6) показані результати класифікації, де кожен піксель отримав відповідну мітку класу:

1. Ліс (позначений як 0) – область із щільною зеленню, представлена середнім значенням RGB.
2. Поле (позначене як 1) – відкриті території з менш щільною зеленню або без зелені.
3. Дорога (позначена як 2) – асфальтовані зони з характерними значеннями RGB.

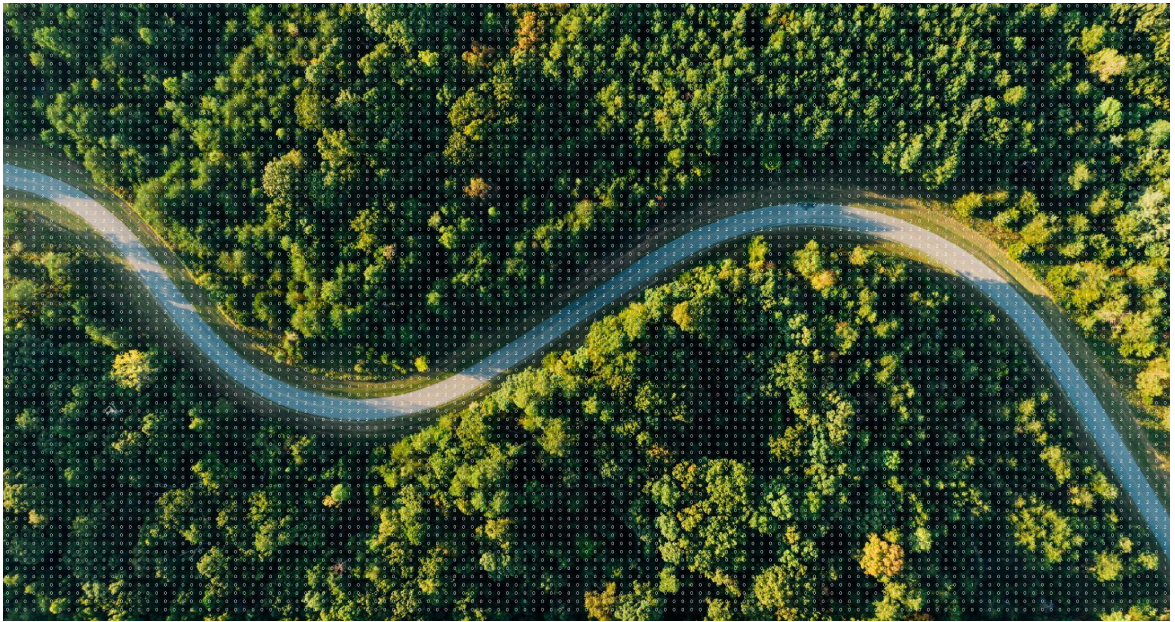


Рисунок 5 – Лісова місцевість з дорогою

На цьому зображенні (рис. 5) дорога чітко відокремлена від лісової ділянки, що дозволило алгоритму легко ідентифікувати ділянки дороги та лісу. По результатам видно, що більшість лісових ділянок позначено як 0, а дороги – як 2. Це вказує на високу точність алгоритму у розпізнаванні меж між класами навіть у складних природних умовах.

