

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Кваліфікаційна робота магістра

**МОДЕЛІ ТА МЕТОДИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ МАШИННОГО
НАВЧАННЯ БОРТОВОЇ СИСТЕМИ БЕЗПЛОТНОГО ЛІТАЛЬНОГО
АПАРАТУ З ВИКОРИСТАННЯМ СИМУЛЯТОРА**

Здобувач вищої освіти гр. ІН.м-13

Олександр ГРИНЕНКО

Науковий керівник:
доктор технічних наук, професор,
професор кафедри комп'ютерних наук

Анатолій ДОВБИШ

В.о. завідувача кафедри
кандидат технічних наук, доцент,
доцент кафедри комп'ютерних наук

Ігор ШЕЛЕХОВ

СУМИ 2022

Сумський державний університет

Факультет ЕЛІТ Кафедра Комп'ютерних наук
 Спеціальність «122 - Комп'ютерні науки»

Затверджую:

В.о. зав.кафедри _____

“ _____ ” _____ 20__ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Гриненку Олександрю Валерійовичу

1. Тема роботи Моделі та методи інформаційної технології машинного навчання бортової системи безпілотного літального апарату з використанням симулятора

затверджую наказом по СумДУ від «___» _____ 20__ р. № _____

2. Термін задачі здобувачем кваліфікаційної роботи _____

3. Вхідні дані до кваліфікаційної роботи _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Вступ; 2) Інформаційний огляд; 3) Постановка завдання; 4) Вибір методу рішення;
5) Інформаційне та програмне забезпечення; 6) Висновок; 7) Література; 8) Додаток.

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

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

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання «___» _____ 20__ р. Керівник _____

Завдання прийняв до виконання _____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми та постановка задачі</i>		
2	<i>Вибір методу рішення</i>		
3	<i>Інформаційне та програмне забезпечення</i>		
4	<i>Оформлення кваліфікаційної магістерської роботи</i>		

Здобувач вищої освіти _____

Науковий керівник _____

РЕФЕРАТ

Записка: 71 стор., 36 рис., 3 табл., 1 додаток, 21 джерел.

Об'єкт дослідження – слабоформалізований процес розпізнавання об'єктів на місцевості.

Мета роботи - підвищення функціональної ефективності бортової системи безпілотного авіаційного комплексу для розпізнавання об'єктів в змодельованому середовищі.

Методи дослідження — детерміновано-статистичні методи розпізнавання образів, математичний аналіз вхідних даних, інформаційно-екстремальна інтелектуальна технологія.

Результати – розроблено програмне забезпечення, яке дозволяє формувати вхідний інформаційний опис та перевіряти, у рамках симуляції, створені інтелектуальні системи. Під час роботи було відтворено процес функціонування безпілотного літального засобу з бортовою системою розпізнавання наземних об'єктів. Був проведений експеримент, за результатами якого був сформульований новий підхід до ідентифікації кадрів зображення у рамках обраної технології.

ЗМІСТ

ВСТУП.....	8
1. ІНФОРМАЦІЙНИЙ ОГЛЯД НАУКОВО-ТЕХНІЧНОЇ ПРОБЛЕМАТИКИ ГАЛУЗІ.....	9
1.1. Сучасний стан розвитку автономних безпілотних літальних апаратів.....	9
1.2. Постановка задачі.....	27
2. ОПИС МЕТОДУ ДОСЛІДЖЕННЯ.....	28
2.1. Основні принципи та положення інформаційно-екстремальної інтелектуальної технології аналізу даних.....	28
2.2. Основні властивості автономної бортової системи розпізнавання, що навчається.....	3
2	
2.3. Інформаційні критерії оптимізації параметрів машинного навчання.....	34
2.4. Базовий алгоритм інформаційно-екстремального машинного навчання автономної бортової системи відеонавігації.....	40
3. ІНФОРМАЦІЙНЕ. АЛГОРИТМІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ АВТОНОМНОЇ БОРТОВОЇ СИСТЕМИ РОЗПІЗАВАННЯ НАЗЕМНИХ ОБ'ЄКТІВ.....	46
3.1. Формування вхідного інформаційного опису автономної бортової системи розпізнавання наземних об'єктів.....	46
3.2. Вхідний математичний опис бортової системи розпізнавання.....	55
3.3. Результати програмного моделювання.....	57
3.4. Короткий опис програмної реалізації.....	63
ВИСНОВКИ.....	64
СПИСОК ЛІТЕРАТУРИ.....	65
ДОДАТОК.....	68

ВСТУП

Основним недоліком сучасних безпілотних літальних апаратів є їх повна або часткова залежність від командного пункту з яким ті встановлюють зв'язок по радіоелектронному каналу зв'язку. Незважаючи на різноманітність практичних задач, які здатні виконувати безпілотники, у більшості випадках вони виступають у ролі ретранслятору зображення місцевості. Такий підхід може бути виправданим для цивільного використання, але не для застосування їх у військовій галузі. Проблема в тому, що залежність пристрою від командного пункту або системи GPS робить його вразливим до радіоелектронної протидії. Використання якої може повністю дезорієнтувати безпілотник за декілька секунд.

Одним із можливих, але складних, варіантів вирішення цієї проблеми є створення автономної бортової системи розпізнавання наземних об'єктів на основі моделей та методів машинного навчання. Насамперед задача ускладнюється існуючими науково-методологічними проблемами інтелектуальних систем. До них можна віднести модальний характер більшості досліджень, та незавершеність теорії ефективного машинного навчання. Пропонується для розробки автономної БСР використовувати симулятор, який дозволить відтворити, максимально наближені до реальних, умови функціонування безпілотника. Це необхідно для ефективного навчання та формування вхідного математичного опису бортової системи розпізнавання образів. Такий підхід дозволить розробляти та одразу перевіряти функціональну ефективність інтелектуальних систем за допомогою комп'ютерної симуляції, зокрема, в умовах, які відрізняються від типових.

У рамках роботи буде імітуватися процес функціонування безпілотника при втручанні довільних зовнішніх факторів, які впливають на точність розпізнавання наземних об'єктів

1. ІНФОРМАЦІЙНИЙ ОГЛЯД НАУКОВО-ТЕХНІЧНОЇ ПРОБЛЕМАТИКИ ГАЛУЗІ

1.1. Сучасний стан розвитку автономних безпілотних літальних апаратів.

Не зважаючи на активний технологічний розвиток створити в повній мірі автономну інтелектуальну систему (ІС), на даний момент часу, не можливо. У сучасному науковому дискурсі автономність – це більше перспективний напрям досліджень, а ніж реальна фізична розробка. Створити щось, що здатне перевершити за своїми когнітивними можливостями людський мозок, поки що, не вдалося. Тому під терміном «автономність», як правило, розуміється реалізація певної інтелектуальної компоненти, яка здатна приймати рішення у рамках чітко визначеної задачі. Наприклад, розпізнавання рукописного тексту, автокерування транспортом, прогнозування погоди, створення картин тощо.

Сучасні безпілотні літальні апарати (БПЛА), як не дивно, здатні виконувати ряд задач автономно. Цікаво, що саме для безпілотників подібні інтелектуальні системи розробляються чи не найактивніше. Це обумовлено тим, що БПЛА набули широкого застосування в багатьох економічних галузях, де вони здатні виконувати різноманітний спектр задач:

- *Логістика.* Хоч БПЛА поступаються вантажівкам, які здатні перевозити багатотонний вантаж на відстані понад тисячі кілометрів, але, тим не менш, безпілотники мають одну суттєву перевагу. Вони пересуваються у повітряному просторі, що надає їм можливості до обрання довільного маршруту, а це, в свою чергу, дозволяє здійснювати доставку майже у будь-яке місце. Більше того при наявності GPS на борту БПЛА можна розробити автономну систему, що здатна переміщуватися з точки А до точки Б. Таким чином, наприклад компанія Amazon, у рамках своєї програми «Amazon Prime Air» [9], реалізувала автоматичну квадрокоптерну доставку своїх товарів у межах більшості великих міст США та Європи;

- **Сільське господарство.** Специфікою цієї сфери є контроль над великою площею території. У класичному вигляді для цього використовуються мобільні групи спеціалістів, які виїжджають на об'єкти для здійснення агроробіт та специфічного аналізу місцевості. У більшості таких випадках цей процес можна автоматизувати за рахунок використання БПЛА, які здатні, наприклад, виконувати автоматичний контроль над великими площами території. Класифікувати зображення та виявляти ділянки, що зазнали засухи або затоплення. Більше того у такий спосіб можна здійснювати комплексний моніторинг сільськогосподарських угідь, що надає можливість прогнозування врожайності та запобігати знищенню посадженої агрокультури [10];
- **Електроенергетика.** Мережа ліній електропередачі (ЛЕП) відноситься до критичної інфраструктури, при цьому важливо розуміти, що вона охоплює собою площу усєї країни. Це означає, що адекватна ревізія за допомогою виїзної групи спеціалістів достатньо довга і трудомістка задача, яку можна оптимізувати, а у деяких випадках і повністю автоматизувати, за рахунок використання БПЛА. Адже безпілотник без проблем здатний з повітря здійснювати перевірку неперервної інфраструктурної мережі і робити це значно швидше ніж людина. Окрім цього, при використанні теплових камер є можливість одночасно виконувати й аналіз тепломереж з виявленням можливих дефектів на ранніх стадіях;
- **Нафтогазова промисловість.** Переважна більшість нафтогазових родовищ знаходяться в важкодоступних місцях, наприклад, в океані. Використання безпілотників в цій економічній сфері дозволяє, по-перше, виконувати моніторинг як самого родовища, так й прилеглої інфраструктури по транспортуванню нафти або газу. По-друге, зменшує штат спеціалістів, які повинні постійно знаходитися на об'єкті, але разом з цим підвищується їх безпека, бо, як зазначалося раніше, подібні виробництва розташовані в важкодоступних місцях, які є не найбезпечнішими для людини;

- **Екологія.** Сфера інтересу, де безпілотники виконують надзвичайно велику різноманітність задач. Наприклад, контроль лісового фонду [11], через аналіз великої території та вчасного виявлення не законної вирубки лісу. У такий же спосіб вдається прогнозувати можливі пожежі, що актуально для субтропічних та тропічних регіонів. Також БПЛА використовують для моніторингу популяцій рідкісних тварин у Африці. Окрім цього можна згадати, що у Китаї впроваджені безпілотники, які здійснюють атмосферний аналіз у межах мегаполісів, це дозволяє виявляти виробництва, які порушують обмеження по викидам шкідливих речовин у повітря [12].

Звісно, що це далеко не повний список сфер використання БПЛА, але цього вже достатньо для уявлення варіантів використання безпілотників. Тим не менш, усі зазначені галузі мають цивільне спрямування, бо воєнне технічне оснащення та їх задачі необхідно виділити окремо з більш конкретизованим оглядом функціональних можливостей. Для опису інтелектуальних компонент воєнних БПЛА варто зазначити і їх технічні особливості, адже вони на пряму впливають як на рід діяльності безпілотника, так й на його програмне забезпечення.

Нещодавно Повітряні сили США сумісно з приватною компанією “Kratos” [13] завершили випробування своєї спільної розробки. Реактивного безпілотного літального апарату на базі літака «XQ-58A Valkyrie» (рис.1.1). Із відкритих джерел відомо, що БПЛА є частково автономним, як мінімум, було зазначено серед специфічних функцій: можливість автоповернення цього габаритного пристрою до точки зльоту. Теоретично це робить його стійким до впливу засобів радіоелектронної боротьби (РЕБ) зі сторони ворога, адже навіть при втрачанні зв'язку з центром керування безпілотник все рівно здатний повернутися на базу. Першу партію планується виготовити вже в 2023 році.



Рисунок 1.1. БПЛА «XQ-58A Valkyrie»

Якщо не брати до уваги закриті розробки деяких країн, наприклад американський MQ-9 «REAPER», то серед добре відомих та розповсюджених воєнних БПЛА виділяється «Bayraktar Mini-UAV» (рис.1.2) від компанії «Baykar» [14]. Для автоматичної навігації він може використовувати опорні точки, які прив'язані до геоінформаційних орієнтирів на місцевості, а для запобігання РЕП застосовуються алгоритми автоповернення та згадана автонавігація. У рамках бойових дій цей БПЛА продемонстрував точність визначення цілі майже до 10 метрів, що є надзвичайним результатом враховуючи висоту його польоту. При цьому це розробка приватної турецької компанії, тому про його технічні та програмні особливості відомо дещо більше ніж повністю.



Рисунок 1.2. Процес запуску Bayraktar «Mini UAV»

Серед представників воєнних БПЛА є і повністю автономні, наприклад, дрон-камікадзе «STM Kargu» (рис.1.3) [15] турецького виробництва. Він оснащений бортовою системою розпізнавання (БСР) зображень в реальному часі. За допомогою машинного навчання безпілотник може класифікувати як статичну так й рухому ціль. У даному випадку під автономністю розуміється повна незалежність пристрою від людини, але це породжує ряд морально-етичних питань [16]. Так, наприклад, згідно з доповіддю групи експертів по Лівії при ООН, цей дрон в 2020 році сам без наказу ліквідував людину під час бойових дій, що призвело до обговорення цієї проблеми у 2021 році [21]. Не дивлячись на це, компанія STM розглядає можливість поставок апарату трьом неназваним країнам, а теперішніми користувачами є Азербайджан та Перу.



Рисунок 1.3. STM «Kargu»

Серед вітчизняних безпілотників можна відмітити Raybird-3 (рис 1.4) [17]. Розроблений в 2016 році компанією «Скаетон» апарат керується оператором з командного пункту, який покриває радіус у 250 км. БПЛА здатний пролетіти автономно до 2.5 тис. кілометрів, але заздалегідь прокладеним маршрутом. Комплекс керування здатний вести безліч апаратів, а керувати безпілотником може безліч станцій. Таким чином на полі бою вдасться створити комунікаційну мережу в межах якої БПЛА зможе вільно пересуватися.



Рисунок 1.4. Український «Raybird-3» на станції запуску

Усі технічні характеристики розглянутих БПЛА зазначені в таблиці:

Таблиця 1.1. Порівняння технічних характеристик БПЛА

Назва апарату	Розмах крила	Макс. висота польоту	Тривалість польоту	Крейсерська швидкість	Макс. злітна маса	Радіус польоту
XQ-58A Valkyrie	6700мм	13км	-	882 км/год	2.7 кг	3.9 км
STM Kargu	600мм	2.8 км.	30 хв.	72 км/год	7.06 кг	5 км.
Bayraktar Mini-UAV (Б)	1900мм	3.6 км.	> 1 год.	70 км/год	5 кг	15 км.
Raybird-3	2960мм	3.5 км.	18-28 год.	110 км/год	23 кг.	1 тис. км.

Воєнна галузь – є надзвичайно специфічною та регламентованою. Це створює ряд не типових умов для роботи ІС. При цьому технічно-програмне оснащення повинно задовольняти не тривіальні потреби воєнних. Наприклад, виникає необхідність у створенні захищеного каналу зв'язку між БПЛА та пунктом керування [18], але який не повинний впливати на оперативність роботи. Адже воєнні безпілотники, в переважній більшості, здійснюють обробку

отриманої інформації у реальному часі. Із-за цього від ІС вимагається не лише забезпечити оперативність прийняття рішення, а ще й бути достатньо адаптованими до довільних змін на місцевості. Цього можна досягти реалізацією двох компонент:

- 1) інваріантність вирішальних правил інтелектуальної системи, що, теоретично, дозволить підвищити правильність прийняття рішень;
- 2) перенавчання інтелектуальної системи, що дозволить адаптуватися до змін;

Інтелектуальне програмне забезпечення сучасних безпілотників можливо умовно розділити на наступні кластери: т

- **Псевдо інтелектуальні системи.** До цієї групи слід віднести усі алгоритми обробки зображення, які дозволяють, наприклад, виділити об'єкт або його контури, але не ідентифікувати його. Сюди ж варто віднести різного роду попередження про набрання максимально можливої висоти, ймовірне скоре втрачання зв'язку з пристроєм . По своєму призначенню це методи, які спрощують роботу оператора, але самі не приймають жодних рішень.
- **Автономні компоненти.** Це реалізація деяких складових автономної системи. Наприклад, автостабілізація, адаптація до зміни погодних умов, автоматична посадка або зліт пристрою тощо. Інакше кажучи увесь функціонал, який є невід'ємною частиною БПЛА і забезпечує автономне використання деяких його технічних можливостей.
- **Автономні системи.** Найбільш цікава і при цьому найменш формалізована група. До неї варто включити усі алгоритми прийняття рішень реалізовані при використанні БПЛА. Тобто, це не обов'язково повинні бути повністю автономні пристрої, але, як мінімум, здатні за наявної інформації робити деякі висновки, а при необхідності виконувати роботу самостійно. Інакше кажучи, це комплекс реалізацій, який спрямований для автоматизації дій БПЛА або з використанням БПЛА.

