

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ЦЕНТР ЗАОЧНОЇ, ДИСТАНЦІЙНОЇ ТА ВЕЧІРНЬОЇ ФОРМ НАВЧАННЯ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Кваліфікаційна робота магістра

**На тему:**

**«ІНФОРМАЦІЙНИЙ СИНТЕЗ БОРТОВОЇ СИСТЕМИ  
АВТОНОМНОГО БЕЗПЛОТНОГО ЛІТАЛЬНОГО АПАРАТУ ДЛЯ  
РОЗПІЗНАВАННЯ СТАНУ АГРОКУЛЬТУР»**

Здобувач освіти гр. ІН.мз – 13с

Артем СУЛІМ

Науковий керівник,  
професор, д.т.н.

Анатолій ДОВБИШ

Завідувач кафедри  
доцент, к.т.н.

Ігор ШЕЛЕХОВ

Суми 2023

Сумський державний університет

(назва вузу)

Факультет ЦЗДВН Кафедра Комп'ютерних наукСпеціальність «122 - Комп'ютерні науки»

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_

« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТА

Суліма Артема Валерійовича

(прізвище, ім'я, по батькові)

1. Тема роботи «Інформаційний синтез бортової системи автономного безпілотного літального апарату для розпізнавання стану агрокультур»

затверджена наказом по університету від « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р. № \_\_\_\_\_

2. Термін здачі студентом закінченої роботи \_\_\_\_\_

3. Вихідні дані роботи \_\_\_\_\_

Зображення ділянок агрокультур, отримані за результатами

аерофотозйомки \_\_\_\_\_

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

1. Аналітичний огляд застосування безпілотних літальних апаратів в сільському господарстві.

2. Аналіз методів машинного навчання та розпізнавання наземних об'єктів.

3. Основні положення методу досліджень.

4. Формування вхідних даних.

5. Розробка алгоритму інформаційно-екстремального машинного навчання бортової системи

картографування (сегментації) агрокультур.

6. Алгоритм функціонування бортової системи розпізнавання в режимі екзамену.

7. Короткий опис програмного комплексу.

8. Результати комп'ютерного моделювання.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) \_Слайди: 17

1)Актуальність. 2)Мета, об'єкт і предмет дослідження.3) Функціональні категорійні моделі машинного навчання. 4) Схема алгоритму машинного навчання. 5) Інформаційний критерій оптимізації параметрів машинного навчання. 6) Результати комп'ютерного моделювання.

7) Висновки.

6. Консультанти по роботі, із зазначенням розділів роботи, що стосуються їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання \_\_\_\_\_

Керівник \_\_\_\_\_  
(підпис)

Завдання прийняв до виконання \_\_\_\_\_  
(підпис)

### КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів магістерської кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Приклади застосування БПЛА у сільському господарстві		
2	Аналітичний огляд методів машинного навчання та розпізнавання.		
3	Основні положення методу досліджень.		
4	Розроблення алгоритмів машинного навчання.		
5	Програмна реалізація алгоритмів машинного навчання.		
6	Аналіз результатів та оформлення КМР		
7	Захист КМР		

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

## РЕФЕРАТ

**Записка:** 51 стор., 15 рис., 2 таблиці, 1 додаток., 15 джерел.

**Об'єкт дослідження** — створення повністю автономної бортової системи не керованого безпілотного літального апарату для розпізнавання різних станів агрокультур.

**Мета роботи** – синтез автоматизованої бортової системи не керованого безпілотного літального апарату для сегментації агрокультур з метою спостереження за їх розвитком в період вегітації.

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

**Результати** – розроблено та реалізували програмно алгоритми інформаційно-екстремального машинного навчання бортової системи автономного безпілотного літального апарату (БПЛА) для сегментації агрокультур з метою спостереження за їх розвитком у період вегітації.