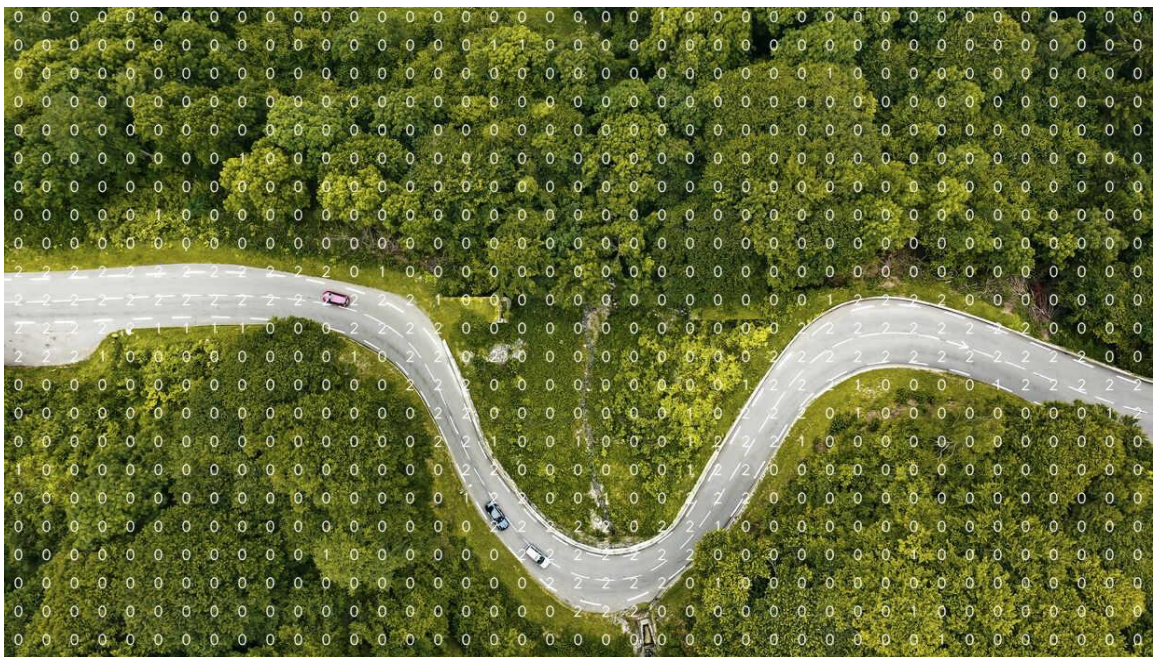


Рисунок 6 – Місцевість з більш звивистою дорогою

Алгоритм продемонстрував здатність точно розпізнавати дороги навіть у разі складних криволінійних ділянок (рис 6). Звивисті дороги також були чітко класифіковані як клас 2. Завдяки використанню інформаційно-екстремальних критеріїв і оптимізації параметрів, алгоритм успішно визначив межі між дорогою та прилеглими лісовими ділянками.



Рисунок 7 – Дорога з відкритою місцевістю

На останньому зображенні (рис. 7) представлена довга пряма ділянка дороги без значної кількості лісових насаджень. Результати класифікації підтверджують, що алгоритм успішно визначає дорогу (клас 2) та прилеглі поля (клас 1). Таке розмежування особливо важливе для задач, що вимагають високої точності, таких як навігація або моніторинг дорожньої інфраструктури.

Отже, отримані результати комп'ютерного моделювання демонструють високу точність та надійність алгоритму. Використання функцій та методів, таких як *predict_terrain_with_tolerance* та *overlay_predictions* (Додаток Б), дозволило досягти глибокого розуміння текстур і кольорових особливостей зображень. Гнучкість алгоритму в обробці різних сценаріїв з різними типами місцевості є однією з основних переваг розробленої системи.

ВИСНОВКИ

У результаті виконання роботи було створено інформаційно-екстремальний алгоритм для автоматичної класифікації місцевості за зображеннями, отриманими від безпілотних літальних апаратів (БПЛА). Було розглянуто різноманітні підходи до обробки зображень та обрано оптимальні інструменти для досягнення поставлених цілей. Під час розробки враховано всі вимоги до програмного продукту.

У ході виконання кваліфікаційної роботи магістра було виконано наступні завдання:

1. Проведено аналіз сучасних інформаційних технологій та методів машинного навчання для обробки зображень.
2. Розроблено алгоритм класифікації зображення за допомоги інформаційно-екстремальних критеріїв, критерію Кульбака й XGBoost.
3. Виконано комп'ютерне моделювання та протестовано алгоритм на реальних даних.
4. Проаналізовано точність класифікації та налаштування моделі.
5. Розроблено багаторівневий підхід до класифікації з використанням рекурсивного бінарного дерева.
6. Проаналізовано отримані результати та зроблено висновки щодо якості роботи інформаційної технології.