Наприклад, автокорегування артилерійського вогню, трекінг та фокус на цілі, автоповернення, автоматична класифікація об'єктів в кадрі, прогнозування за наявною геоінформацією тощо. Тим не менш, як зазначалося раніше, в повній мірі автономних систем не існує, тому цю групу варто розглядати саме як комплекс реалізацій, які спрямовані для виконання конкретної задачі.

Як було зазначено в минулому розділі, БПЛА набули широкого використання у різних сферах людської діяльності. Проте, що стосується алгоритмічного забезпечення, то воно, в переважній більшості, може мати далеко не одне призначення. Тому ІС, що використовуються в одній економічній галузі можуть бути адаптовані й до іншої. Найважливіше тут концепт та використані моделі машинного навчання.

Невід'ємною частиною будь-якого БПЛА є етап зльоту та посадки. Якщо річ йде про малогабаритні пристрої типу квадрокоптер, то тут все відносно легко, але цього не можна сказати про безпілотники типу планер. У роботі [2] автори розглядають проблему автоматичної посадки Saker 4 UAV (рис. 1.5) з наступними технічними властивостями: вага – 25 кг, розмах крила – 3.75 м., максимальна швидкість польоту – 120 км/год, максимальна висота – 5 км.



Рисунок 1.5. Планер Saker 4 UAV

У запропонованій реалізації [2] потокове відео розбивається на упорядкований набір фреймів на кожному з яких знаходять, так звану, зону

інтересу (ЗІ), у даному випадку – це злітно-посадкова смуга. Тим не меш, цього не достатньо для здійснення посадки, адже розпізнані ділянки включають в себе зайві об'єкти, які заважають скоректувати приземлення. Тому, після класифікації ЗІ, виконується додатковий етап обробки зображення з пошуком країв об'єкту, що дозволить вірно розрахувати положення злітно-посадкової смуги для автоматичної посадки БПЛА. Схематично увесь процес зображено на рисунку 1.6.

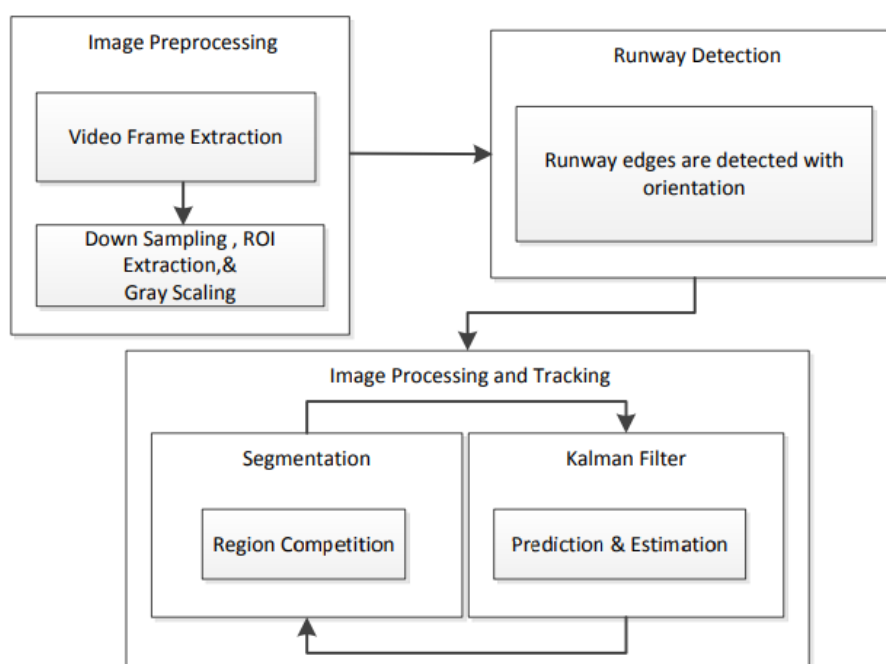


Рисунок 1.6. Схематичне зображення процесу функціонування БСР

Щоб досягти бажаного результату використовується деяка мета інформація про задачу. Річ у тім, що будь-яку злітно-посадкову смугу можна описати двома паралельними лініями. Проте зображення з борту БПЛА ми отримуємо з деякою деформацією перспективи, тому подібна апроксимація не можлива. Тим не менш, нам відомо, що ЗІ добре контрастує з оточенням, таким чином задача машинного навчання зводиться до пошуку таких відрізків $d_i, i = 1, \dots, 4$, які описують мінімальну кількість інформації поза зоною інтересу, візуально це показано на рисунку 1.7.

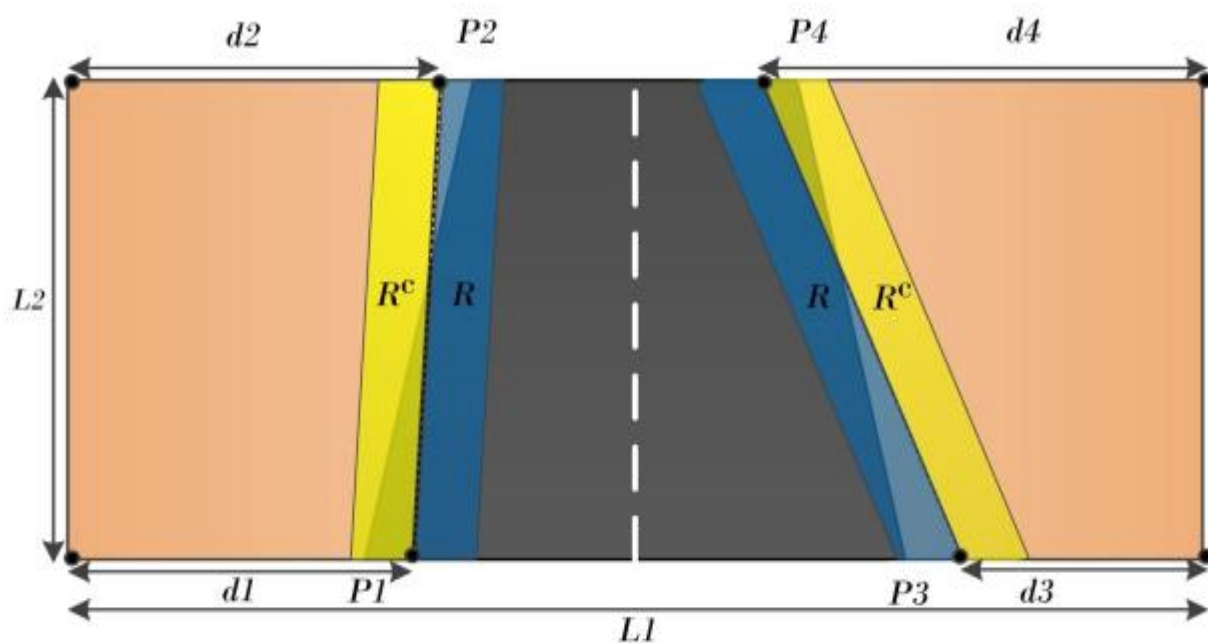


Рисунок 1.7. Апроксимація злітно-посадкової смуги

Формалізовано задачу оптимізації країв об'єкту можна подати наступним чином. Припустимо, що дві сусідні ділянки R та R^c однакові за площею. Необхідно знайти таке їх положення, щоб вони мали максимально різне значення усередненої яскравості. Нехай, зображення буде позначатися $I: \Omega \rightarrow \mathfrak{R}^k$, де $k = 3$ для кожної компоненти RGB. У свою чергу $R, R^c \subset \Omega$ і визначенні як сусідні регіони (рис.1.7). Усереднене значення яскравості будемо позначати як $u, v \in \mathfrak{R}^k$, тоді

$$u = \frac{1}{|R|} \int_R I(x) dA, \quad v = \frac{1}{|R^c|} \int_{R^c} I(x) dA, \quad (1.1)$$

де dA - елемент обраної області.

Для визначення оптимальної позиції відрізків необхідно мінімізувати наступну цільову функцію:

$$E(d_1, d_2, d_3, d_4) = -\frac{1}{2} |u - v|^2 \quad (1.2)$$

Звісно, що є класичні методи визначення країв об'єкту, які базуються на обчисленні часткових похідних, але вони дуже чутливі, до шумів та положення об'єкту у кадрі. Як зазначалося раніше специфіка розробленої ІС передбачає деяку деформацію початкового зображення, тому потрібно або забезпечити інваріантність вирішальних правил, як це зробили автори роботи [2], або виконувати афінні перетворення.

Другою, не менш важливою, специфікою роботи БСР є те, що прийняття рішень повинно відбуватися у реальному часі. Тому потрібно забезпечити систему достатнім рівнем оперативності. Для цього БПЛА починає корегувати свою посадку, коли він вже знає, що наближається до злітно-посадкової смуги, бо та є в зоні видимості його камери або був отриманий відповідний сигнал від пункту керування. Інакше кажучи ЗІ знаходиться в зоні досяжності БПЛА. Знову ж таки, із-за специфіки розміщення камери та особливостей злітно-посадкової смуги можна обмежити можливі варіанти розташування ЗІ, відповідно й відрізків d_i , $i = 1, \dots, 4$. Припускається, що вона буде знаходитися в деякому діапазоні $[-\theta_{min}, \theta_{max}]$.

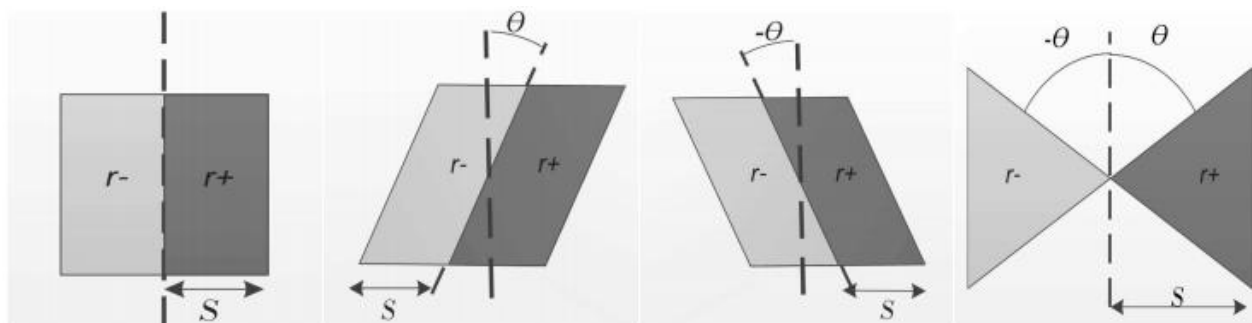


Рисунок 1.8. Можливі варіанти розміщення країв об'єкту

Згідно рисунку 1.8., пряма, що описує край ЗІ може бути відхилена відносно центру рецепторного поля не більше ніж на $[-\theta_{min}, \theta_{max}]$, $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, тоді (2) прийме наступний вигляд

$$E'(x, \theta) = -\frac{1}{2} (u'(x, \theta) - v'(x, \theta))^2 \quad (1.3)$$

На рисунку 1.9. показаний приклад визначення країв об'єкту інтересу з графіком залежності цільової функції (1.3).

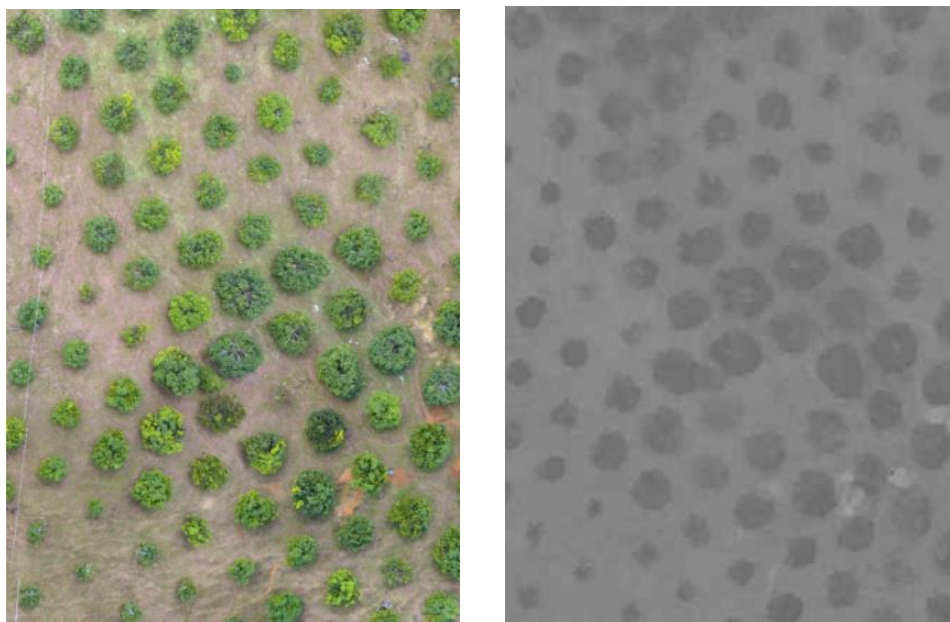


Рисунок 1.9. Обрання найкращого кандидата

Звісно, що це далеко не увесь алгоритм БСР, бо як зазначалося раніше автономні системи – це комплекс як програмних, так й технічних компонент. Як мінімум, у своїй роботі [2] автори використовують ще декілька кванторів оптимізації для підвищення інваріативності розробленої моделі при розпізнаванні злітно-посадкової смуги. А також зазначають усі технічні можливості використаного ними БПЛА, бо саме вони впливають на вибір алгоритмів машинного навчання.

Перейдемо до іншої економічної галузі, а саме агрономії. На прикладі роботи [3] показаний варіант використання БПЛА для ідентифікації об'єктів інтересу та визначення найближчих сусідів між ними. Практична цінність розробки полягає у тому, що безпілотник може самостійно виконувати контроль процесу зростання агрокультури та визначати її стадію росту. Автори [3] розглядають це питання на прикладі плантації мангових дерев.

Як основний інструмент була використана безкоштовна бібліотека аналізу зображень OpenCV. На першому кроці був обраний найкращий кольоровий простір з усіх можливих. Необхідно, щоб при його використанні мангові дерева максимально сильно контрастували з фоном. Для отриманих зображень найкраще підійшов YCrCb.



а

б

Рисунок 1.10. Результат переходу від RGB до YCrCb: а – плантація в RGB спектрі; б – плантація в YCrCb;

Наступним кроком є сегментація зображення з подальшим виділенням крон дерев. Для цього був використаний метод адаптивного гаусівського порогу, для Cr спектру. Принцип роботи алгоритму достатньо простий, він виокремлює найтемніші ділянки зображення від фону. Для цього використовується деяке порогове значення пікселя, яке визначається як зважена сума його сусідів. Якщо яскравість менша порогу, то припускаємо, що цей піксель належить дереву. Для позбавлення шуму був використаний фільтр ерозія з бібліотеки OpenCV, який зменшує область світлого зображення. Тим самим виділяючи і збільшуючи темні ділянки.

На цьому етапі, використавши лише стандартні методи обробки зображення, вже вдалося сегментувати об'єкти. У нашому випадку це мангові дерева. Звісно, що система не здатна класифікувати результат сегментації і визначити, що саме вона бачить, але у рамках цього завдання це і не потрібно. Задачею БПЛА є контроль над ростом мангових дерев при заданих і не змінних умовах функціонування БСР.