КЛЮЧОВІ СЛОВА: ІНФОРМАЦІЙНО\_ЕКСТРЕМАЛЬНЕ МАШИННЕ  
НАВЧАННЯ, БОРТОВА СИСТЕМА РОЗПІЗНАВАННЯ, СЕГМЕНТАЦІЯ,  
ІНФОРМАЦІЙНИЙ КРИТЕРІЙ, СТАН АГРОКУЛЬТУРИ,  
ВИРІШАЛЬНІ ПРАВИЛА

## ЗМІСТ

<b>ВСТУП</b> .....	6
<b>РОЗДІЛ 1 ІНФОРМАЦІЙНИЙ ОГЛЯД</b> .....	7
1.1. Огляд сучасних агродронів.....	7
1.2. Огляд сучасних систем аналізу геоінформаційних даних.....	11
1.3. Огляд методів інтелектуального аналізу даних.....	13
<b>РОЗДІЛ 2 ОПИС МЕТОДУ ДОСЛІДЖЕННЯ</b> .....	17
2.1. Основні положення інформаційно-екстремальної інтелектуальної технології аналізу даних .....	17
2.2. Оцінка функціональної ефективності здатної навчатися системи розпізнавання .....	20
<b>РОЗДІЛ 3 ІНФОРМАЦІЙНЕ, АЛГОРИТМІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ БОРТОВОЇ СИСТЕМИ РОЗПІЗНАВАННЯ</b> .....	22
3.1. Вхідний інформаційний опис.....	22
3.2. Машинне навчання бортової системи автономного безпілотного літального апарату з оптимізацією контрольних допусків на ознаки розпізнавання.....	25
3.3. Функціонування автономної бортової системи в режимі екзамену	31
3.4. Короткий опис програмного забезпечення .....	34
3.5. Результати комп'ютерного моделювання.....	36
<b>ВИСНОВКИ</b> .....	40
<b>СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ</b> .....	41
<b>ДОДАТОК А</b> .....	43

## ВСТУП

У наш час дрони-безпілотники займаються всім, від боротьби з хворобами до прибирання сміття в океані, доставки піци і багато чого іншого. Технологія безпілотників вже давно використовується оборонними організаціями і технічно підкованими споживачами. Однак переваги цієї технології виходять далеко за рамки тільки цих секторів. З ростом доступності дронів багато з найнебезпечніших і високооплачуваних робочих місць в комерційному секторі дозріли для заміщення їх літаючими дронами. Варіанти використання безпечних і економічно ефективних рішень за участю дронів варіюються від збору даних до доставки. У міру вдосконалення технологій автономії і запобігання зіткнень здатність дронів виконувати все більш складні завдання зростає.

За даними PwC, глобальний ринок бізнес-послуг, що розвивається, з використанням дронів оцінюється більш ніж в 127 мільярдів доларів. І оскільки все більше корпорацій прагнуть отримати вигоду з цих комерційних можливостей, інвестиції в розробку безпілотників зросли.

Безпілотний літальний апарат – це апарат, що працює за допомогою комбінації технологій, включаючи комп'ютерний зір, штучний інтелект, технологію обходу об'єктів та інші. Але безпілотники також можуть бути наземними або морськими транспортними засобами, що працюють автономно.

Основна мета кваліфікаційної магістерської роботи це розробка бортової системи розпізнавання (БСР) автономного БПЛА для сегментації агрокультур з метою спостереження за їх розвитком в період вегетації.

# РОЗДІЛ 1

## ІНФОРМАЦІЙНИЙ ОГЛЯД

### 1.1. Огляд сучасних агро-дронів

Сільськогосподарські дрони, також відомі як СГ-дрони або агроботи, — це безпілотні літальні апарати (БПЛА), призначені для використання в сільському господарстві [1, 2]. Вони оснащені високотехнологічними датчиками та камерами, які можуть збирати цінні дані та виконувати різноманітні завдання, такі як картографування посівів, аналіз ґрунту, моніторинг стану рослин, керування зрошувальною системою тощо. Ці безпілотники дають фермерам можливість швидко та ефективно збирати інформацію про свої поля, даючи їм уявлення про їхні врожаї та ґрунт, що може допомогти їм прийняти обґрунтовані рішення про те, як найкраще доглядати за своєю землею.