У результаті дослідження розроблено інформаційну систему, яка забезпечує точну класифікацію природних об'єктів та інфраструктури, таких як ліси, поля, дороги. Висока точність та можливість обробки великих обсягів даних дозволяють використовувати систему для моніторингу територій у режимі реального часу. Це значно скорочує час на прийняття рішень та підвищує ефективність обробки даних для таких галузей, як сільське господарство, екологічний моніторинг та управління дорожньою інфраструктурою.

За результатами роботи інформаційної системи, можна зробити висновок, що застосування розробленого алгоритму для автономного аналізу зображень підвищує точність та ефективність прийняття рішень на основі оброблених даних. Це досягається завдяки використанню інформаційно-екстремальних методів машинного навчання, які оптимізують параметри обробки зображень, а також забезпечують мінімізацію помилок класифікації. Результати дослідження підтверджують, що впровадження запропонованого підходу дозволяє зменшити час обробки великих обсягів даних і покращити якість аналізу складних структур, таких як лісові масиви, дорожні покриття або польові ділянки.

Подальші дослідження в цьому напрямку можуть бути зосереджені на розширенні функціональних можливостей алгоритму, інтеграції з іншими інформаційними системами та оптимізації ресурсів для використання в реальних умовах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. БПЛА: Що Це Таке та Як і Де Їх Використовують: веб-сайт. URL: <https://mohyliv.info/bpla/> (дата звернення: 06.11.2024)
2. Дрони - багатомільярдне майбутнє сільського господарства: веб-сайт. URL: https://agro.24tv.ua/droni-bagatomilyardne-maybutnye-silskogo-gospodarstva-povini-sogodni_n1522661 (дата звернення: 06.11.2024)
3. Стале відновлення України: внесок БПЛА у цифрову трансформацію лісового господарства: веб-сайт. URL: <https://culver.aero/uk/news/stalie-vidnovliennia-ukrayini-vniesok-bp-la-u-tsifrovu-transformatsiiu-lisovoghoghospodarstva> (дата звернення: 06.11.2024)
4. Еволюція штучного інтелекту в безпілотах: Наслідки та адаптація для технологій протидії безпілотах: веб-сайт. URL: <https://sentrycs.com/uk/the-counter-drone-blog/the-evolution-of-ai-in-drones-implications-and-adaptations-for-counter-drone-technology/> (дата звернення: 06.11.2024)
5. Голенко М. Ю., Іванов Д. І., Єфіменко А. А., Воротніков В. В. Аналіз методів розпізнавання об'єктів та компресії зображень під час аерофотозйомки з БПЛА. URL: <https://eztuir.ztu.edu.ua/bitstream/handle/123456789/8200/146.pdf> (дата звернення: 06.11.2024)
6. Big Data: що це таке, як шукати, зберігати та використовувати: веб-сайт. URL: <https://unihost.com/blog/uk/big-data/> (дата звернення: 06.11.2024)
7. Штучний інтелект та машинне навчання: що спільного, у чому різниця між ними і чому це важливо?: веб-сайт. URL: <https://blog.colobridge.net/uk/2024/05/artificial-intelligence-and-machine-learning-ua/> (дата звернення: 06.11.2024)
8. Машинне навчання: як штучний інтелект вчиться і розвивається: веб-сайт. URL: <https://bizmag.com.ua/mashynne-navchannya/> (дата звернення: 06.11.2024)