Наступним кроком потрібно визначити центри сегментованих об'єктів та поєднати їх в одну неперервну систему. Звісно, що якщо ми це зробимо напряду, то отримаємо результат, де кожен поєднаний з кожним. Тому нам потрібно оптимізувати цю карту, щоб об'єкти поєднувалися лише зі своїми найближчими сусідами. По-перше, слід формалізувати кожне сегментоване дерево. Нехай, усі вони будуть замінені колом з радіусом 30 пікселів для простоти подальшої роботи і візуального сприйняття, бо початково усі об'єкти мають довільні розміри. По-друге, центри кіл потрібно поєднати між собою у неперервну мережу. Для цього було використано перетворення Хафа, яке шукає точки, що знаходяться на одній умовній лінії та поєднує їх. Визначаємо, що на одній лінії може бути до 5 об'єктів, а відстані між ними можуть досягати понад 400 пікселів. Результатом роботи алгоритму є наступна карта (рис.1.11).

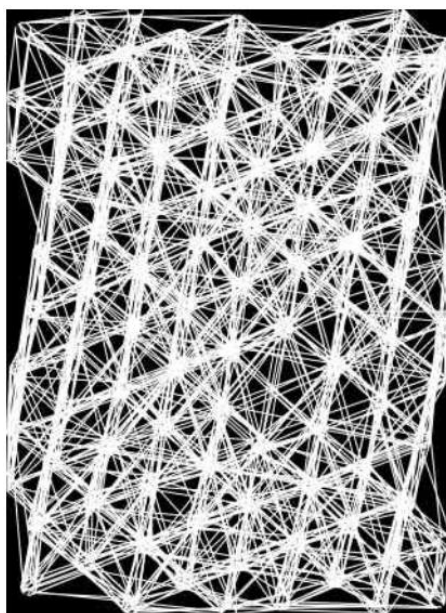


Рисунок 1.11. Результат роботи перетворення Хафа

Як видно з рисунку алгоритм знайшов занадто багато зв'язків для кожного з об'єктів, необхідно мінімізувати їх кількість. Для кожної точки окремо усі лінії об'єднують у групи, а потім усереднюють їх кут. Таким чином отримується результуюче зображення (рис. 1.12).

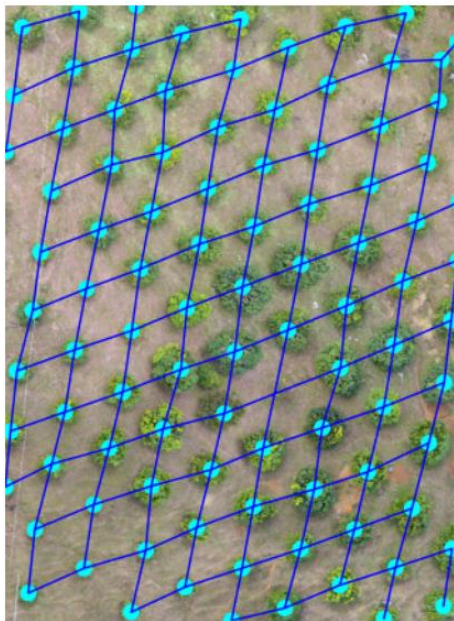


Рисунок 1.12. Неперервна мережа із мангових дерев

Важливо в котре відмітити, що та чи інша реалізація ІС залежить насамперед від специфіки вхідних даних. Універсального рішення для кожної подібної задачі не існує. У даному випадку вдалося сегментувати об'єкти на зображенні та поєднати їх в одну неперервну мережу. В цілому, для цієї конкретної задачі цього достатньо. При цьому в подальшому, маючи результати сегментації за тривалий час, система може слідкувати за перебігом росту мангових дерев. Наприклад, через фіксування збільшення або зменшення об'єкту з часом.

Розглянемо ще один приклад використання БПЛА у задачі розпізнавання образів [4]. На цей раз завдання полягає у моніторингу позицій сільськогосподарської худоби на відкритій місцевості. Звісно, що можна повісити на кожну з тварин GPS трекер, який буде записувати і відправляти свої координати через деякі інтервали часу. Тим не менш, використання дрону для цих цілей дасть більш комплексну інформацію, при цьому з меншими зусиллями. Але, що більш важливо, такий підхід є набагато універсальнішим. У подальшому таку розробку вдасться адаптувати, наприклад, для задач моніторингу популяції диких тварин.

Автори пропонують аналізувати зображення комплексно, через призму різних кольорових просторів. Таких як: RGB, HSV, YCrCb, HSL та LAB. Річ у тім, що при використанні лише RGB на практиці з'явився ряд не очевидних проблем. Наприклад, не однорідність кольору рослин. Тобто була присутня варіативність вхідного математичного опису. Тварини могли зливатися з сухою травою або з тіннями. Тому для вирішення цього питання розробниками була використана деяка мета інформація про умови роботи БПЛА. Був доданий користувацький інтерфейс, де людина обирала такий кольоровий простір на якому тварини максимально сильно контрастували з фоном. Далі система автоматично підбирає поріг яскравості, щоб бінаризувати зображення. Останнім кроком є використання морфологічних операцій ерозія та дилатація.

Варто пояснити, що для двох множин A і B з простору \mathbb{R}^2 ерозія A по структурному елементу B визначається як

$$A \ominus B = \{z \in E | B_z \subseteq A\}$$

Інакше кажучи, ерозія множини A по примітиву B , це таке геометричне місце точок для всіх таких позицій центру z , при зсуві яких множина B цілком міститься в A .

У свою чергу дилатація множини A по множині B визначається як

$$A \oplus B = \{z \in E | (B^s)_z \cap A \neq \emptyset\}$$

При цьому дилатація A по B це множина всіх таких переміщень z , при яких A і B збігаються принаймні в одному елементі.

Всі ці налаштування виконує оператор, таким чином він адаптує БСР до конкретних умов функціонування. В результаті ми отримуємо максимально достовірні зони інтересу, які у рамках роботи [4] представлені як домашні кози. Але навіть так отримані ЗІ можуть містити хибні об'єкти, щоб перевірити це

автори як раз пропонують комплексно оцінити зображення. Для цього система порівнює такі характеристики: середнє значення та середнє відхилення, стандартне відхилення діагональних колірних каналів для RGB, LAB, HSV та YCrCb; окремо локальні бінарні візерунки та гістограму орієнтованих градієнтів для HSV та YCrCb. За допомогою цих характеристик було навчено десять моделей, таблиця 1.2.

Таблиця 1.2 Список використаних моделей машинного навчання

Абривіатура	Повна назва	Абривіатура	Повна назва
ADA	Ada Boost Classifier	CART	Decision Tree Classifier
LDA	Liner Discriminant Analysis	XT	Extra Trees Classifier
RF	Random Forest Classifier	NB	Gaussian Naïve Bayes
QDA	Quadratic Discriminant Analysis	MLPC	Multiple Layer Perceptron Classifier
Grad	Gradient Boosting Classifier		

При цьому датасети були підібрані таким чином, щоб містити не тільки зону інтересу, а й інші об'єкти, які можуть знаходитися в ЗІ. Точність класифікації перевірялася за наступною формулою:

$$\frac{TP + TN}{TP + TN + FP + FN}$$

де TP – істинно позитивні класифікації; TN – істинно негативні класифікації; FP – помилково позитивні класифікації (помилка I-го роду); FN – помилково негативні класифікації (помилка II-го роду);

Автори [4] зазначають, що для тестування своєї системи вони зібрали вибірку з 571 зображення з борту БПЛА, які були зроблені протягом 11 днів. Кінцева модель здатна вірно класифікувати тварин з 74% точністю і майже у 79% розпізнавати рух. Схематично увесь процес роботи БСР показано на рисунку 1.13.

Тому при розробці БСР, а також будь-якої іншої ІС, потрібно використовувати максимальну кількість мета інформації про об'єкт дослідження. Це дозволить як найкраще підійти до вирішення чітко визначеної задачі.

1.2. Постановка задачі.

Нехай є чітко заданий сценарій функціонування безпілотного літального апарату. При незмінних умовах побудувати набір вирішальних правил, які повинні реалізувати сегментацію зображення регіону, що спостерігається, на техногенні та природні об'єкти. Оскільки при своїй роботі безпілотник зазнає впливу зовнішніх факторів, які змінюють його положення, то необхідно забезпечити вирішальні правила інваріантністю до довільних початкових умов формування зображень наземних об'єктів. Для цього пропонується наділити БСР модулем, який комплексно оцінює точність розпізнавання об'єкту інтересу і якщо вона стає меншою деякого порогового значення, то необхідно адаптувати вхідний математичний опис.

Для розв'язку поставленої задачі потрібно виконати наступні завдання:

- реалізувати алгоритми машинного навчання БСР з оптимізацією системи контрольних допусків на ознаки розпізнавання;
- побудувати вирішальні правила в просторі ознак класів розпізнавання;
- побудувати 3D-модель місцевості для формування вхідного математичного опису БСР;
- реалізувати алгоритм функціонування системи в режимі екзамену і оцінити його функціональну ефективність;

У рамках магістерської роботи буде розглядатися питання забезпечення достатньої достовірності розпізнавання образів на борту безпілотника при неконтрольованій зміні інформаційного опису. Для перевірки ефективності розробленого алгоритму буде використана 3D-модель місцевості. У такий спосіб вдасться поставити експеримент функціонування БСР в умовах, які максимально наближені до реальних.

2. ОПИС МЕТОДУ ДОСЛІДЖЕННЯ

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

Згідно з працею [1] основна ідея машинного навчання в рамках ІЕІ-технології аналізу даних полягає в трансформації апріорного у загальному випадку нечіткого розбиття простору ознак у чітке розбиття класів розпізнавання способом оптимізації параметрів функціонування системи. Водночас здійснюється цілеспрямований пошук глобального максимуму багато екстремальної функції статистичного інформаційного критерію в робочій (допустимій) області її визначення з одночасним відновленням оптимальних роздільних гіперповерхонь, що відновлюються в радіальному базисі бінарного простору ознак розпізнавання.

Методи інформаційно-екстремального машинного навчання базуються, крім відомих принципів системного аналізу, на таких специфічних принципах:

- максимізації інформації, обґрунтованому екстремальністю сенсорного сприйняття образу, що експериментально доведено вченими-фізіологами. Цей принцип реалізують способом введення додаткових інформаційних обмежень, що збільшують різноманітність класифікованих об'єктів;

- дуальності, що полягає в реалізації на етапі апріорного моделювання простих алгоритмів за умови їх цілеспрямованого уточнення способом поглиблення машинного навчання для наближення вирішальних правил до безпомилкових за навчальною матрицею;

- апріорної недостатності обґрунтування гіпотез (принцип Бернуллі – Лапласа), згідно з яким за умов апріорної невизначеності даних доцільно розглядати апріорні гіпотези однаково ймовірними, тобто рішення приймаються системою за найгірших у статистичному розумінні умов;

- рандомізації вхідних даних, що дозволяє досліджувати детерміновано-статистичні характеристики процесу;

– редукції даних, що обумовлює необхідність оптимізації в інформаційному розумінні словника ознак розпізнавання способом видалення з нього неінформативних та ознак, які заважають;

1. відкладених рішень О. Г. Івахненка, що полягає в необхідності повторення процедур машинного навчання для досягнення мети побудови безпомилкових за навчальною матрицею вирішальних правил;

– зовішнього доповнення, що обґрунтовує необхідність використання навчальної або контрольної (екзаменаційної) вибірки для оцінювання функціональної ефективності машинного навчання

Вирішальні правила в процесі оптимізації параметрів машинного навчання в рамках ІЕІ-технології будують згідно з принципом відкладених рішень Івахненка О. Г. за багатоциклічною ітераційною процедурою пошуку максимального граничного значення усередненого за алфавітом класів розпізнавання інформаційного критерію оптимізації:

$$g_{\xi}^* = \arg \max_{G_{\xi}} \{ \max_{G_{\xi-1}} \{ \dots \{ \max_{G_1 \cap G_E} \frac{1}{M} \sum_{m=1}^M E_m \} \dots \} \}, \quad (2.1)$$

де E_m – інформаційний критерій оптимізації параметрів навчання системи розпізнавати реалізації класу X_m^o ;

G_{ξ} – допустима область значень ξ -ї ознаки розпізнавання;

G_E – допустима область визначення функції інформаційного критерію оптимізації параметрів машинного навчання.

Водночас на алгоритм інформаційно-екстремального машинного навчання (2.1) накладають обмеження:

$$(\forall X_m^o \in \tilde{\mathfrak{R}}^{|M|}) [X_m^o \neq \emptyset], \quad (2.2)$$

де $\tilde{\mathfrak{R}}^{|M|}$ – розбиття простору ознак на класи розпізнавання, потужність якого дорівнює $\text{Card } \tilde{\mathfrak{R}} = M$;

$$(\exists X_k^o \in \tilde{\mathfrak{R}}^{|M|})(\exists X_l^o \in \tilde{\mathfrak{R}}^{|M|})[X_k^o \neq X_l^o \rightarrow X_k^o \cap X_l^o \neq \emptyset]; \quad (2.3)$$

$$(\forall X_k^o \in \tilde{\mathfrak{R}}^{|M|})(\forall X_l^o \in \tilde{\mathfrak{R}}^{|M|})[X_k^o \neq X_l^o \rightarrow \text{Ker}X_k^o \cap \text{Ker}X_l^o \neq \emptyset], \quad (2.4)$$

де $\text{Ker}X_k^o$ – ядро класу розпізнавання X_k^o ;

$\text{Ker} X_l^o$ – ядро класу розпізнавання вищого до класу розпізнавання X_k^o ;

$$(\forall X_k^o \in \tilde{\mathfrak{R}}^{|M|})(\forall X_l^o \in \tilde{\mathfrak{R}}^{|M|}) \left[X_k^o \neq X_l^o \rightarrow [(d_k^* < d(x_k \oplus x_l)) \& \& (d_l^* < d(x_k \oplus x_l))] \right], \quad (2.5)$$

де d_l^* – оптимальний радіус контейнера класу розпізнавання X_l^o ;

$d(x_k \oplus x_l)$ – кодова відстань між вектором x_k , усередненим за ансамблем векторів ознак класу розпізнавання X_k^o , і відповідним вектором x_l класу розпізнавання X_l^o ;

$$\bigcup_{X_m^o \in \tilde{\mathfrak{R}}} X_m^o \subseteq \Omega_B, \quad (2.6)$$

де Ω_B – бінарний простір Геммінга.

У виразах (2.2) – (2.6) прийнято $k \neq l$ і $k, l, m = \overline{1, M}$.

Глибина інформаційно-екстремального машинного навчання характеризується кількістю параметрів функціонування системи, що оптимізуються за інформаційним критерієм.

Основна ідея машинного навчання методами ІЕІ-технології, так само як і у ШНМ – адаптація вхідного математичного опису системи розпізнавання до максимальної повної ймовірності класифікаційних рішень. Принципова відмінність методів інформаційно-екстремального машинного навчання від нейроподібних структур полягає в тому, що їх розробляють у рамках функціонального підходу до моделювання когнітивних процесів, притаманних людині під час формування та прийняття класифікаційних рішень, тобто вони безпосередньо моделюють механізм природного інтелекту. Водночас процес

машинного навчання розглядають як оптимізацію параметрів системи розпізнавання, що впливають на її функціональну ефективність. Такі параметри називають параметрами машинного навчання. Як критерій оптимізації в методах ІЕІ-технології можна використовувати будь-яку статистичну інформаційну міру різноманітності аналізованих об'єктів. Якщо в ШНМ глибина машинного навчання обумовлена кількістю прихованих шарів, то в методах ІЕІ-технології її визначають за кількістю параметрів машинного навчання, що оптимізуються. Водночас достатню глибину інформаційно-екстремального машинного навчання визначають згідно з принципом відкладених рішень О. Г. Івахненка за умови досягнення граничного максимального значення усередненого за алфавітом класів розпізнавання інформаційного критерію оптимізації. Вирішальні правила будують за одержаними в процесі машинного навчання оптимальними геометричними параметрами контейнерів класів розпізнавання, що відновлюються в радіальному базисі бінарного простору ознак Геммінга. Побудова вирішальних правил у рамках геометричного підходу робить їх майже інваріантними до багатовимірності простору ознак розпізнавання, тому що сучасні комп'ютерні комплекси можуть обробляти двійкові вектори, що містять 2^{85} ознак розпізнавання. Крім того, такі вирішальні правила характеризуються високою оперативністю прийняття класифікаційних рішень у разі функціонування автономної БСР в робочому режимі, що є важливим фактором для детектування наземних навігаційних орієнтирів.