Рисунок 1.1 – Дрон картограф XAG XMission M500

Одним із найважливіших застосувань сільськогосподарських дронів є

картографування або сегментація посівів . Ці дрони можуть літати над полями, знімаючи зображення високої роздільної здатності, які можна використовувати для створення детальних карт посівів і ґрунту. Ця інформація може бути використана для визначення ділянок поля, які зазнають стресу, або для створення запису про зростання врожаю з часом. Крім того, це може допомогти фермерам спланувати більш ефективні системи зрошення, а також оцінити врожайність сільськогосподарських культур і зробити прогноз щодо майбутніх врожаїв (рис. 1.2) [1].



Рисунок 1.2 – Дрон картограф XAG XMission M500 в роботі



Інше важливе застосування сільськогосподарських дронів – аналіз ґрунту (рис. 1.3). Ці дрони можуть бути оснащені датчиками, які можуть вимірювати поживні речовини в ґрунті та допомагати ідентифікувати ділянки, де бракує життєво важливих мінералів. Потім цю інформацію можна використовувати для підвищення родючості ґрунту та сприяння здоровому росту рослин.

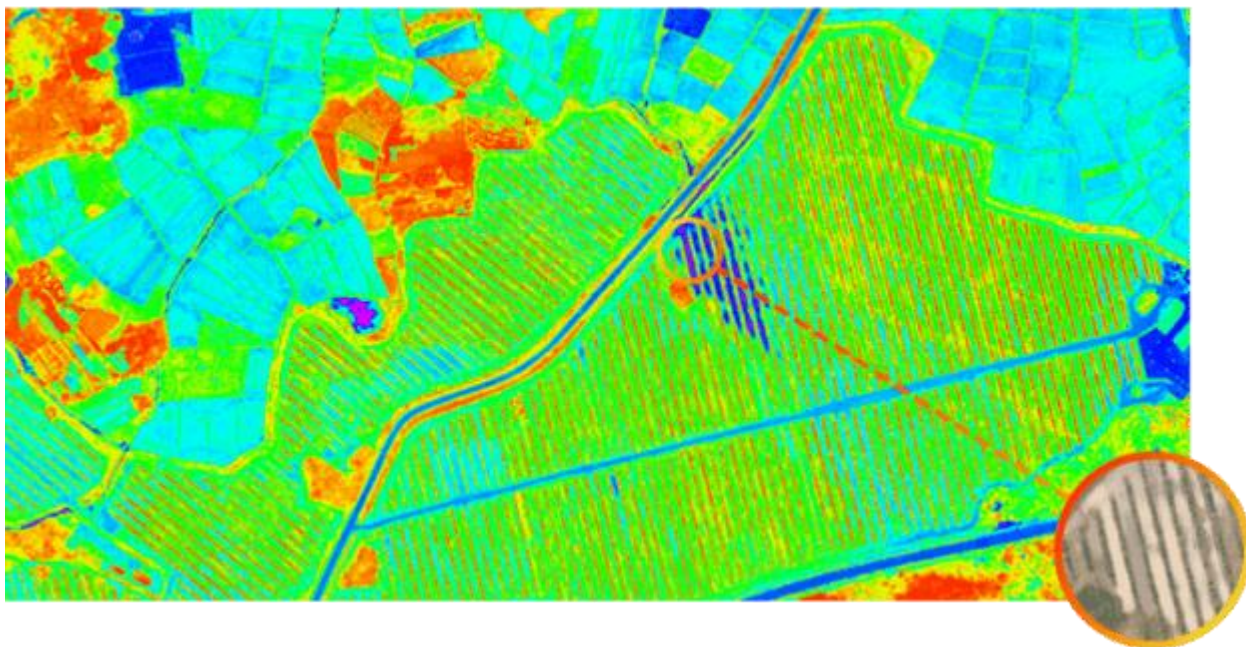


Рисунок 1.3 – Фото з дрона картографа XAG XMission M500  
(зелений – нормальний ґрунт, синій – аномальний ґрунт)

Окрім збору даних та аналізу, сільськогосподарські дрони також можна використовувати для керування зрошувальними системами (рис.1.4). Використовуючи датчики для вимірювання рівня вологості ґрунту, дрони можуть допомогти фермерам визначити, коли зрошувати свої культури та скільки води використовувати. Це допомагає зберегти воду та підвищити врожайність, і може бути особливо корисним у регіонах, де води не вистачає.