9. Що таке Machine Learning?: веб-сайт. URL:
<https://denovo.ua/resources/what-is-machine-learning> (дата звернення: 06.11.2024)
10. Типи задач у машинному навчанні: від класифікації до кластеризації:
веб-сайт. URL: <https://freshtech.global/ua/blog/types-of-tasks-in-machine-learning-from-classification-to-clustering> (дата звернення: 06.11.2024)
11. XGBoost Documentation: веб-сайт. URL:
<https://xgboost.readthedocs.io/en/latest/index.html> (дата звернення: 06.11.2024)
12. Дерево рішень – Методи прийняття управлінських рішень: веб-сайт.
URL: https://stud.com.ua/31896/menedzhment/derevo_rishen (дата звернення:
06.11.2024)
13. Шелехов І. В., Пилипенко С. О., Столярчук О. О., Романенко Т. А.
Інформаційно-екстремальне машинне навчання системи контролю знань. URL:
<http://samit.khpi.edu.ua/article/view/2079-0023.2018.44.09/151801> (дата звернення:
06.11.2024)
14. Machine Learning Tutorial: веб-сайт. URL:
<https://www.guru99.com/uk/machine-learning-tutorial.html> (дата звернення:
06.11.2024)
15. Ільницький А. І., Бурба О. І. Statistical Criteria for Assessing the
Informativity of the Sources of Radio Emission of Telecommunication Networks and
Systems in Their Recognition. URL:
[https://www.researchgate.net/publication/336913472_STATISTICAL_CRITERIA_F
OR_ASSESSING_THE_INFORMATIVITY_OF_THE_SOURCES_OF_RADIO_E
MISSION_OF_TELECOMMUNICATION_NETWORKS_AND_SYSTEMS_IN_T
HEIR_RECOGNITION](https://www.researchgate.net/publication/336913472_STATISTICAL_CRITERIA_FOR_ASSESSING_THE_INFORMATIVITY_OF_THE_SOURCES_OF_RADIO_EMISSION_OF_TELECOMMUNICATION_NETWORKS_AND_SYSTEMS_IN_THEIR_RECOGNITION) (дата звернення: 06.11.2024)
16. Основи машинного навчання, Вступ в машинне навчання. Етапи
вирішення завдань машинного навчання: веб-сайт. URL:
https://stud.com.ua/139970/informatika/osnovi_mashinnogo_navchannya (дата
звернення: 06.11.2024)

17. KL Divergence in Machine Learning: веб-сайт. URL:
<https://encord.com/blog/kl-divergence-in-machine-learning/> (дата звернення:
06.11.2024)
18. Метод k-найближчих сусідів: веб-сайт. URL:
https://uk.wikipedia.org/wiki/Метод_k-найближчих_сусідів (дата звернення:
06.11.2024)
19. Оцінка точності: веб-сайт. URL:
[https://ukrayinska.libretexts.org/Науки_про_Землю/Географія_\(фізична\)/Дистанційне_зондування_\(Knudby\)/01:_Розділи/1.07:_Оцінка_точності](https://ukrayinska.libretexts.org/Науки_про_Землю/Географія_(фізична)/Дистанційне_зондування_(Knudby)/01:_Розділи/1.07:_Оцінка_точності) (дата звернення:
06.11.2024)
20. Довбиш А.С. Основи проектування інтелектуальних систем / А.С. Довбиш.-Суми:Видавництво СумДУ, 2009. 171 с.
21. Довбиш А. С., Васильєв А. В., Любчак В. О. Інтелектуальні технології в електронному навчанні. Суми: Видавництво СумДУ, 2013. 172 с.
22. Decision tree learning : веб-сайт. URL:
https://en.wikipedia.org/wiki/Decision_tree_learning (дата звернення: 08.11.2024).

ДОДАТОК А

Лістинг коду програми - 01_generate_labels_datasets.py

```

import numpy as np
import pandas as pd
import cv2
import matplotlib.pyplot as plt
import os

def label_image_quick(image, frame_size):
    h, w, _ = image.shape
    grid_h, grid_w = h // frame_size, w // frame_size
    labels = np.zeros((grid_h, grid_w), dtype=int)

    # Індекс кадру для маркування
    current_frame = [0, 0] # [i, j]

    # Функція для обробки натискань клавіш
    def on_key(event):
        nonlocal current_frame
        if event.key in ['0', '1', '2']:
            labels[current_frame[0], current_frame[1]] = int(event.key)
            plt.close()

        # Переходимо до наступного кадру
        if current_frame[1] < grid_w - 1:
            current_frame[1] += 1
        elif current_frame[0] < grid_h - 1:
            current_frame[0] += 1
            current_frame[1] = 0
        else:
            current_frame[0] = -1

    # Цикл для показу кожного кадру та очікування натискання клавіші
    while current_frame[0] != -1:
        i, j = current_frame
        frame = image[i*frame_size:(i+1)*frame_size, j*frame_size:(j+1)*frame_size]
        plt.imshow(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
        plt.title(f"Виберіть мітку для кадру ({i}, {j}) - натисніть 0 (ліс), 1 (поле), 2 (дорога)")
        plt.gcf().canvas.mpl_connect('key_press_event', on_key)
        plt.show()