Отже, використання ідей і застосування методі ІЕІ-технології відкриває широкі перспективи для вирішення проблеми інформаційного синтезу здатної навчатися автономної БСР як складової комплексної комплексної бортової системи БПЛА.

2.2. Основні властивості автономної бортової системи розпізнавання, що навчається

Розглянемо основні властивості БСР, здатної функціонувати в режимах машинного навчання та розпізнавання образів.

Функціональна ефективність – це властивість, що характеризує здатність БСР виконувати поставлене перед нею основне завдання.

Функціональна стійкість до кіберзагроз та несанкціонованих втручань – це властивість БСР повертатися до попереднього робочого функціонального стану після припинення дії збурюючих факторів, зокрема у вигляді атак або несанкціонованих дій.

Керованість – це існування необмеженого керування, що може перевести БСР із довільного початкового функціонального стану в будь-який інший заданий стан за кінцевий інтервал часу.

Спостережуваність – це можливість визначення поточного функціонального стану БСР шляхом аналізу вхідної, робочої та вихідної інформації за час виконання польотного завдання. З точки зору теоретико-інформаційного підходу умовою спостережуваності БСР є наявність умовної кількості інформації, яку вона одержує, аналізує і передає, тобто повинна виконуватися нерівність

$$I = H - H(\gamma) > 0$$

або

$$0 < I \leq H,$$

де H – апіорна (безумовна) ентропія, що характеризує невизначеність даних на вході БСРВН і визначається за формулою

$$H = -\sum_{l=1}^M p(\gamma_l) \log_2 p(\gamma_l), \quad (2.7)$$

де $p(\gamma_l)$ – безумовна ймовірність прийняття гіпотези γ_l ;

M - кількість гіпотез;

$H(\gamma)$ – апостеріорна ентропія, що характеризує залишкову невизначеність після прийняття БСВН класифікаційного рішення і визначається як:

$$H(\gamma) = -\sum_{l=1}^M \sum_{m=1}^M p(\gamma_l) p(\mu_m / \gamma_l) \log_2 p(\mu_m / \gamma_l), \quad (2.8)$$

де $p(\mu_m / \gamma_l)$ – апостеріорна умовна ймовірність прийняття рішення μ_m за умови, що прийнята гіпотеза γ_l .

Система може втратити спостережуваність як за умови впливу зовнішніх, наприклад, погодних умов, наслідків кібератаки тощо, так і внутрішніх збурюючих факторів. Прикладом внутрішнього фактору може бути випадок невинуватого збільшення або зменшення поля контрольних допусків на ознаки розпізнавання, що має наслідком збіг структурованих векторів ознак різних класів розпізнавання, які характеризують відповідні наземні навігаційні орієнтири. Тобто спостережуваність БСВН можна трактувати як властивість системи розрізняти класи розпізнавання.

5. Інформаційна спроможність, що визначається кількісними інформаційними характеристиками БСВН.

6. Точність, що визначається через точнісні характеристики відповідної системи оцінювань рішень. Наприклад, для двохальтернативної системи оцінювань достовірність БСВН характеризується повною ймовірністю правильного прийняття класифікаційних рішень:

$$P_t = p_1 D_1 + p_2 D_2,$$

де p_1 – безумовна ймовірність прийняття основної гіпотези γ_1 ;

p_2 – безумовна ймовірність прийняття альтернативної гіпотези γ_2 ;

D_1 – перша достовірність;

D_2 – друга достовірність;

і повною ймовірністю неправильного прийняття класифікаційних рішень:

$$P_f = p_1\alpha + p_2\beta,$$

де α – помилка першого роду;

β – помилка другого роду.

Оскільки гіпотези γ_1 і γ_2 складають повну групу подій, то має місце

$$P_t + P_f = 1.$$

7. Надійність БСР. Це властивість системи забезпечувати виконання заданих функцій, зберігаючи в часі функціональну ефективність системи в заданих межах.

8. Стабільність БСР. Це властивість системи зберігати незмінними свої характеристики в процесі експлуатації.

9. Енерговитрати, пов'язані з функціонуванням БСР.

10. Вартість як сукупні витрати на всіх етапах життєвого циклу системи.

Зрозуміло, що властивості БСР не обмежуються вищенаведеними, оскільки теорія проектування систем цього класу перебуває в стані непинного розвитку.

2.3. Інформаційні критерії оптимізації параметрів машинного навчання

Центральним питанням інформаційного синтезу здатної навчатися автономної БСР є оцінювання функціональної ефективності процесу машинного навчання, основними критеріями якої є достовірність та оперативність класифікаційних рішень. Як критерії оптимізації параметрів машинного навчання в методах ІЕІ-технології можуть використовуватися різні критерії, що задовольняють таким властивостям інформаційної міри:

- інформаційна міра є дійсна і знакододатна функція від імовірності;
- кількість інформації для детермінованих подій ($p_i = 1$ або $p_i = 0$) дорівнює нулю;
- інформаційна міра має екстремум за значення ймовірності $p_i = \frac{1}{m}$, де m – кількість якісних ознак розпізнавання;
- сумісна інформаційна міра двох незалежних повідомлень дорівнює сумі їх відповідних інформаційних мір.

Серед інформаційних мір для оцінювання функціональної ефективності БСВН, що навчається, перевагу варто віддавати статистичним логарифмічним критеріям, що дозволяють працювати з порівняно малими навчальними вибірками. Серед таких критеріїв найчастіше використовувані ентропійні міри Шеннона та інформаційна міра Кульбака.

Подамо нормований ентропійний критерій оптимізації параметрів машинного навчання системи розпізнавати реалізації класу X_m^o як [1]:

$$E_m^{(k)} = \frac{I_m^{(k)}}{I_{\max}^{(k)}} = \frac{H_m - H_m^{(k)}(\gamma)}{H_m}, \quad (2.11)$$

де $I_m^{(k)}$ – кількість умовної інформації, оброблюваної на k -му кроці машинного навчання системи розпізнавати вектори ознак класу X_m^o ;

$I_{\max}^{(k)}$ – максимальна кількість умовної інформації, одержаної на k -му кроці машинного навчання;

$$H_m = -\sum_{l=1}^M p(\gamma_{l,k}) \log_2 p(\gamma_{l,k}) - \quad (2.12)$$

апріорна (безумовна) ентропія щодо розпізнавання векторів ознак класу X_m^o , яка існує на вході БСВН;

$$H_m^{(k)} = -\sum_{l=1}^M \sum_{m=1}^M p(\gamma_l) p^{(k)}(\mu_m / \gamma_l) \log_2 p^{(k)}(\mu_m / \gamma_l) - \quad (2.13)$$

апостеріорна (умовна) ентропія, що характеризує залишкову невизначеність після k -го кроку навчання системи розпізнавати реалізації класу X_m^o ;

$p(\gamma_l)$ – безумовна ймовірність прийняття гіпотези γ_l ;

$p^{(k)}(\mu_m / \gamma_l)$ – апостеріорна ймовірність прийняття на k -му кроці машинного навчання рішення μ_m за умови, що прийнята гіпотеза γ_l .

Для двохальтернативної системи оцінювань ($M=2$) і рівноймовірних гіпотез, що характеризує згідно з принципом Бернуллі – Лапласа найбільш важкий у статистичному розумінні випадок прийняття рішень, після відповідної підстановки ентропій (2.12) і (2.13) у вираз (2.11) та заміни за формулою Байєса відповідних апостеріорних ймовірностей на апіорні критерій набирає вигляду

$$E_m^{(k)} = 1 + \frac{1}{2} \left(\frac{\alpha_m^{(k)}(d)}{\alpha_m^{(k)}(d) + D_{2,m}^{(k)}(d)} \log_2 \frac{\alpha_m^{(k)}(d)}{\alpha_m^{(k)}(d) + D_{2,m}^{(k)}(d)} + \right. \\ \left. + \frac{\beta_m^{(k)}(d)}{D_{1,m}^{(k)}(d) + \beta_m^{(k)}(d)} \log_2 \frac{\beta_m^{(k)}(d)}{D_{1,m}^{(k)}(d) + \beta_m^{(k)}(d)} + \right. \\ \left. + \frac{D_{1,m}(d)}{D_{1,m}^{(k)}(d) + \beta_m^{(k)}(d)} \log_2 \frac{D_{1,m}(d)}{D_{1,m}^{(k)}(d) + \beta_m^{(k)}(d)} + \right. \\ \left. + \frac{D_{2,m}^{(k)}(d)}{\alpha_m^{(k)}(d) + D_{2,m}^{(k)}(d)} \log_2 \frac{D_{2,m}^{(k)}(d)}{\alpha_m^{(k)}(d) + D_{2,m}^{(k)}(d)} \right), \quad (2.14)$$

де $\alpha_m^{(k)}(d)$ – помилка першого роду прийняття рішення на k -му кроці машинного навчання;

$\beta_m^{(k)}(d)$ – помилка другого роду;

$D_{1,m}^{(k)}(d)$ – перша достовірність;

$D_{2,m}^{(k)}(d)$ – друга достовірність;

d – дистанційна міра, яка визначає радіуси гіперсферичних контейнерів класів розпізнавання, побудованих в радіальному базисі бінарного простору Геммінга;

Оскільки точнісні характеристики є функціями відстані роздільної гіперповерхні від геометричних центрів контейнерів відповідних класів розпізнавання, то критерій (2.14) потрібно розглядати як нелінійний і взаємно-неоднозначний функціонал від точнісних характеристик, що потребує перебування в процесі машинного навчання робочої (допустимої) області його визначення.

Розглянемо процедуру обчислення в практичних задачах критерію (2.14). Оскільки інформаційний критерій є функціоналом від точнісних характеристик, то за репрезентативного обсягу навчальної вибірки необхідно користуватися їх оцінками:

$$\begin{aligned} D_{1,m}^{(k)}(d) &= \frac{K_{1,m}^{(k)}(d)}{n_{\min}}; \quad \alpha_m^{(k)}(d) = \frac{K_{2,m}^{(k)}(d)}{n_{\min}}; \\ \beta_m^{(k)}(d) &= \frac{K_{3,m}^{(k)}(d)}{n_{\min}}; \quad D_{2,m}^{(k)}(d) = \frac{K_{4,m}^{(k)}(d)}{n_{\min}}, \end{aligned} \quad (2.15)$$

де $K_{1,m}^{(k)}(d)$ – кількість подій, які означають належність “своїх” реалізацій класу розпізнавання X_m^o ;

$K_{2,m}^{(k)}(d)$ – кількість подій, які означають неналежність “своїх” реалізацій класу розпізнавання X_m^o ;

$K_{3,m}^{(k)}(d)$ – кількість подій, які означають належність “чужих” реалізацій класу розпізнавання X_m^o ;

$K_{4,m}^{(k)}(d)$ – кількість подій, які означають неналежність “чужих” реалізацій класу розпізнавання X_m^o ;

n_{\min} – мінімальний обсяг репрезентативної навчальної вибірки, який визначається за методом, запропонованим у праці [6].

Після підставлення відповідних позначень (2.15) у вираз (2.14) одержимо робочу формулу для обчислення в рамках ІЕІ-технології ентропійного критерію оптимізації параметрів машинного навчання системи розпізнавати структуровані вектори ознак класу X_m^o :

$$\begin{aligned}
 E_m^{(k)} = 1 + \frac{1}{2} & \left(\frac{K_{1,m}^{(k)}(d)}{K_{1,m}^{(k)}(d) + K_{3,m}^{(k)}(d)} \log_2 \frac{K_{1,m}^{(k)}(d)}{K_{1,m}^{(k)}(d) + K_{3,m}^{(k)}(d)} + \right. \\
 & + \frac{K_{2,m}^{(k)}(d)}{K_{2,m}^{(k)}(d) + K_{4,m}^{(k)}(d)} \log_2 \frac{K_{2,m}^{(k)}(d)}{K_{2,m}^{(k)}(d) + K_{4,m}^{(k)}(d)} + \\
 & + \frac{K_{3,m}^{(k)}(d)}{K_{1,m}^{(k)}(d) + K_{3,m}^{(k)}(d)} \log_2 \frac{K_{3,m}^{(k)}(d)}{K_{1,m}^{(k)}(d) + K_{3,m}^{(k)}(d)} + \\
 & \left. + \frac{K_{4,m}^{(k)}(d)}{K_{2,m}^{(k)}(d) + K_{4,m}^{(k)}(d)} \log_2 \frac{K_{4,m}^{(k)}(d)}{K_{2,m}^{(k)}(d) + K_{4,m}^{(k)}(d)} \right). \tag{2.16}
 \end{aligned}$$

У праці [1] запропоновано модифікацію диференціальної інформаційної міри Кульбака, яка подається як добуток відношення правдоподібності на міру відхилень відповідних розподілів імовірностей. Для двохальтернативних апріорно рівноймовірних рішень модифікований критерій Кульбака, який обчислюється на k -му кроці машинного навчання системи розпізнавати реалізації класу X_m^o , має вигляд

$$E_m^{(k)} = \log_2 \left(\frac{2 - (\alpha_m^{(k)}(d) + \beta_m^{(k)}(d))}{\alpha_m^{(k)}(d) + \beta_m^{(k)}(d)} \right) * [1 - (\alpha_m^{(k)}(d) + \beta_m^{(k)}(d))]. \tag{2.17}$$

Нормована форма критерію (2.17) має вигляд

$$E_{K,m}^{(k)} = \frac{E_{K m}^{(k)}}{E_{K \max}^{(k)}}, \tag{2.18}$$

де $E_{K_{\max}}^{(k)}$ – значення інформаційного критерію (2.17) при

$$D_{1,m}^{(k)}(d) = D_{2,m}^{(k)}(d) = 1; \alpha_m^{(k)}(d) = \beta_m^{(k)}(d) = 0.$$

Робоча модифікація критерію (2.17) після відповідного підставлення оцінок (2.15) набуває вигляду

$$E_m^{(k)} = \frac{\lfloor n - (K_{2,m}^{(k)}(d) + K_{3,m}^{(k)}(d)) \rfloor}{n} * \log_2 \left\{ \frac{2n + 10^{-r} - \lfloor K_{2,m}^{(k)}(d) + K_{3,m}^{(k)}(d) \rfloor}{\lfloor K_{2,m}^{(k)}(d) + K_{3,m}^{(k)}(d) \rfloor + 10^{-r}} \right\}, \quad (2.19)$$

де 10^{-r} – достатньо мале число, яке вводиться для уникнення поділу на нуль.

Розглянемо схему обчислення змінних $K_1^{(k)} - K_4^{(k)}$. На рисунку 2.1 показано структуру навчальної матриці під час побудові оптимального контейнера класу розпізнавання X_1^o . Навчальна матриця складається з векторів реалізацій двох найближчих сусідніх класів розпізнавання: $\{x_1^{(j)}\} \in X_1^o$ і $\{x_2^{(j)}\} \in X_2^o$.