Рисунок 1.4 – Дрон-обприскувач XAG V40 в роботі

Використання сільськогосподарських дронів стає все більш поширеним, оскільки все більше фермерів використовують цю технологію для покращення своєї діяльності. Ці безпілотники допомагають революціонізувати спосіб управління сільським господарством, надаючи фермерам цінну інформацію про їхні культури та ґрунт, а також допомагаючи їм приймати обґрунтовані рішення, які призведуть до здоровішого врожаю, підвищення врожайності та підвищення стійкості.

Підсумовуючи, сільськогосподарські дрони є потужним інструментом для сучасних фермерів. Вони надають цінні дані та розуміння сільськогосподарських культур і ґрунту, а також допомагають фермерам приймати обґрунтовані рішення про те, як найкраще доглядати за своєю землею. З постійним прогресом у технологіях безпілотних літальних апаратів, ймовірно, що роль сільськогосподарських безпілотних літальних

апаратів буде тільки зростати в найближчі роки.

## 1.2. Огляд сучасних систем аналізу геоінформаційних даних

Геоінформаційні системи (ГІС) — це комп'ютерні інструменти для збору, зберігання, аналізу та візуалізації просторових даних. Технологія ГІС розвивалася протягом останніх кількох десятиліть, і сьогодні сучасні ГІС-системи здатні обробляти великі обсяги даних із різних джерел, включаючи супутникові зображення, аерофотознімки та наземні спостереження.