```

```

return labels

def generate_dataset(image, frame_size, labels):
    h, w, _ = image.shape
    data = []

    for i in range(labels.shape[0]):
        for j in range(labels.shape[1]):
            y, x = i * frame_size, j * frame_size
            frame = image[y:y+frame_size, x:x+frame_size]
            avg_color = frame.mean(axis=(0, 1))
            label = labels[i, j] # клас місцевості з масиву міток

            data.append([avg_color[0], avg_color[1], avg_color[2], label])

    # Перетворюємо в DataFrame для зручного збереження
    df = pd.DataFrame(data, columns=["R", "G", "B", "Label"])
    return df

# Основна функція для додавання нового зображення до датасету
def add_image_to_dataset(image_path, frame_size, csv_file="terrain_dataset.csv"):
    # Завантажуємо зображення
    image = cv2.imread(image_path)
    labels = label_image_quick(image, frame_size)

    # Генеруємо датасет
    dataset = generate_dataset(image, frame_size, labels)

    # Додаємо нові дані до загального CSV-файлу
    try:
        # Завантажуємо існуючий датасет, якщо файл вже існує
        existing_data = pd.read_csv(csv_file)
        # Об'єднуємо існуючі дані з новими
        dataset = pd.concat([existing_data, dataset], ignore_index=True)
    except FileNotFoundError:
        # Якщо файл не існує, просто зберігаємо новий датасет
        pass

    # Зберігаємо оновлений датасет
    dataset.to_csv(csv_file, index=False)
    print(f"Додано дані з {image_path} до {csv_file}")

```

```
# Шлях до папки з зображеннями
image_folder = "test_imgs/"

# Отримуємо список всіх зображень у папці
image_paths = [os.path.join(image_folder, f) for f in os.listdir(image_folder) if
f.endswith(('.jpg', '.png', '.jpeg'))]

print("Знайдені зображення:", image_paths)

frame_size = 50 # Розмір кадру

# Додаємо кожне зображення з папки до датасету
for image_path in image_paths:
    add_image_to_dataset(image_path, frame_size)
```

ДОДАТОК Б

Лістинг коду програми - 02_recognize_alg.py

```

import numpy as np
import pandas as pd
import cv2
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, accuracy_score
from xgboost import XGBClassifier
from sklearn.tree import DecisionTreeClassifier

# 1. Завантаження датасету та вилучення RGB ознак
data = pd.read_csv("terrain_dataset.csv")
X = data[["R", "G", "B"]].values
y = data["Label"].values

# 2. Навчання XGBoost класифікатора
def train_classifier(X, y, n_estimators=100, max_depth=None, learning_rate=0.1):
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
    classifier = XGBClassifier(n_estimators=n_estimators, max_depth=max_depth,
learning_rate=learning_rate, random_state=42)
    classifier.fit(X_train, y_train)

    y_pred = classifier.predict(X_test)
    print("Звіт класифікації:\n", classification_report(y_test, y_pred))
    print("Точність:", accuracy_score(y_test, y_pred))

    return classifier

# 3. Прогноз місцевості з оптимізацією толерантності
def predict_terrain_with_tolerance(classifier, image, frame_size, tolerance=0.2):
    h, w, _ = image.shape
    h = (h // frame_size) * frame_size
    w = (w // frame_size) * frame_size
    image = image[:h, :w]

    frames = []
    for y in range(0, h, frame_size):
        for x in range(0, w, frame_size):