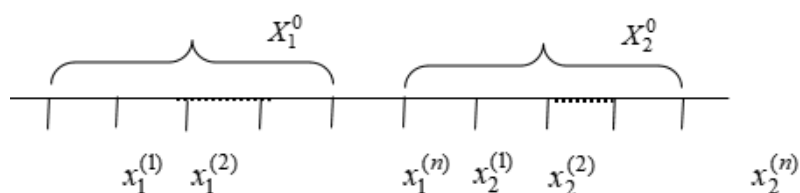


Рисунок 2.1 – Структура навчальної матриці найближчих сусідніх класів розпізнавання

Схема обчислення змінних $K_1^{(k)} - K_4^{(k)}$ має вигляді:

$$\begin{aligned} & (\forall X_1^o \in \mathfrak{R}^{|\Lambda|}) (\forall X_2^o \in \mathfrak{R}^{|\Lambda|}) [\text{if } x_1^{(j)} \in X_1 \text{ then} \\ & \quad K_1(j) := K_1(j-1) + 1 \text{ else } K_2(j-1) + 1], \\ & (\forall X_1^o \in \mathfrak{R}^{|\Lambda|}) (\forall X_2^o \in \mathfrak{R}^{|\Lambda|}) [\text{if } x_2^{(j)} \in X_1 \text{ then} \end{aligned}$$

$$K_3(j) := K_3(j-1) + 1 \text{ else } K_4(j) := K_4(j-1) + 1].$$

Водночас визначення належності, наприклад, вектору $x_m^{(j)}$ до свого класу здійснюється за таким правилом обчислюється кодова відстань $d[x_m \oplus x_m^{(j)}]$ якщо $d[x_m \oplus x_m^{(j)}] \leq d_m$, то $x_m^{(j)} \in X_m^o$, інакше – $x_m^{(j)} \notin X_m^o$.

Таким чином, інформаційні критерії (2.14) і (2.17) є функціоналами як від точнісних характеристик класифікаційних рішень, так і від дистанційних критеріїв, що дозволяє їх вважати загальними критеріями валідності машинного навчання, оскільки вони є узагальненням відомих статистичних і детермінованих (дистанційних) критеріїв близькості.

2.4. Базовий алгоритм інформаційно-екстремального машинного навчання автономної бортової системи відеонавігації

Базовий інформаційно-екстремальний алгоритм оптимізації просторово-часових параметрів функціонування автономної БСР реалізується у внутрішньому циклі процедури машинного навчання (3.1), що й обумовило його назву. Призначенням базового алгоритму навчання є:

- оптимізація геометричних параметрів контейнерів класів розпізнавання;
- обчислення інформаційного критерію оптимізації параметрів машинного навчання системи;
- пошук глобального максимуму інформаційного критерію в робочій (допустимій) області визначення його функції.

Категорійну модель інформаційно-екстремального машинного навчання БСВН за базовим алгоритмом подамо у вигляді спрямованого графа відображення операторами одна на одну відповідних множин, які застосовуються в процесі навчання. Вхідний математичний опис здатної навчатися БСР в праці [1] розглядається у вигляді структури

$$I_{ex} = \langle G, T, \Omega, Z, Y, X; f_1, f_2 \rangle, \quad (2.20)$$

де G – простір вхідних сигналів (факторів);

T – множина моментів часу зняття інформації;

Ω – простір ознак розпізнавання;

Z – простір можливих функціональних станів БСР;

Y – вхідна навчальна матриця;

X – робоча бінарна навчальна матриця, яка в процесі інформаційно-екстремального машинного навчання адаптується до максимальної повної ймовірності прийняття правильних класифікаційних рішень;

f_1 – оператор формування вхідної навчальної матриці Y із джерела інформації, яке задається декартовим добутком $G \times T \times \Omega \times Z$;

f_2 – оператор перетворення вхідної навчальної матриці Y в робочу бінарну навчальну матрицю X .

Функціональну категорійну модель інформаційно-екстремального машинного навчання автономної БСР за базовим алгоритмом показано на рисунку 2.2 [1].

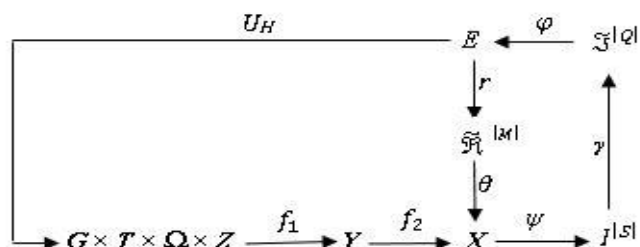


Рисунок 2.2 – Функціональна категорійна модель машинного навчання за базовим алгоритмом

На рисунку 2.2 терм-множина E , яка складається із обчислених на кожному кроці машинного навчання значень інформаційного критерію, є загальною для всіх контурів оптимізації параметрів. Оператор $r : E \rightarrow \tilde{\mathfrak{R}}^{|M|}$ в процесі машинного навчання відновлює в радіальному базисі простору ознак контейнери класів розпізнавання, які утворюють розбиття $\tilde{\mathfrak{R}}^{|M|}$. Оператор θ

відображає розбиття $\tilde{\mathfrak{R}}^{|M|}$ на нечіткий розподіл апріорно класифікованих двійкових векторів ознак класів розпізнавання. Далі оператор $\psi: X \rightarrow I^{|S|}$, де $I^{|S|}$ – множина гіпотез, перевіряє основну статистичну гіпотезу $\gamma_1: x_m^{(j)} \in X_m^0$. Оператор γ визначає множину точнісних характеристик $\mathfrak{Z}^{|Q|}$, де $Q = S^2$, а оператор φ обчислює множину значень E інформаційного критерію оптимізації, якій є функціоналом від точнісних характеристик. Контур оптимізації контрольних допусків замикається через терм-множину D , елементами якої є значення контрольних допусків на ознаки розпізнавання. Оператор u регламентує процес машинного навчання.

Згідно з принципом відкладених рішень О. Г. Івахненка для максимізації інформаційної спроможності системи може бути необхідною оптимізація інших параметрів, які впливають на функціональну ефективність машинного навчання. У цьому разі категорійна модель буде мати додаткові контури оптимізації цих параметрів. Крім того, згідно з принципом повної композиції контури оптимізації повинні мати загальну терм-множину E , елементи якої на кожному кроці машинного навчання.

Вхідною інформацією для навчання за базовим алгоритмом є тривимірний масив обчислюються реалізацій класів розпізнавання $\{y_{m,i}^{(j)} \mid m = \overline{1, M}; i = \overline{1, N}; j = \overline{1, n}\}$; значення параметра поля контрольних допусків δ на ознаки розпізнавання і рівні селекції (квантування) $\{\rho_m\}$ координат усереднених двійкових векторів ознак класів розпізнавання, які за замовчуванням дорівнюють 0,5 для всіх класів розпізнавання. За базовий беруть клас розпізнавання X_1^0 , який характеризує максимальну функціональну ефективність автономної БСВН і стосовно якого визначаються контрольні допуски.

Розглянемо етапи реалізації базового алгоритму інформаційно-екстремального машинного навчання [5]:

1. Обчислюється для навчальної матриці класу розпізнавання X_1^o усереднений вектор ознак $\{y_{1,i} \mid i = \overline{1, N}\}$.

2. Формується масив $\{x_1^{(j)}\}$ двійкових векторів ознак класу розпізнавання X_1^o за правилом

$$x_1^{(j)} = \begin{cases} 1, & \text{if } y_{1,i} - \delta \leq y_{1,i}^{(j)} \leq y_{1,i} + \delta, \\ 0, & \text{if else.} \end{cases} \quad (2.21)$$

3. Формується масив усереднених двійкових векторів-реалізацій $\{x_{m,i} \mid m = \overline{1, M}, i = \overline{1, N}\}$, елементи яких визначаються за правилом

$$x_{m,i} = \begin{cases} 1, & \text{if } \frac{1}{n} \sum_{j=1}^n x_{m,i}^{(j)} > \rho_m, \\ 0, & \text{if else,} \end{cases}$$

де ρ_m – рівень селекції координат двійкового вектора $x_m \in X_m^o$.

4. Розбиття множини усереднених векторів ознак на пари найближчих «сусідів» $\mathcal{R}_m^{[2]} = \langle x_m, x_l \rangle$, де x_l – усереднений вектор ознак сусіднього класу X_l^o , за такою схемою:

а) структурують множини векторів $\{x_m\}$, починаючи з вектора x_1 базового класу X_1^o ;

б) будують матрицю розмірності $M \times M$ кодових відстаней між усередненими векторами ознак усіх класів розпізнавання;

в) для кожного рядка матриці кодових відстаней обчислюють мінімальний елемент, який належить стовпчику вектора, найближчого до вектора, що визначає рядок. За наявності декількох однакових мінімальних елементів вибирають з них будь-який, оскільки вони є рівноправними;

г) формують структуровану множину елементів попарного розбиття $\{\mathcal{R}_m^{[2]} \mid m = \overline{1, M}\}$, яка задає план машинного навчання.

5. Здійснюють оптимізацію кодової відстані d_m , яка змінюється за заданим законом. У цьому разі беруть $E_m(0) = 0$.

6. Процедура закінчується в разі знаходження максимуму інформаційного критерію оптимізації параметрів машинного навчання в робочій області визначення його функції.

Таким чином, базовий алгоритм навчання є ітераційною процедурою пошуку глобального максимуму інформаційного критерію оптимізації параметрів машинного навчання в робочій області визначення його функції:

$$d_m^* = \arg \max_{G_E \cap \{d\}} E_m^*(d). \quad (2.22)$$

На рисунку 2.3 показано геометричну інтерпретацію реалізації базового алгоритму інформаційно-екстремального машинного навчання <СН на прикладі побудови оптимального гіперсферичного контейнера класу розпізнавання X_1^o .

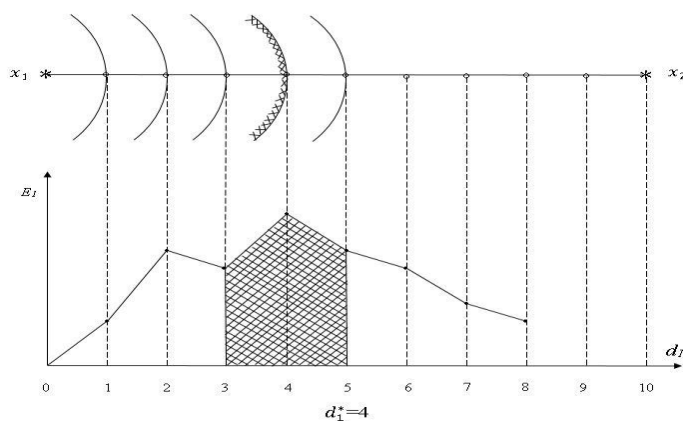


Рисунок 2.3 – Схема реалізації базового алгоритму інформаційно-екстремального машинного навчання

На рисунку 2.3 показано визначені в бінарному просторі Геммінга вершини усередненого двійкового вектору x_1 класу розпізнавання X_1^o та усередненого двійкового вектора x_2 класу розпізнавання X_2^o , який є найближчим сусідом для класу розпізнавання X_1^o . На схемі кодова відстань

Геммінга між векторами x_1 і x_2 дорівнює $d = (x_1 \oplus x_2) = 11$. У процесі машинного навчання за базовим алгоритмом здійснюється покрокове збільшення радіуса d_1 контейнера класу розпізнавання X_1^o на одну кодову одиницю. Водночас згідно з умовою (2.7) на величину радіуса d_1 накладається обмеження

$$d_1 < d(x_1 \oplus x_2) - 1.$$

На кожному кроці машинного навчання обчислюється значення інформаційного критерію E_1 оптимізації радіуса контейнера класу розпізнавання X_1^o . Як оптимальний радіус беруть екстремальне значення глобального максимуму критерію E_1 , обчислене в робочій (допустимій) області визначення функції інформаційного критерію. На рисунку 2.3 робоча область обчислення інформаційного критерію позначена заштрихованою ділянкою. Для двохальтернативних рішень робоча область існує за умови, що їх перша і друга достовірності перевищують відповідно помилки першого та другого родів. Аналіз рисунка 5 засвідчує, що максимальне значення критерію E_1^* одержане на четвертому кроці машинного навчання, тобто оптимальний радіус контейнера класу розпізнавання X_1^o дорівнює $d_1^* = 4$ (тут і далі в кодових одиницях).

Таким чином, основною функцією базового алгоритму машинного навчання в рамках ІЕІ-технології є обчислення на кожному кроці навчання інформаційного критерію та організація пошуку його глобального максимуму в робочій області визначення функції критерію з метою визначення оптимальних геометричних параметрів розбиття простору ознак на класи розпізнавання. При гіперсферичному контейнері класів розпізнавання такими параметрами при інформаційно-екстремальному машинному навчанні за базовим алгоритмом є оптимальні кодові відстані $\{d_m^*\}$ і оптимальні усереднені вектори-реалізації $\{x_m^*\}$ для заданого алфавіту $\{X_m^o\}$.

3. ІНФОРМАЦІЙНЕ. АЛГОРИТМІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ АВТОНОМНОЇ БОРТОВОЇ СИСТЕМИ РОЗПІЗНАВАННЯ НАЗЕМНИХ ОБ'ЄКТІВ

3.1. Формування вхідного інформаційного опису автономної бортової системи розпізнавання наземних об'єктів

Перед початком формування вхідного інформаційного опису було проведено аналіз специфіки предмету дослідження. У тому числі й умов функціонування безпілотного літального апарату. Таким чином вдалося виявити, що усі об'єкти, які ідентифікує БСР можна розділити за типом на два кластери:

- Техногенного походження
- Природного походження

Перша група включає в себе все, що було створено людиною. Наприклад: будинки, теплотраса, опора ліній електропередачі, тощо. У свою чергу до природних об'єктів можна віднести дерева, ліс, водоймища та різного роду рослинність.

Безпосереднє моделювання 3Д макету місцевості було розпочато з розмітки майбутньої карти. Звісно, що для її створення можна взяти реально існуючу територію та спробувати відтворити її у рамках симуляції, але слід враховувати, що для адекватного навчання та тестування БСР потрібна така місцевість, яка буде достатньо інформативною. Тобто матиме адекватну різноманітність об'єктів та особливостей ландшафту, але при цьому, на поточний момент часу, бажано забезпечити низький перетин ознак класів розпізнавання. Таким чином, щоб спростити класифікацію об'єктів для БСР, а в подальшому поступово ускладнювати місцевість та завдання, які ставляться перед ІС. У цьому і полягає головна ідея симулятора.

Враховуючи це, було прийняте рішення самостійно спланувати макет селища, яке б містило в собі усі необхідні об'єкти і при цьому було б достатньо

зручним для ідентифікації з борту безпілотної літака. На рисунку 3.1 схематично зображено план майбутньої місцевості.