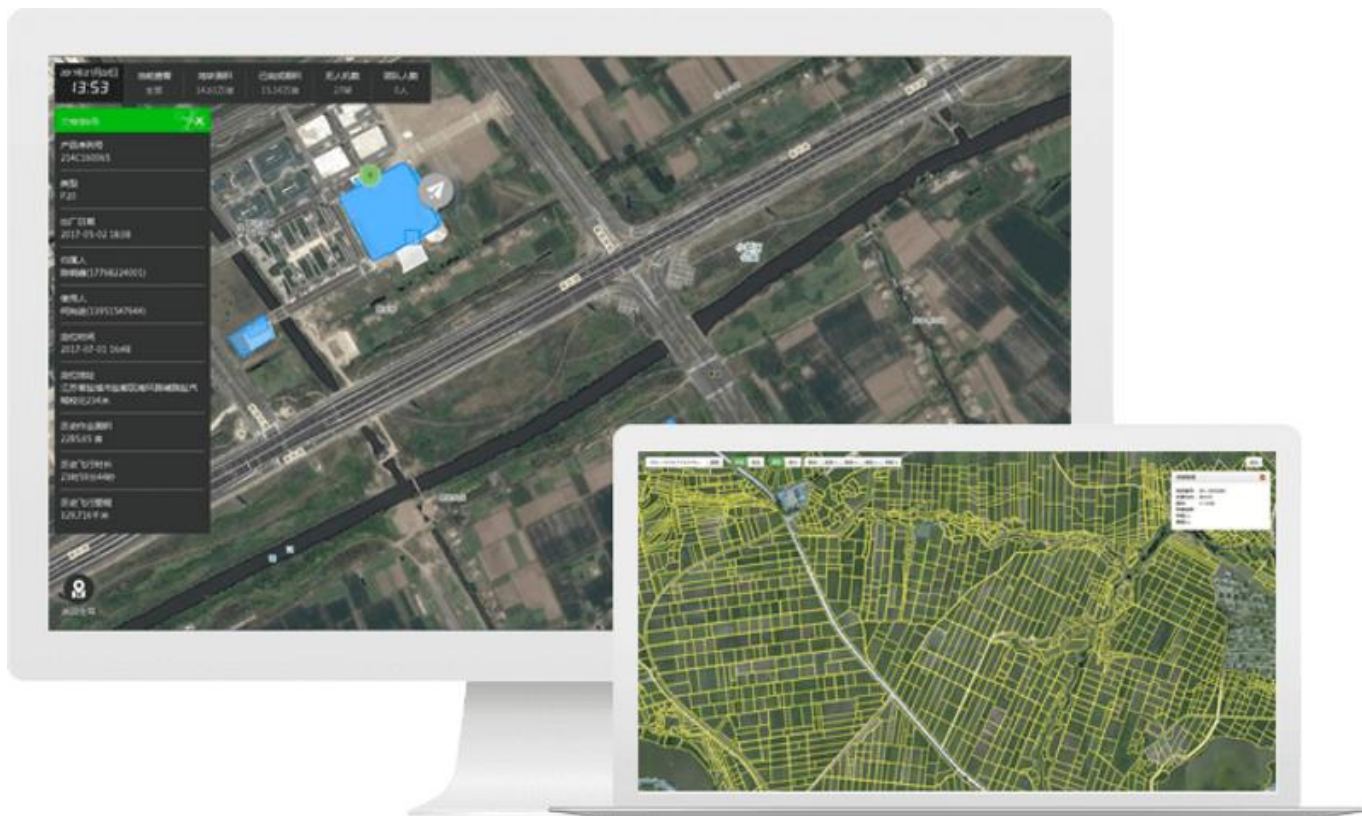


Рисунок 1.5 – XFIS (система польотної інформації XAG)

XFIS, розроблена компанією XAG [3], є системою управління літаками, яка дозволяє здійснювати моніторинг у реальному часі, візуалізувати інформацію про польоти та переглядати історичні дані про літаки XAG. Система також підтримує функцію електронного огороження,



налаштування забороненої для польотів зони та функції дистанційного блокування відповідно до національної безпеки та стороннього нагляду за польотними даними

Сучасна система аналізу геоінформаційних даних зазвичай складається з таких компонентів [3]:

1. Збір даних і керування ними: це передбачає отримання просторових даних із різних джерел, таких як датчики, бази даних і краудсорсингові платформи. Потім дані впорядковуються, очищаються та зберігаються в базі даних або системі ГІС.
2. Аналіз даних: це передбачає використання алгоритмів, статистичних методів і моделей машинного навчання для отримання інформації з просторових даних. Дані можна аналізувати, щоб виявити закономірності, тенденції та зв'язки між різними характеристиками та змінними.
3. Візуалізація: це передбачає використання карт, графіків та інших візуальних зображень даних для передачі думок і висновків ширшій аудиторії. Програмне забезпечення ГІС надає ряд інструментів візуалізації, включаючи 3D-моделювання, анімацію та віртуальну реальність.
4. Підтримка прийняття рішень: це передбачає використання технології ГІС для підтримки процесів прийняття рішень шляхом надання інформації та рекомендацій на основі аналізу даних. Це можна використовувати в різних сферах, таких як міське планування, управління навколишнім середовищем і планування реагування на надзвичайні ситуації.
5. Співпраця та обмін: це передбачає обмін даними, ідеями та результатами між зацікавленими сторонами, включаючи державні установи, академічні установи та організації приватного сектору.

Сучасні ГІС-системи підтримують співпрацю через веб-платформи, хмарні обчислення та мобільні пристрої.

Загалом сучасна система аналізу геоінформаційних даних змінила наш спосіб розуміння та управління світом навколо нас. Він надає потужний інструмент для прийняття обґрунтованих рішень, вирішення складних проблем і передачі результатів у змістовний і доступний спосіб.

### **1.3. Огляд методів інтелектуального аналізу даних**

Інтелектуальний аналіз даних у сільському господарстві передбачає використання передових обчислювальних методів і алгоритмів для аналізу великих наборів даних, створених різними джерелами, такими як супутникові зображення, сенсорні мережі, метеостанції та агрономічні бази даних. Метою цього аналізу є покращення виробництва сільськогосподарських культур і врожайності, зменшення відходів і витрат на виробництво, оптимізація використання ресурсів і забезпечення продовольчої безпеки.