```

```

    frame = image[y:y+frame_size, x:x+frame_size]
    avg_color = np.mean(frame, axis=(0, 1)) # Середні значення RGB
    frames.append(avg_color)

frames = np.array(frames)
predictions = classifier.predict_proba(frames)

optimized_predictions = []
for pred in predictions:
    max_prob = np.max(pred)
    best_class = np.argmax(pred)

    if max_prob < 1 - tolerance:
        possible_classes = np.where(pred >= max_prob - tolerance)[0]
        best_class = possible_classes[np.argmax(pred[possible_classes])]

    optimized_predictions.append(best_class)

optimized_predictions = np.array(optimized_predictions)
grid_h, grid_w = h // frame_size, w // frame_size
prediction_grid = optimized_predictions.reshape(grid_h, grid_w)

return prediction_grid

# 4. Накладення прогнозів на зображення
def overlay_predictions(image, prediction_grid, frame_size,
output_path="output_image.jpg"):
    overlay_image = image.copy()
    grid_h, grid_w = prediction_grid.shape

    for i in range(grid_h):
        for j in range(grid_w):
            y, x = i * frame_size, j * frame_size
            label = prediction_grid[i, j]
            cv2.putText(
                overlay_image,
                str(label),
                (x + frame_size // 3, y + frame_size // 2),
                cv2.FONT_HERSHEY_SIMPLEX,
                0.5,
                (255, 255, 255),
                1,
                cv2.LINE_AA

```

```

    )

    cv2.imwrite(output_path, overlay_image)
    print(f"Зображення збережено як {output_path}")

# 5. Побудова розподілів впевненості класів
def plot_class_confidences(predictions, tolerance,
save_path="class_confidence_distribution.png"):
    confidence_df = pd.DataFrame(predictions, columns=["Forest", "Field", "Road"])
    for column in confidence_df.columns:
        sns.histplot(confidence_df[column], kde=True, label=column, bins=20)
        plt.axvline(1 - tolerance, color="red", linestyle="--", label=f"Tolerance Threshold
({1 - tolerance})")
        plt.title("Class Confidence Distribution")
        plt.xlabel("Confidence Level")

    plt.legend()
    plt.savefig(save_path)
    plt.show()

# 6. Розрахунок та побудова критерію Кульбака для дельти та радіуса
def calculate_kullback_criterion(predictions, labels, delta):
    classes = np.unique(labels)
    kullback_values = []

    for cls in classes:
        class_preds = predictions[labels == cls]
        within_tolerance = np.sum((class_preds >= (1 - delta)))
        kullback_value = within_tolerance / len(class_preds)
        kullback_values.append(kullback_value)

    return np.mean(kullback_values)

def plot_kullback_vs_delta(classifier, X, y, delta_range=np.linspace(0.1, 0.5, 10),
save_path="kullback_vs_delta.png"):
    kullback_scores = []
    for delta in delta_range:
        predictions = classifier.predict_proba(X).max(axis=1)
        kullback_score = calculate_kullback_criterion(predictions, y, delta)
        kullback_scores.append(kullback_score)

    plt.plot(delta_range, kullback_scores, marker='o')
    plt.xlabel("Delta")

```

```

plt.ylabel("Критерій Кульбака")
plt.title("Залежність критерію Кульбака від параметра Delta")
plt.savefig(save_path)
plt.show()

def plot_kullback_vs_radius(X, y, target_class="", class_name="",
save_path="kullback_vs_radius.png", radius_range=range(10, 110, 10)):
    kullback_scores = []

    class_data = X[y == target_class]

    if len(class_data) == 0:
        print(f"No data for class '{class_name}'. Cannot generate plot.")
        return

    center = class_data.mean(axis=0)

    for radius in radius_range:
        distances = np.linalg.norm(class_data - center, axis=1)
        within_radius_count = np.sum(distances <= radius)

        if within_radius_count == 0:
            print(f"No data within radius {radius} for class '{class_name}'. Skipping this
radius.")
            kullback_scores.append(0)
            continue

        within_radius = within_radius_count / len(class_data)

        kullback_score = -within_radius * np.log(within_radius) if within_radius > 0
else 0
        kullback_scores.append(kullback_score)

    plt.figure(figsize=(10, 5))
    plt.plot(radius_range, kullback_scores, marker='o', color="green",
label=f'{class_name} Class')
    plt.xlabel("Радіус контейнерів")
    plt.ylabel("Критерій Кульбака")
    plt.title(f"Залежність критерію Кульбака від радіуса контейнерів для
{class_name}")

    plt.legend()
    plt.grid(True)