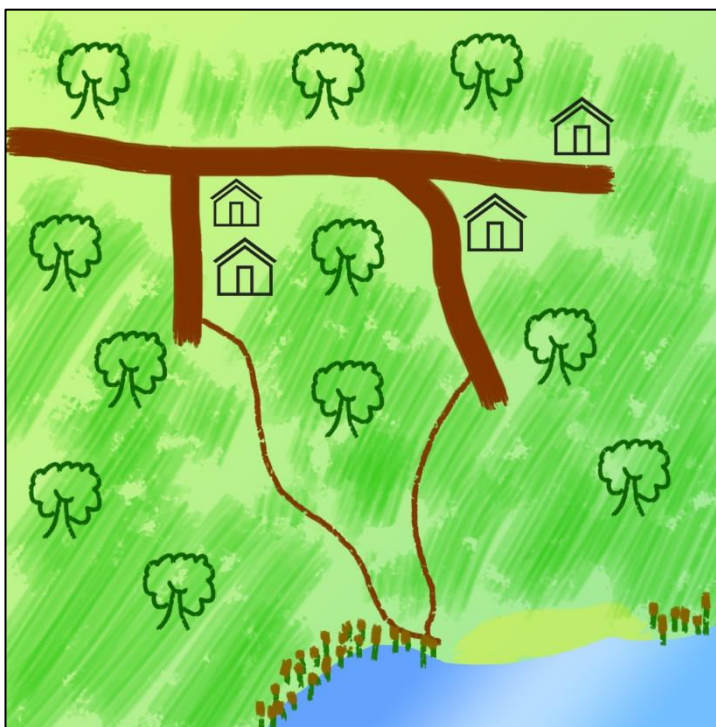


Рисунок 3.1. План місцевості

Для відтворення 3Д макету було обрано інструмент Unreal Engine 4 (UE4), який дозволяє створити інтерактивну карту достатніх для випробування розмірів. Перше завдання, яке необхідно виконати при моделюванні ландшафту – це самостійне створення текстури поверхні майбутньої місцевості. При цьому UE4 дозволяє розробити її таким чином, щоб вона враховувала неоднорідність поверхні ландшафту, накладаючи на схили текстуру каменю. Це надасть пагорбам та впадинам більш реалістичного вигляду, що буде помітно з борту безпілотної літака. Окрім цього, згідно з ескізу (рис. 3.1), територія буде мати ділянку з пляжем та пішохідні доріжки. Тобто, необхідно додати ще шари піску та гравію. Враховуючи ці умови у мережі інтернет були знайдені такі текстури, які є безшовними (при складенні їх поряд не видно переходів між секторами) і задовольняють вказані потреби:

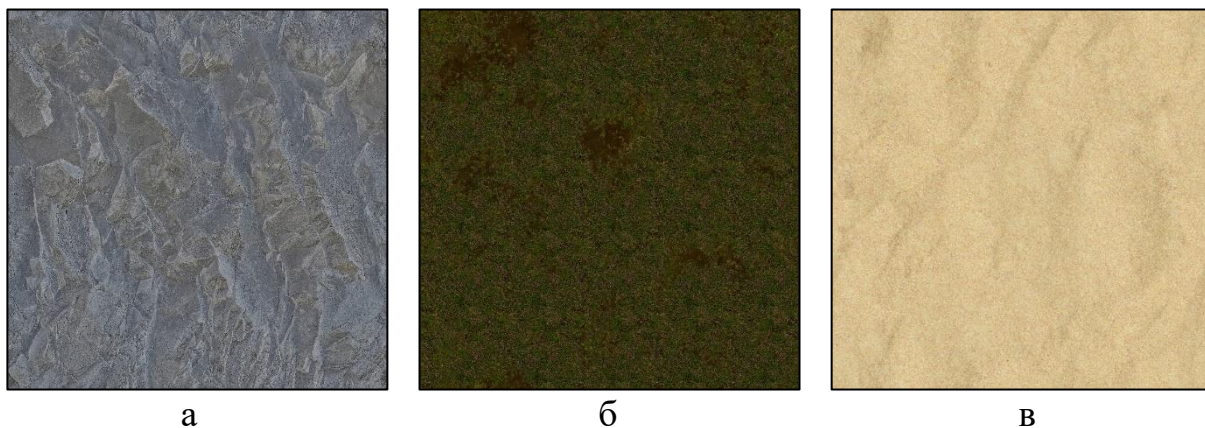


Рисунок 3.2. Текстури ландшафту: а) камінь; б) земля з травою; в) пісок

Вбудовані інструменти довільної деформації ландшафту дозволяють розробити такі об'єкти як озеро без втручання інших утиліт (World Editor, тощо.), а вже існуючі матеріали покриття фігури зроблять звичайну площину – водою з невеликими хвилями. За допомогою інструменту «Sculpt» робимо пологий схил, що пізніше стане пляжним берегом, а далі по периметру – різкий спуск, що буде диким берегом. Після чого треба створити геометричну фігуру куб та розтягнути до розмірів озера з вирівнюванням його по висоті таким чином, щоб він імітував водну поверхню. Як його буде налаштовано рівно, накладаємо матеріал води. Результатом стане такий вид:

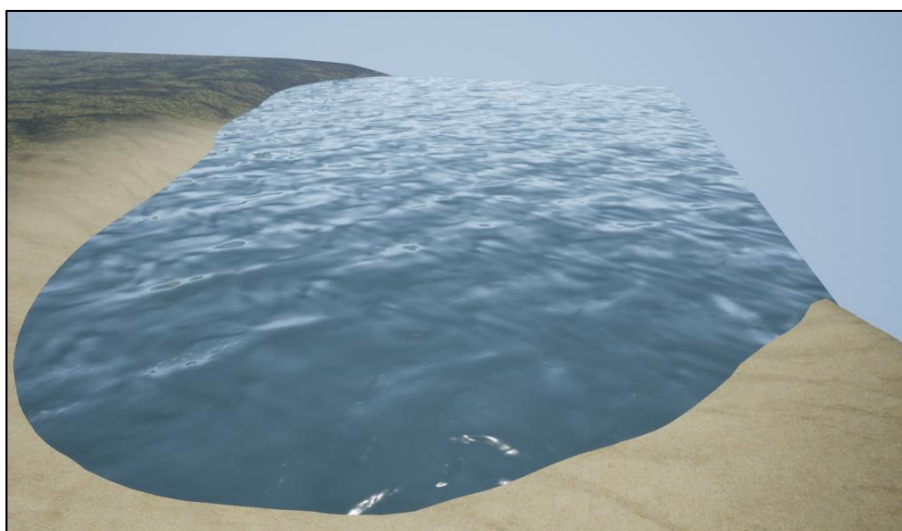


Рисунок 3.3. Готова основа озера

Ця основа водоймища буде доповнена рослинністю, характерною для подібних місць, висока трава в дикій частині узбережжя та надводні лілії.

Згідно плану (рис. 3.1) селище повинно мати автомобільні дороги. Тим не менш, для віддаленого від адміністративного центру населеного пункту, як правило, не характерні добре асфальтовані проїжджі частини. Для нашого регіону більш звичні ґрунтові або піщані дороги в таких місцях. Уздовж яких немає тротуарів чи бордюрів, їх замінює проста протоптана доріжка, іноді усипана гравієм.

Процес створення подібних інфраструктурних об'єктів в UE4, схожий на вже описаний алгоритм. Спочатку, за допомогою інструментів формування ландшафту, вручну робимо грязьову, неоднорідну дорогу. У нашому випадку (рис.3.1), вона проходить через усю карту, маючи при цьому два розгалуження у вигляді троп. Звісно, що як і на минулому етапі, необхідні відповідні текстури, які будуть накладені на заздалегідь підготовлені ділянки ландшафту, рисунок 3.4



Рисунок 3.4. Текстури доріг: а) ґрунт; б) гравій

Невід'ємною частиною будь-якого ландшафту є рослинність. Без неї пейзаж виглядає порожнім та штучним. За допомогою вбудованого в Unreal Engine 4 інструменту «Foliage» можна зручно розмножити по карті такі об'єкти як: дерева, трава, кущи, тощо. Слід зазначити, якщо виставляти кожне дерево вручну, то це, по перше, займе багато часу, а, по друге, кожна модель буде вважатися за окремий об'єкт, який потребуватиме індивідуального рендеренгу тіней та матеріалів, що значно збільшить навантаження на процесор та відеокарту комп'ютера. Для заповнення мапи було обрано готові моделі дерев

«European HornBeam», створені професійними художниками. При розміщенні об'єктів враховувалися умови поставленого завдання. Техногенні споруди повинні бути добре помітні. Древа не мають їх закривати.

Водоймище також потребує оформлення у вигляді лілій та рогозу. За допомогою вищезазначеного інструментарію, було розмножено водні рослини, із набору «MS Aquatic», довкола узбережжя. Після цього озеро виглядає так:

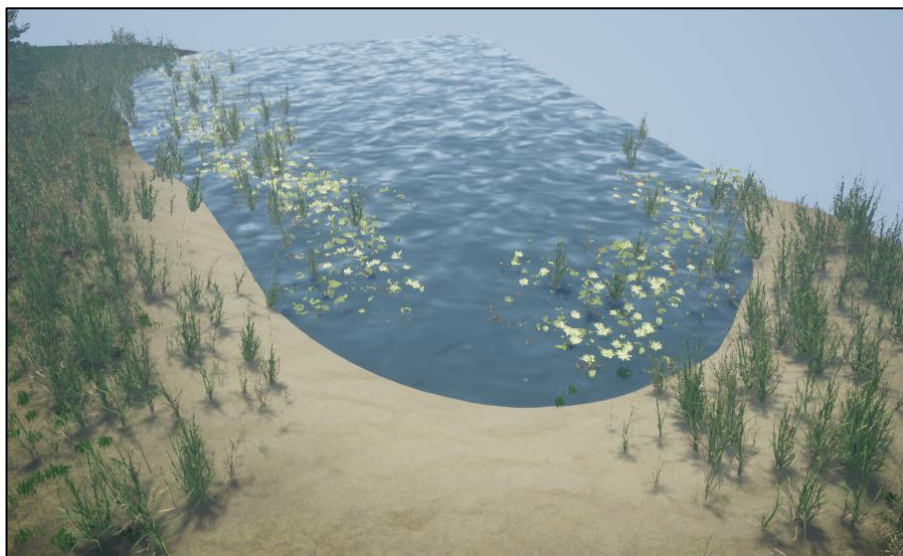


Рисунок 3.5. Озеро з декораціями

Наступним етапом заповнення карти є розміщення техногенних об'єктів. Для їх моделювання можна застосувати два підходи:

1. Придумати будівлі, наповнення дворів та інше з нуля;
2. Знайти реально існуючі об'єкти та відтворити їх у вигляді 3Д моделей;

Підхід для створення техногенних об'єктів, що будуть розміщені на карті, є змішаним. Деякі будинки мають реальний прототип, а деякі були побудовані на основі власного досвіду та знань про місцевість. Прикладом створення з «фотографії» є приватний будинок у м. Суми на одній з вулиць Роменського мікрорайону (рис. 3.6). Для його переносу в 3Д треба визначити характерні риси цієї споруди:

- Кількість і розташування вікон
- Архітектура побудови

- Матеріал
- Форма даху та його матеріал



а



б

Рисунок 3.6. Прототип будинку: а) фото фронтальної сторони будинку; б) Фото з боку

З фотографії 3.6.а розуміємо, що з фронтальної сторони будинку є близько п'яти пластикових вікон, одне з яких маленького розміру та знаходиться на горищі. Форма будинку квадратна з цегляними стінами, є прибудова у вигляді коридору, матеріал той же. Вхідні двері пластикові з великою кількістю скла. Матеріал даху є шиферним, на стику двох половин є металева кришка, яка запобігає краплям дощу затікати під дах. Біля входу є невеличкий ганок з плитки та підпорка для даху, оформлена каменем. Результатом моделювання є наступний комплексний об'єкт:



Рисунок 3.7. Будинок, створений по фото

На мапі також присутня споруда (рис.3.8), яка створена по узагальненому образу приватних житлових будинків. Має дах із шиферу, горище з входом з вулиці по драбині. Вікна пластикові, але двері старі, над якими висить іржавий плафон з лампочкою.

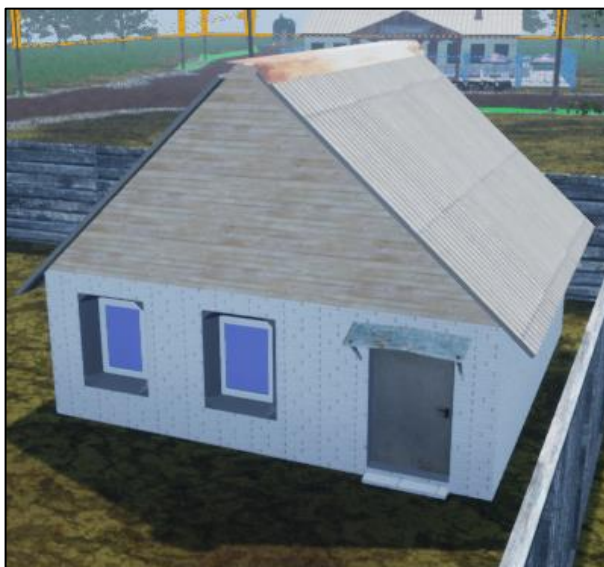


Рисунок 3.8. Не маючий прототипу будинок

На всій карті можна побачити невеликі квіткові клумби, які знаходяться перед будинками та іншими спорудами. Такі насадження особливо характерні для приватного сектору, або селищ. Огороджені клумби по-різному: від старих покришок до міні паркану. За допомогою вже описаного інструменту «Foliage»

квіти розмножені та не перевантажують систему при рендерингу карти. Приклад клумби зображено на наступному рисунку:



Рисунок 3.9. Квіткова клумба перед будинком

Навіть в маленькому селищі має бути крамниця, тому на створеній карті вона також є. У вигляді типової пострадянської споруди. Має перед собою невелику територію для культурного відпочинку з лавочками та столиками.



Рисунок 3.10. Гастроном

На передньому плані можна помітити клумби, огорожені старими автомобільними покрішками. За рогом є невеликий склад бочок та білої цегли.



Рисунок 3.11. Бочки та цегла на піддонах

Уздовж доріг є дерев'яні ліхтарі. Їх вже не зустріти в великих або середніх за розмірами містах, бо там є бетонні, але в селищах міського типу або маленьких поселеннях все ще трапляються подібні рудименти. Вони були розставлені по карті за допомогою вбудованої функції Spline, яка поділяється на сегменти, між якими є контрольні точки. В яких і будуть розміщені стовпи, а дроти будуть протягнуті уздовж сегментів.

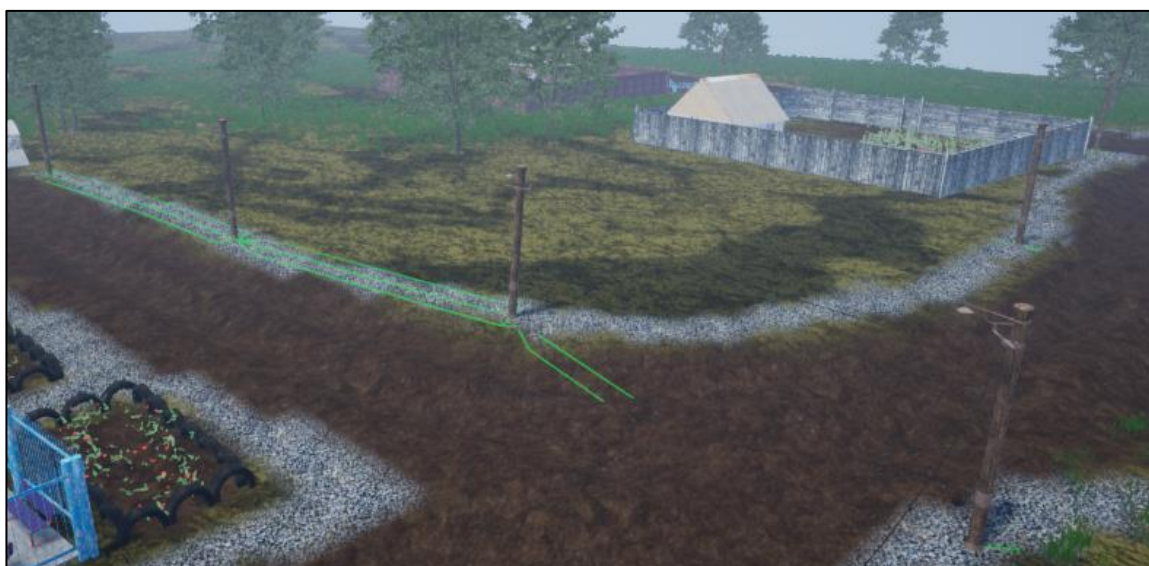


Рисунок 3.12. Об'єднані дротами стовпи

Останній набір рукотворних об'єктів, який вартий уваги, розташований в глухому куті головної дороги. На цьому місці знаходиться покинутий дім, який

практично повністю був зруйнований та невеликий іржавий автомобільний причеп на цеглі. В середині будинку все в траві, а біля вагончика немає доріжок. Такі місця є не тільки в селищах, але й на окраїнах міст.