Серед сучасних інтелектуальних алгоритмів аналізу даних, які використовуються в різних програмах і галузях, виділяють [4]:

1. Штучні нейронні мережі (ШНМ): це алгоритми машинного навчання, змодельовані за структурою та функціями людського мозку. ШНМ використовуються для розпізнавання зображень і мови, обробки природної мови та інших складних завдань аналізу даних.
2. Глибоке навчання: глибоке навчання — це підполе машинного навчання, яке використовує штучні нейронні мережі з кількома прихованими шарами для складних завдань аналізу даних, таких як класифікація зображень, розпізнавання мовлення та обробка природної мови.
3. Деревя рішення. Деревя рішення — це деревоподібні моделі, які використовують низку рішень і їхні можливі наслідки для аналізу

даних і прогнозування результатів. Вони зазвичай використовуються для завдань класифікації та регресії.

4. Випадковий ліс: Випадковий ліс — це метод ансамблевого навчання, який створює кілька дерев рішень і поєднує їхні результати, щоб зробити прогноз. Цей метод може зменшити переобладнання, підвищити точність передбачення та обробляти складні набори даних.
5. Мащини підтримки векторів (анг. Support Vector Machines або скорочено SVM): — це тип алгоритму машинного навчання, який використовується для завдань класифікації та регресії. Вони працюють, знаходячи лінію або гіперплощину, яка розділяє точки даних на різні класи.
6. K-Means Clustering: K-Means Clustering — це алгоритм неконтрольованого навчання, який групує точки даних у кластери на основі подібності. Він зазвичай використовується для сегментації ринку, стиснення зображень і виявлення аномалій.

Це лише кілька прикладів сучасних інтелектуальних алгоритмів аналізу даних. Вибір алгоритму залежатиме від конкретного завдання аналізу даних і вимог програми.

У сільському господарстві існує кілька основних методів інтелектуального аналізу даних, зокрема:

1. Точне землеробство: цей метод використовує дані в реальному часі та технології картографування, щоб надати детальну інформацію про стан ґрунту, погоду та врожай у масштабі поля. Ця інформація використовується для прийняття обґрунтованих рішень щодо посадки, удобрення, зрошення та боротьби зі шкідниками.
2. Прогнозування врожайності: це передбачає використання алгоритмів машинного навчання для аналізу даних про врожайність культур і прогнозування майбутньої врожайності на основі таких факторів, як погода, якість ґрунту та стан рослин.

3. Класифікація культур: цей метод передбачає використання даних дистанційного зондування, таких як супутникові зображення, для класифікації культур на основі їх стадії росту, типу культури та загального стану. Цю інформацію можна використовувати для прийняття обґрунтованих рішень щодо управління рослинництвом і визначення територій, які потребують уваги.
4. Прогнозування погоди: цей метод використовує історичні дані про погоду та математичні моделі для прогнозування майбутніх погодних умов та їхнього впливу на посіви. Цю інформацію можна використовувати для прийняття рішень щодо посадки, зрошення та боротьби зі шкідниками.
5. Аналіз ґрунту: цей метод передбачає використання датчиків і алгоритмів машинного навчання для аналізу даних про якість ґрунту та надання рекомендацій щодо внесення добрив і управління поживними речовинами.

Все частіше і частіше для виконання задач з багатопараметричної оптимізації та адаптивного пошуку починають використовуватися генетичні алгоритми (ГА). В них використовують базові принципи біологічної спадковості та природного відбору, через це ГА може виконувати випадковий пошук глобального екстремуму. Також досить часто генетичні алгоритми по'єднують з іншими методами, наприклад, для попереднього аналізу вхідних даних, що і дозволяє побудувати більш оптимальний класифікатор.