```



```

plt.savefig(save_path)
plt.show()

# 7. Класифікація за допомогою декурсивного бінарного дерева
def recursive_binary_tree_classification(X, y, max_depth=3):
    tree_classifier = DecisionTreeClassifier(max_depth=max_depth,
random_state=42)
    tree_classifier.fit(X, y)
    return tree_classifier

# Основний запуск
image_name = "01"
image_path = f'input_images/{image_name}.jpeg'
image = cv2.imread(image_path)
frame_size = 30
tolerance = 0.2

classifier = train_classifier(X, y, n_estimators=100, max_depth=5, learning_rate=0.1)
terrain_prediction = predict_terrain_with_tolerance(classifier, image, frame_size,
tolerance)
overlay_predictions(image, terrain_prediction, frame_size,
f'output_images/{image_name}.jpeg")

# Побудова та збереження розподілу впевненості класів
plot_class_confidences(classifier.predict_proba(X), tolerance,
"class_confidence_distribution.png")

# Побудова графіків критерію Кульбака
delta_range = np.linspace(0.1, 0.5, 10)
plot_kullback_vs_delta(classifier, X, y, delta_range, "kullback_vs_delta.png")

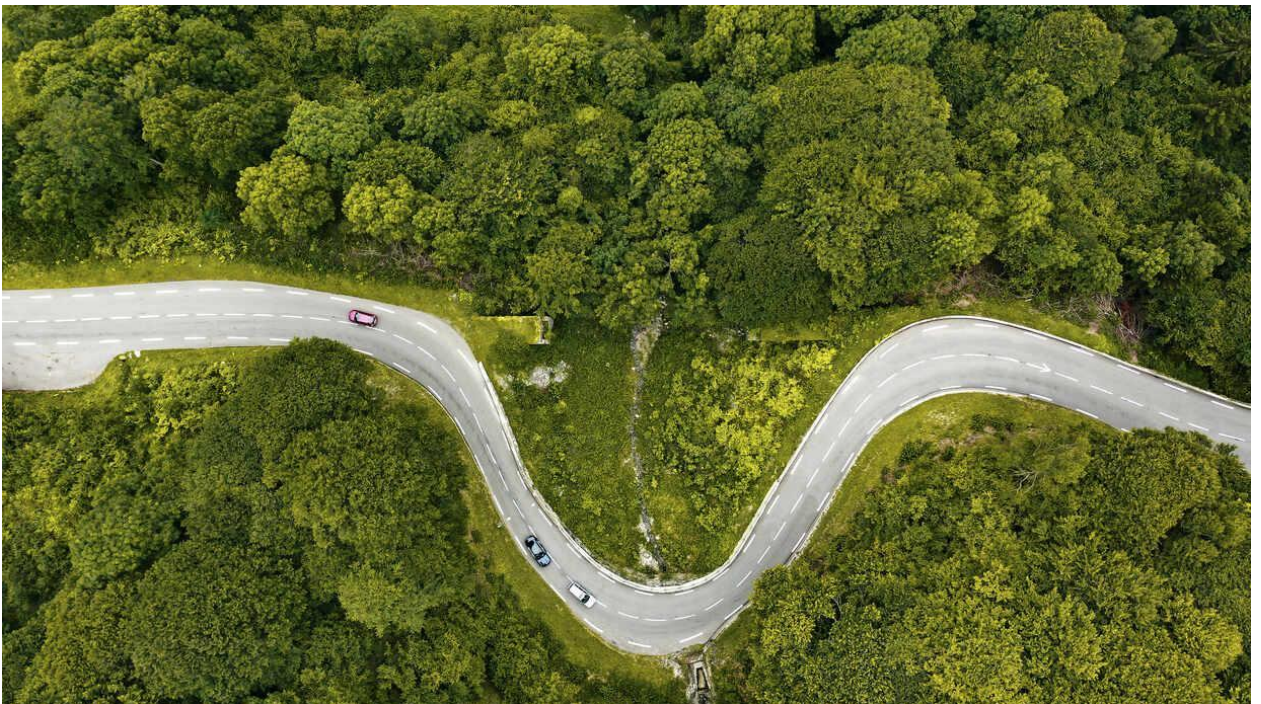
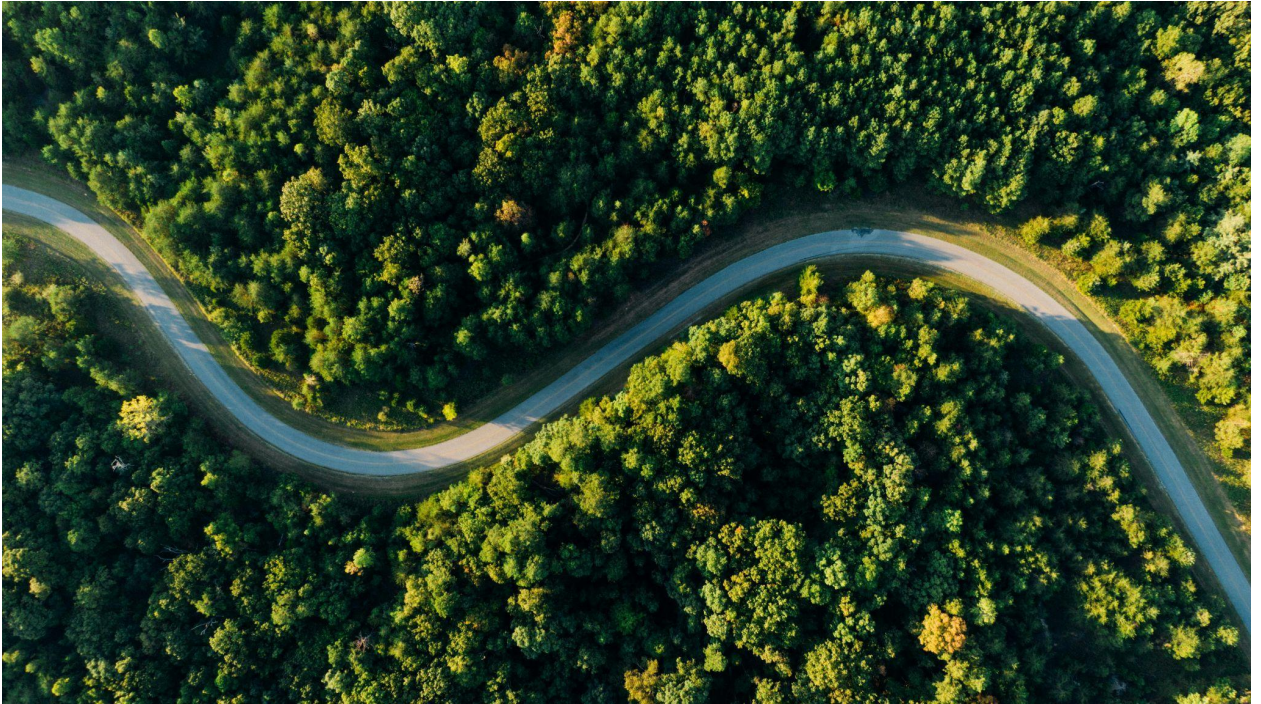
plot_kullback_vs_radius(X, y, target_class=0, class_name="Forest",
save_path="kullback_vs_radius_forest.png")
plot_kullback_vs_radius(X, y, target_class=2, class_name="Road",
save_path="kullback_vs_radius_road.png")

# Побудова декурсивного бінарного дерева
tree_classifier = recursive_binary_tree_classification(X, y, max_depth=3)

```

ДОДАТОК В

Вхідні фото





ДОДАТОК Г

Вихідні фото