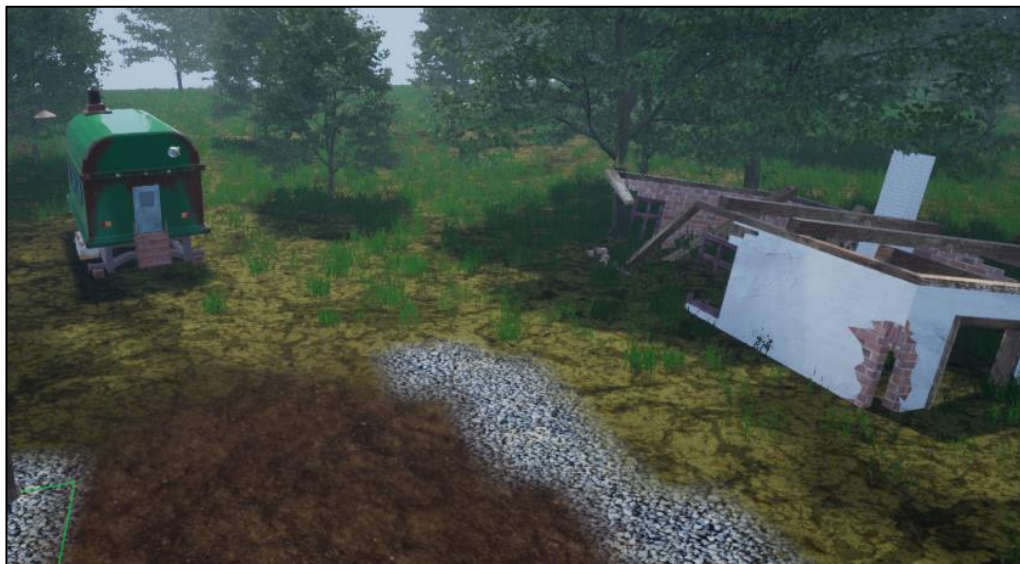


Рисунок 3.13. Глухий кут та покинуті об'єкти

3.2. Вхідний математичний опис бортової системи розпізнавання.

Для перевірки функціональної ефективності, тестування та формування вхідного математичного опису був використаний багатofункціональний симулятор динамічного оточення [7, 8]. На рисунку 3.14 зображено вид з борту БПЛА, приблизно на висоті 70-100 метрів.



Рисунок 3.14. 3D-модель місцевості

Враховуючи специфіку завдання та функціонування БПЛА було прийняте рішення бінаризувати зображення місцевості: X_1^0 - техногенні об'єкти; X_2^0 – природні об'єкти. На рисунку 3.15 показано кадри ділянок, розміром 52 на 52 пікселів, які були обрані як класи розпізнавання місцевості (рис.3.14).

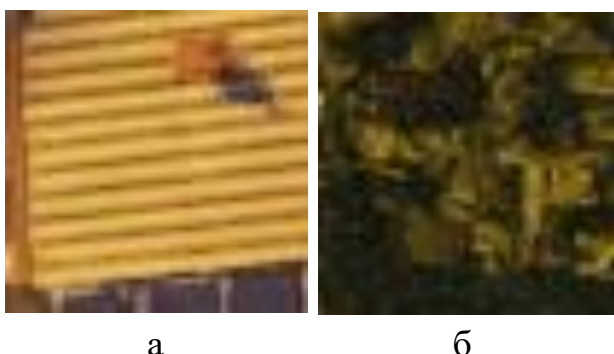


Рисунок 3.15. Класи розпізнавання: а – клас X_1^0 ; б – клас X_2^0 ;

Варто зазначити, що X_1^0 та X_2^0 є не стаціонарними за яскравістю класами розпізнавання. Це означає, що під час функціонування системи ці об'єкти можуть займати довільне положення в кадрі.

Враховуючи концепції ІЕІ-технології, яка була обрана як метод дослідження, сформуємо зміст вхідного математичного опису БСР:

- 1) словник ознак розпізнавання;
- 2) алфавіт класів розпізнавання;
- 3) вхідна навчальна матриця яскравості $\|y_{m,i}^{(j)}\|$, $m = \overline{1, M}$; $i = \overline{1, N}$, $j = \overline{1, n}$
- 4) робоча, бінарна навчальна матриця $\|x_{m,i}^{(j)}\|$, яка в процесі машинного навчання адаптується до максимальної повної ймовірності прийняття правильних класифікаційних рішень;
- 5) нормоване поле допусків δ_H на яскравості ознак розпізнавання, яке визначає область значень параметра δ поля контрольних допусків;
- 6) рівень селекції ρ_m координат усередненого двійкового вектору-реалізації класу розпізнавання X_m^o , який є рівнем квантування реалізацій вхідної навчальної матриці. За замовчуванням приймається $\rho_m = 0.5$.

3.3. Результати програмного моделювання

Для реалізації інформаційно-екстремального машинного навчання бортової системи розпізнавання, шляхом оброблення кадрів зображення місцевості в декартовій системі координат, була сформована вхідна навчальна матриця. Як критерій оптимізації було обрано модифіковану міру Кульбака в своїй нормованій формі (2.18).

На рисунку 3.16 показано графік залежності усередненого, за алфавітом класів розпізнавання, нормованого критерію (2.18) від параметра δ поля контрольних допусків, отриманих при їх паралельній оптимізації. Тут і далі робочу область визначення функції критерію виділено темним кольором.

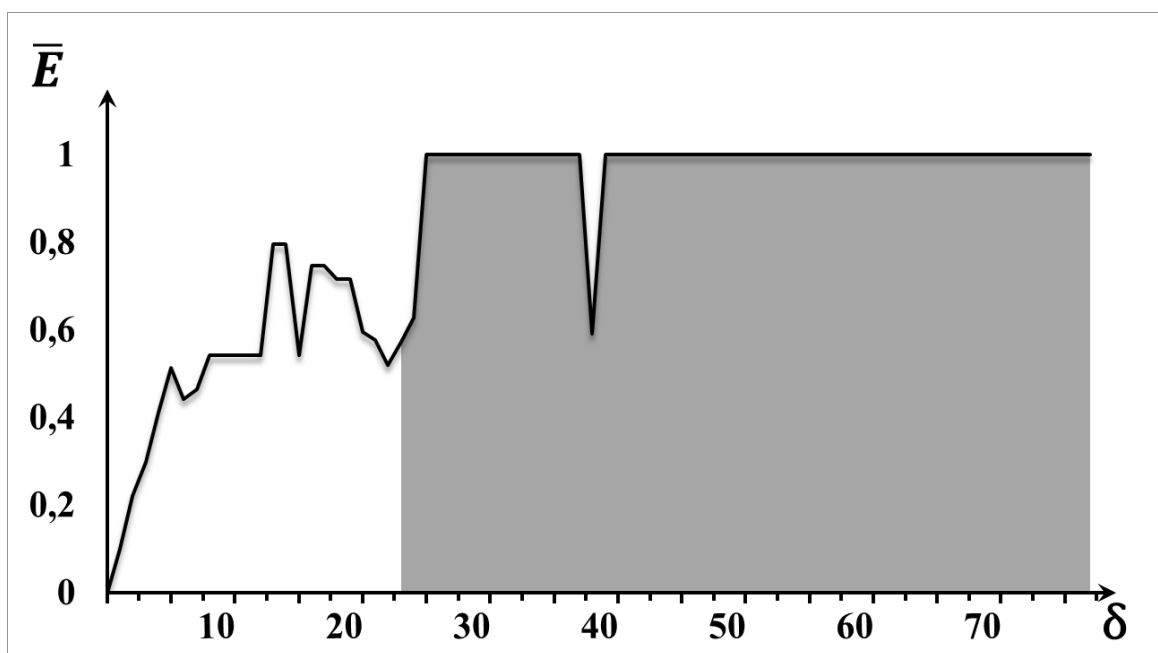


Рисунок 3.16 – Графік залежності інформаційного критерію (2.13) від параметра поля контрольних допусків на ознаки розпізнавання

Аналіз рисунку 3.16 показує, що оптимальне значення параметра поля контрольних допусків проблемно визначити візуально, адже критерій набуває свого максимального значення далеко не в одній точці і утворює, так зване, «плато». Оскільки ми бінаризуємо карту, то нам потрібні такі класи, які будуть охоплювати максимально можливу кількість ознак, тому для обрання

найкращого значення δ застосуємо дистанційний критерій. Таким чином, щоб відстань між класами була максимальна.

Таблиця 3.1 Таблиця залежності інформаційного критерію (2.18) від параметра поля контрольних допусків на ознаки розпізнавання

δ	\bar{E}	D_1	β	d_1	d_c
30	1	1	0	38	45
31	1	1	0	41	48
32	1	1	0	43	50
33	1	1	0	44	51
34	1	1	0	44	51
35	1	1	0	43	51
36	1	1	0	42	51

Аналіз таблиці 3.1 показує, що оптимальне значення параметра поля контрольних допусків дорівнює $\delta^* = 33$ (в градаціях яскравості) при максимальному значенні інформаційного критерію $\bar{E}^* = 1$. Оскільки ми вже досягли максимально граничного значення інформаційного критерію, то немає необхідності у додатковій реалізації послідовного алгоритму оптимізації системи контрольних допусків.

На рисунку 3.17 показано графіки залежності інформаційного критерію (2.18) від радіусів контейнерів розпізнавання, оптимальні значення яких дозволяють побудувати вирішальні правила (2.22).

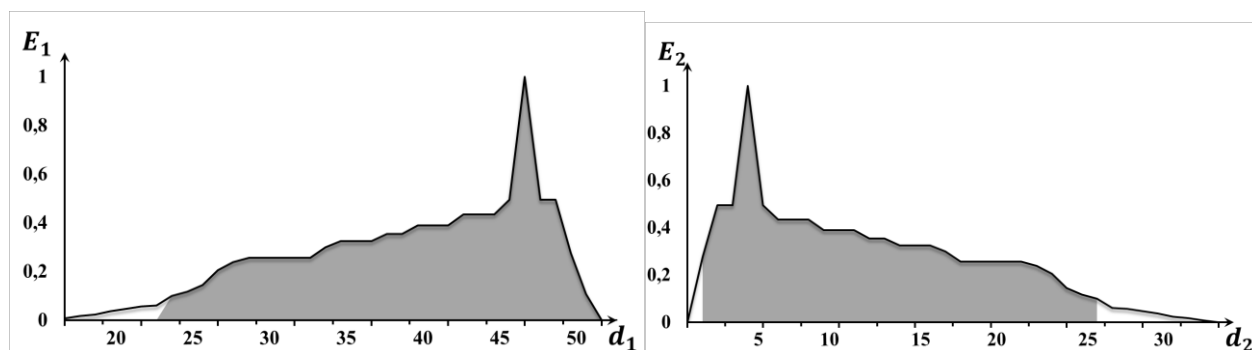


Рисунок 3.17 – Графіки залежності критерію (2.18) від радіусів контейнерів класів розпізнавання: а – клас X_1^0 ; б – клас X_2^0 ;

Аналіз рисунку 3.17 показує, що оптимальні значення радіусів контейнерів класів розпізнавання дорівнюють: $d_1^* = 46$ (тут і далі в кодових одиницях) – для класу X_1^0 ; $d_2^* = 4$ – для класу X_2^0 ;

Перевірка ефективності машинного навчання бортової системи розпізнавання здійснювалось в режимі екзамену, за результатами якого було отримане оцифроване зображення місцевості, рисунок 3.14.

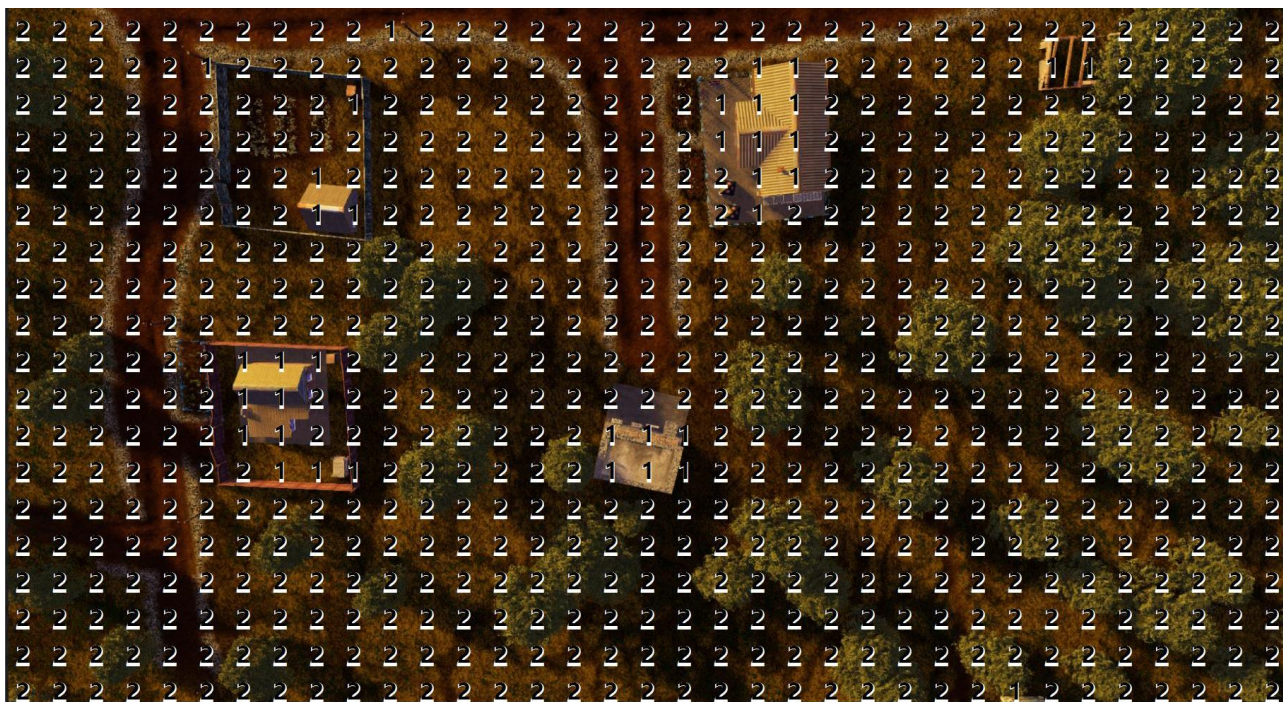


Рисунок 3.18. Результат класифікації кадрів

Візуальний аналіз рисунку 3.18 показує високу достовірність класифікаційних правил. Тим не менш, не можна стверджувати, що при втручанні зовнішніх факторів на роботу БПЛА об'єкти будуть розпізнаватися так само добре. Проведемо експеримент, де при початкових, не змінних умовах (рис.3.14) додамо ледь помітну димку, яка прикриє собою рецепторне поле безпілота. Спробуємо ідентифікувати новий інформаційний опис місцевості.



Рисунок 3.19. Результат класифікації кадрів при зміні початкових умов

Як можна побачити з рисунку 3.19 вирішальні правила помилково ідентифікували димку, як техногенний об'єкт. Це пов'язано, насамперед, з недостатньою чутливістю розробленого класифікатору. Теоретично це можна виправити на етапі паралельної оптимізації системи контрольних допусків (табл.3.1). Замість обрання максимально віддалених між собою класів, можна спробувати навпаки обрати найближчі один до одного. Але це не дасть стовідсоткової впевненості в результаті.

Для забезпечення релевантності класифікатору на етапі екзамену пропонується ввести додатковий квантор перевірки ідентифікації за допомогою апріорно отриманої статистичної інформації. Знаючи, що попередній результат (рис.3.18) є високо достовірним спробуємо вивести деякі закономірності для класів розпізнавання, а саме: усереднене значення яскравості (3.1) та усереднене значення дисперсії яскравості (3.2)

$$\mu_m = \frac{1}{n} \sum_{i=1}^N \theta_i \quad (3.1)$$

де μ_m – усереднене значення яскравості пікселів для класу m , $m = \overline{1, M}$; θ_i – значення яскравості RGB-складової в i -му пікселі рецепторного поля зображення кадру; N – кількість ознак;

Дисперсію шукаємо як міру відхилення кожної окремої ознаки від усередненого значення яскравості $\bar{\theta}$, за формулою (3.2):

$$S_m = \frac{1}{N-1} \sum_{i=1}^N (\theta_{i,m} - \bar{\theta})^2 \quad (3.2)$$

де S_m – значення дисперсії для класу m , $m = \overline{1, M}$; N – кількість ознак; θ_i – усереднене значення яскравості пікселів, що входять до i -ї ознаки; $\bar{\theta}$ – усереднене значення яскравості кадру;

Подібний статистичний аналіз показує, що на рисунку 3.18 до X_1^0 відносяться 37 кадрів, до X_2^0 – 663. Після виконання обчислень маємо наступні значення середньої яскравості та дисперсії цих двох груп: $\theta_1 = 73,82$, $\theta_2 = 33,16$; $S_1 = 1516,20$, $S_2 = 290,19$;

Розрахуємо середньоквадратичне відхилення для показників усередненої яскравості кожного з кадрів відповідних класів, за наступною формулою:

$$\sigma_m = \sqrt{\frac{\sum_{i=1}^L (\theta_{m,i} - \mu_m)^2}{L}} \quad (3.2)$$

де σ_m – це середньо квадратичне відхилення для класу m , $m = \overline{1, M}$; L – це загальна кількість ідентифікованих кадрів; $\theta_{m,i}$ – це усереднене значення яскравості i -го кадру; μ_m – це усереднене значення яскравості для усіх кадрів відповідного класу;

У нашому випадку середньоквадратичне відхилення для обох класів має наступний вигляд: $\sigma_0 = 15,83$, $\sigma_1 = 15,60$.