Генетичні алгоритми можна використовувати в сільськогосподарських дронах для оптимізації різних аспектів роботи дронів, таких як схеми польоту, схеми обприскування та застосування пестицидів. Ось кілька прикладів:

1. Оптимізація схем польоту: можна використовувати генетичні алгоритми для оптимізації схем польоту сільськогосподарських дронів, щоб охопити максимальну площу з мінімальним перекриттям

і мінімальним часом польоту. Алгоритми також можуть допомогти знайти найефективніший спосіб перельоту з одного поля на інше, зменшуючи час і споживання енергії дроном.

2. Оптимізація схеми обприскування: під час обприскування сільськогосподарських культур можна використовувати генетичні алгоритми для оптимізації застосування пестицидів та інших хімікатів. Наприклад, алгоритми можуть допомогти визначити ідеальну швидкість покриття, найкращу орієнтацію та висоту для польоту дрона, а також ідеальне перекриття між послідовними циклами розпилення для забезпечення рівномірного покриття.
3. Оптимізація застосування пестицидів: використання генетичних алгоритмів також може допомогти оптимізувати застосування пестицидів. Наприклад, алгоритми можуть допомогти визначити ідеальну кількість пестициду для застосування на основі типу культури, умов поля та погодних умов.

Генетичні алгоритми можуть підвищити ефективність і ефективність роботи сільськогосподарських безпілотників, що призведе до кращої врожайності, зниження витрат і зменшення впливу на навколишнє середовище.

Загалом, використання інтелектуального аналізу даних у сільському господарстві стає все більш важливим, оскільки населення світу продовжує зростати, а попит на продукти харчування продовжує зростати. Ці методи дозволяють точніше та ефективніше використовувати ресурси, покращувати виробництво сільськогосподарських культур і покращувати продовольчу безпеку для громад у всьому світі.



## РОЗДІЛ 2

### ОПИС МЕТОДУ ДОСЛІДЖЕННЯ

#### 2.1. Основні положення інформаційно-екстремальної інтелектуальної технології аналізу даних

Інформаційно-екстремальна інтелектуальна технологія (ІЕІ-технологія) являється перспективним підходом для створення та аналізу бортових геоінформаційних систем (ГІС) розпізнавання об'єктів, що зазвичай передбачають використання розширених алгоритмів, методів машинного навчання та інструментів великих даних для аналізу величезних обсягів даних і отримання з них важливої інформації. До основних положень цієї технології відносяться [5 – 7]:

1. Збір даних. Першим кроком в аналізі даних є збір відповідних даних із різних джерел, таких як бази даних, веб-сайти та датчики. Потім зібрані дані обробляються, очищаються та перетворюються в придатний для використання формат.
2. Попередня обробка даних: цей крок передбачає видалення нерелевантних, повторюваних і неузгоджених даних, а також обробку відсутніх значень. Мета цього кроку полягає в тому, щоб переконатися, що дані готові для аналізу, і мінімізувати ризик отримання неправильних висновків.
3. Дослідження даних: цей крок передбачає візуалізацію та узагальнення даних, щоб отримати уявлення про їх характеристики, такі як розподіл, зв'язки та закономірності. Мета цього кроку — зрозуміти дані та виявити потенційні проблеми чи проблеми.
4. Моделювання даних: цей крок передбачає вибір і застосування статистичних моделей і моделей машинного навчання до даних, щоб робити прогнози або класифікувати дані за різними категоріями. Цей крок є критично важливим для прийняття обґрунтованих рішень і

розробки систем прогнозування.

5. Візуалізація даних: цей крок передбачає створення візуальних представлень даних, таких як діаграми, графіки та карти, щоб допомогти визначити закономірності, тенденції та зв'язки. Цей крок важливий для інформування зацікавлених сторін про результати аналізу.
6. Перевірка даних: цей крок передбачає перевірку точності та надійності аналізу та моделей. Цей крок має вирішальне значення для того, щоб результати аналізу були достовірними та надійними.

Загалом основна фішка інформаційно-екстремального (ІЕ) машинного навчання укладається в адаптації вхідного математичного опису для максимальної достовірності розпізнавання зображення. За досягненими, в процесі машинного навчання, оптимальними геометричними параметрами контейнерів класів розпізнавання синтезуються фінальні правила, які дозволяють приймати достовірні рішення в реальному часі.