3.4. Короткий опис програмної реалізації

Для написання алгоритму навчання та екзамену була використана мова програмування Python. Для зручності сприйняття опис коду, розробленого програмного забезпечення, занесений до таблиць і в Додаток.

Таблиця 3.2 Опис складових класу MachineLearning

Методи	
Назва	Призначення
def __init__(self, m)	Конструктор класу. Приймає на вхід кількість класів та запускає метод по завантаженню їх в пам'ять
def realization(self, delta):	Центральний метод, який по чергово запускає усі методи, які використовуються при навчанні системи
def loadImage(self):	Завантаження зображення, через вказання шляху до нього
def makeBitMap(self, fileName):	Безпосереднє завантаження зображення в пам'ять
def makeDopusk(self, delta):	Створення системи контрольних допусків
def makeBinaryMap(self, k):	Створення бінарної карти зображення
def makeEtalonVector(self, k):	Створення еталонних векторів
def makeSK(self, k):	Створення масиву кодових відстаней реалізацій
def makePara(self):	Пошук відстані між класами
def INFk(self, INFk_d, baseClass):	Розрахунок функціонального критерію
def makeRadius(self):	Пошук оптимальних радіусів
def averageKFE(self):	Усереднене значення критерію
def checkKFE(self, d, k):	Перевірка чи існує робоча область
def parallelOptim(self):	Алгоритм паралельної оптимізації СКД
def exam(self):	Алгоритм екзамену
def makeGrafics(self):	Побудова графіків

ВИСНОВКИ

1. У рамках роботи була реалізована бортова система розпізнавання образів. Зроблено це було за допомогою інформаційно-екстремальної технології машинного навчання. Для обраної моделі була виконана паралельна оптимізація системи контрольних допусків на ознаки розпізнавання. Це дозволило побудувати оптимальні геометричні параметри контейнерів класів розпізнавання.

2. Перевірка функціональної ефективності вирішальних правил була виконана під час етапу екзамену, де вдалося отримати достовірний результат класифікації кадрів зображення.

3. Для формування вхідного математичного опису, а також тестуванні розробленої бортової системи, був використаний багатофункціональний симулятор динамічного оточення, який дозволив реалізувати імітацію умов функціонування безпілотної, які наближені до реальних.

4. Був поставлений експеримент при якому зображення, яке отримує безпілотної, завдалося зовнішньому впливу, що призвело до погіршення результату класифікації кадрів. Було запропоновано декілька варіантів вирішення поставленої проблеми.

5. Реалізовано новий підхід до ідентифікації кадрів у рамках обраної технології. Вводиться додатковий статистичний квантор перевірки класифікації, який дозволяє підвищити релевантність вирішальних правил. У рамках роботи це дозволило зменшити вплив зовнішнього шуму на результат ідентифікації зображення

СПИСОК ЛІТЕРАТУРИ

1. Довбиш А. С. Основи проектування інтелектуальних систем: Навчальний посібник / А. С. Довбиш. – Суми: Видавництво Сум ДУ, 2009. – 171 с.
2. Abu-Jbara, Khaled, et al. "A robust vision-based runway detection and tracking algorithm for automatic UAV landing." 2015 International conference on unmanned aircraft systems (ICUAS). IEEE, 2015.
3. Quiroz, Ricardo A. Arango, Fernada Pereira Guidotti, and Albeiro Espinosa Bedoya. "A method for automatic identification of crop lines in drone images from a mango tree plantation using segmentation over YCrCb color space and Hough transform." 2019 XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA). IEEE, 2019.
4. Vayssade, Jehan-Antoine, Rémy Arquet, and Mathieu Bonneau. "Automatic activity tracking of goats using drone camera." Computers and Electronics in Agriculture 162 (2019): 767-772.
5. Protsenko O. Informational and extreme machine learning for onboard recognition system of ground objects / O. Protsenko, T. Savchenko, M. Myronenko, O. Prihodchenko // Proceedings - 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies, DESSERT, 2020. – P. 213-218.
6. Довбиш А.С. Інтелектуальні інформаційні технології в електронному навчанні / А.С. Довбиш, А.В. Васильєв, В.О. Любчак. – Суми: Видавництво СумДУ, 2013.– 177 с.
7. Савченко, Т.Р. Інформаційна технологія розпізнавання об'єктів. Машинне навчання бортової системи розпізнавання наземних об'єктів [Текст]: робота на здобуття кваліфікаційного ступеня бакалавра; спец.: 122 - комп'ютерні науки (інформатика) / Т.Р. Савченко; наук. кер. А.С. Довбиш. - Суми: СумДУ, 2021. - 67 с.

8. Гриненко, О.В. Інформаційна технологія розпізнавання об'єктів. Бортова система безпілотного авіаційного комплексу для розпізнавання природних та інфраструктурних об'єктів [Текст]: робота на здобуття кваліфікаційного ступеня бакалавра; спец.: 122 - комп'ютерні науки (інформатика) / О.В. Гриненко; наук. кер. А.С. Довбиш. - Суми: СумДУ, 2021. - 77 с.

9. Amazon Prime Air prepares for drone deliveries. <https://www.aboutamazon.com/news/transportation/amazon-prime-air-prepares-for-drone-deliveries>

10. ВИКОРИСТАННЯ ДРОНІВ У СІЛЬСЬКОМУ ГОСПОДАРСТВІ. <https://dronecenter.ua/review-of-the-dji-phantom-4-pro-quadcopter>

11. Tang, L., Shao, G. Drone remote sensing for forestry research and practices. J. For. Res. 26, 791–797 (2015). <https://doi.org/10.1007/s11676-015-0088-y>

12. Dong, Li, et al. "Analysis on the Characteristics of Air Pollution in China during the COVID-19 Outbreak." Atmosphere 12.2 (2021): 205.

13. Kratos Defense & Security Solutions. <https://www.kratosdefense.com/>

14. Baykar Makina. <https://baykartech.com/en/>

15. KARGU Combat Proven Rotary Wing Loitering Munition System. <https://www.stm.com.tr/en/kargu-autonomous-tactical-multi-rotor-attack-uav>

16. Konert, A., & Balcerzak, T. (2021). Military autonomous drones (UAVs) - from fantasy to reality. Legal and Ethical implications. Transportation Research Procedia, 59, 292–299. <https://doi.org/10.1016/j.trpro.2021.11.121>

17. БпАК – RAYBIRD 3. <https://skyeton.com/raybird-3>

18. Naumenko, I., Piatachenko, V., Myronenko, M., & Savchenko, T. (2022). Information-Extreme Machine Learning of an On-board Ground Object Recognition System with a Choice of a Base Recognition Class.

19. Naumenko, Igor, Mykyta Myronenko, and Taras Savchenko. "Information-extreme machine training of on-board recognition system with optimization of RGB-component digital images." Radioelectronic and Computer Systems 4 (2021): 59-70.

20. Dovbysh, A., Naumenko, I., Myronenko, M., & Savchenko, T. (2020). Information-extreme machine learning on-board recognition system of ground objects with the adaptation of the input mathematical description. In CMIS (pp. 913-925).

21. Possible First Use of AI-Armed Drones Triggers Alarm Bells. https://www.voanews.com/a/africa_possible-first-use-ai-armed-drones-triggers-alarm-bells/6206728.html

ДОДАТОК

```

from os import listdir
import numpy as np
from PIL import Image, ImageDraw
import math
import matplotlib.pyplot as plt

from tkinter import Tk
from tkinter.filedialog import askopenfilename

class machineLearning:
    #кількість ознак і реалізацій
    N = 0
    n = 0
    #кількість класів
    m = 0
    #список, який зберігає значення яскравості класів
    brightness = []
    #верхні та нижні допуски
    vd = []
    nd = []
    #список, який зберігає бінарні карти зображень
    binaryMap = []
    #еталонні вектори
    ev = []
    #масив кодових відстаней
    #1-відстань від сусідньої ознаки до поточного центру
    #2-відстань від своєї ознаки до свого центру
    #3-відстань від сусідньої ознаки до сусіднього центру
    #4-відстань від своєї ознаки до сусіднього центру
    sk = []
    #відстань між класами
    paraD = 0
    #значення радіусів
    my_do = []
    #значення критерію
    em = []
    #значення дельта
    delta = 37
    #конструктор класу
    def __init__(self, m):
        self.m = m
        #завантажуємо класи
        for i in range(self.m):
            self.makeBitMap(self.loadImage())

    def realization(self, delta):
        self.sk = []
        self.binaryMap = []
        self.ev = []

        self.makeDopusk(delta)
        #створення бінарної карти класів
        for i in range(self.m):

```

```

        self.makeBinaryMap(i)
        self.makeEtalonVector(i)
self.makePara()
self.makeRadius()

#обрання шляху до зображення
def loadImage(self):
    Tk().withdraw()
    fileName = askopenfilename(
        filetypes=(("BMP files", "*.bmp"),
                   ("All files", "*.*")))
    return fileName

#завантаження зображення
def makeBitMap(self, fileName):
    #Відкриваємо зображення
    image = Image.open(fileName)
    #розмір зображення
    self.N = image.size[0]
    self.n = image.size[1]
    #завантажуємо пікселі
    pix = image.load()

    array = np.zeros((self.N, self.n))
    for x in range(self.N):
        for y in range(self.n):
            array[x, y] = pix[x, y]
    self.brightness.append(array)

#створення СКД
def makeDopusk(self, delta):
    self.vd = np.zeros((self.N))
    self.nd = np.zeros((self.n))

    for i in range(self.N):
        sum = 0
        for j in range(self.n):
            sum += self.brightness[0][i,j]
        sum //= self.n

        self.vd[i] = sum + delta
        self.nd[i] = sum - delta

#створення бінарної карти зображення
def makeBinaryMap(self, k):
    array = np.zeros((self.N, self.n))
    for i in range(self.N):
        for j in range(self.n):
            if self.nd[i] <= self.brightness[k][i, j] <= self.vd[i]:
                array[i, j] = 1
    self.binaryMap.append(array)

#створення еталонних векторів
def makeEtalonVector(self, k):
    array = np.zeros((self.N))
    for i in range((self.N)):

```

```

        evSum = 0
        for j in range(self.n):
            evSum += self.binaryMap[k][i, j]
        if evSum / self.n >= 0.5:
            array[i] = 1
        self.ev.append(array)

#створення масиву кодових відстаней реалізацій
def makeSK(self, k):
    self.sk = np.zeros((2, self.n))
    k2 = 0
    if k == 0:
        k2 = 1
    for j in range(self.n):
        for i in range(self.N):
            #відстань від сусідньої ознаки до поточного центру
            if self.ev[k][i] != self.binaryMap[k2][i, j]:
                self.sk[1, j] += 1
            #відстань від своєї ознаки до свого центру
            if self.ev[k][i] != self.binaryMap[k][i, j]:
                self.sk[0, j] += 1

#пошук відстані між класами
def makePara(self):
    for i in range(self.N):
        if self.ev[0][i] != self.ev[1][i]:
            self.paraD += 1

#розрахунок функціонального критерію
def INFK(self, INFK_d, baseClass):
    k1 = 0
    k4 = 0
    #підрахунок реалізацій
    for INFK_j in range(self.n):
        #кількість власних реалізацій в контейнері
        if self.sk[0, INFK_j] <= INFK_d:
            k1 += 1
        #кількість сусідніх реалізацій в контейнері
        if self.sk[1, INFK_j] <= INFK_d:
            k4 += 1
    #характеристики
    #перша достовірність
    INFK_d1 = float(k1)/float(self.n)
    #помилка другого роду
    INFK_beta = float(k4)/float(self.n)

    d1_b = INFK_d1 - INFK_beta
    #критерій Кульбака
    kullback = d1_b * math.log((1 + d1_b + 0.1) / (1 - d1_b + 0.1)) /
math.log(2)
    return kullback, INFK_d1, INFK_beta

allEMR = []
trueEMR = []
#пошук оптимальних радіусів
def makeRadius(self):

```

```

self.allEMR = []
self.trueEMR = []

self.my_do = np.zeros((self.m))
self.em = np.zeros((self.m))
for k in range(self.m - 1, -1, -1):
    #значення критерію для кожного радіусу
    allEM = np.zeros((self.N))
    #значення критерію в робочій області
    trueEM = np.zeros((self.N))
    self.makeSK(k)
    for d in range(self.N):
        #розрахунок КФЕ для кожного радіусу
        eK, t_d1, t_beta = self.INFK(d, k)
        if eK >= self.em[k] and t_d1 >= 0.5 and d < self.paraD:
            if eK > self.em[k]:
                self.my_do[k] = d
                self.em[k] = eK
                trueEM[d] = eK
            allEM[d] = eK
    self.allEMR.append(allEM)
    self.trueEMR.append(trueEM)

#усереднене значення критерію
def averageKFE(self):
    avr = 0
    for k in range(self.m):
        avr += self.em[k]
    avr /= self.m
    return avr

#перевірка чи існує робоча область
def checkKFE(self, d, k):
    eK, t_d1, t_beta = self.INFK(d, k)
    if t_d1 >= 0.5 and t_beta <= 0.5:
        return True
    return False

limitD = 80
emAll = []
emTrue = []
#алгоритм паралельної оптимізації СКД
def parallelOptim(self):
    emLast = 0

    self.emAll = np.zeros((self.limitD))
    self.emTrue = np.zeros((self.limitD))
    for delta in range(self.limitD):
        self.realization(delta)

        emNow = self.averageKFE()
        if self.checkKFE(self.my_do[0], 0):
            self.emTrue[delta] = emNow
            self.emAll[delta] = emNow

        if emNow > emLast and self.checkKFE(self.my_do[0], 0):

```

```

        emLast = emNow
        self.delta = delta
#фінальний етап навчання з оптимальним дельта
machine.realization(self.delta)

#алгоритм екзамену
def exam(self):
    #створення екзаменаційної навчальної матриці
    self.makeBitMap(self.loadImage())
    self.makeBinaryMap(self.m)

    res = np.zeros((self.m))
    for k in range(self.m):
        for j in range(self.N):
            d_exam = 0
            for i in range(self.n):
                d_exam += math.fabs(self.ev[k][i] -
self.binaryMap[self.m][i, j])
            res[k] += 1 - (d_exam / self.my_do[k])
        res[k] /= self.n

    k_max = -1
    mu_max = 0
    for i in range(machine.m):
        if mu_max < res[i]:
            mu_max = res[i]
            k_max = i
    return k_max

#побудова графіків
def makeGrafics(self):
    fig, ax = plt.subplots(figsize=(5, 3))
    x = list(range(0, self.N))

    plt.subplot(2, 2, 1)
    plt.stackplot(x, self.trueEMR[0], color="blue")
    plt.plot(x, self.allEMR[0], color="black")
    plt.title('class 1', fontsize=15)

    plt.subplot(2, 2, 2)
    plt.stackplot(x, self.trueEMR[1], color="blue")
    plt.plot(x, self.allEMR[1], color="black")
    plt.title('class 2', fontsize=15)

    plt.subplot(2, 1, 2)
    x = list(range(0, self.limitD))
    plt.stackplot(x, self.emTrue, color="blue")
    plt.plot(x, self.emAll, color="black")
    plt.title('Paralleloptimization', fontsize=15)

    plt.show()

machine = machineLearning(2)
machine.parallelOptim()
res = machine.exam()
machine.makeGrafics()

```