Під час машинного навчання за ІЕІ-технологією здійснюється оптимізація різних параметрів за операцією багатоциклічного ітераційного пошуку усередненого за алфавітом класів розпізнавання інформаційного критерію [5]:

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

де  $E_m^{(k)}$  - обчислений на  $k$ -му кроці навчання інформаційний критерій оптимізації параметрів контейнеру класу  $X_m^o$ ;

$G_{\xi}$  - допустима область значень  $\xi$ -ї ознаки розпізнавання;

$G_E$  - допустима область визначення функції інформаційного критерію оптимізації параметрів навчання.

На алгоритм інформаційно-екстремального машинного навчання (2.1) накладаються такі обмеження [5]:

$$\begin{aligned}
& (\forall X_m^o \in \tilde{\mathfrak{R}}^{|M|}) [X_m^o \neq \emptyset]; \\
& (\exists X_m^o \in \tilde{\mathfrak{R}}^{|M|}) (\exists X_c^o \in \tilde{\mathfrak{R}}^{|M|}) [X_m^o \neq X_c^o \rightarrow X_m^o \cap X_c^o \neq \emptyset]; \\
& (\forall X_m^o \in \tilde{\mathfrak{R}}^{|M|}) (\forall X_c^o \in \tilde{\mathfrak{R}}^{|M|}) [X_m^o \neq X_c^o \rightarrow \text{Ker} X_m^o \cap \text{Ker} X_c^o \neq \emptyset], \\
& (\forall X_m^o \in \tilde{\mathfrak{R}}^{|M|}) (\forall X_c^o \in \tilde{\mathfrak{R}}^{|M|}) [X_m^o \neq X_c^o \rightarrow (d_m^* < d(x_m \oplus x_c)) \& (d_m^* < d(x_m \oplus x_c))]. \\
& \bigcup_{X_m^o \in \tilde{\mathfrak{R}}} X_m^o \subseteq \Omega; \quad m \neq c; \quad m, c = \overline{1, M},
\end{aligned} \tag{2.2}$$

де  $\tilde{\mathfrak{R}}^{|M|}$  - в загальному випадку нечітке розбиття простору ознак на класи розпізнавання;

$\text{Ker} X_m^o, \text{Ker} X_c^o$  - ядра найближчих сусідніх класів  $X_m^o$  і  $X_c^o$  відповідно;

$d(x_m \oplus x_c)$  - кодова відстань між еталонними (усередненими) векторами-реалізаціями  $x_m \in X_m^o$  і  $x_c \in X_c^o$ ;

$\Omega$  - простір ознак розпізнавання.

Умови (2.2) допускають перетин класів розпізнавання, але не допускають перетину їх ядер, щоб не було поглинання одним класом розпізнавання своїх найближчих сусідів.

За одержаними в процесі ІЕ машинного навчання оптимальними, в інформаційному розумінні геометричними, параметрами контейнерів класів розпізнавання синтезуються детерміновані фінальні правила.

У багатоциклічній процедурі (2.1) стандартний алгоритм ІЕ машинного навчання здійснюється у внутрішньому циклі оптимізації параметрів навчання. Основною функцією базового алгоритму на кожному кроці навчання є обчислення інформаційного критерію оптимізації і організація пошуку його глобального максимуму в робочій області визначення функції критерію. При цьому екстремальні значення параметрів машинного навчання

визначають відповідні оптимальні геометричні параметри розбиття простору ознак на класи розпізнавання.

## 2.2. Оцінка функціональної ефективності здатної навчатися системи розпізнавання

Основним питанням інформаційного синтезу системи розпізнавання зображень в рамках ІЕІ-технології, є оцінка функціональної ефективності процесу машинного навчання, яка визначає максимально можливу достовірність рішень, що приймаються на екзамені. В ІЕІ-технології можуть використовуватися різні критерії оптимізації параметрів машинного навчання, що задовольняють властивостям інформаційних мір.

Поміж інформаційних мір найбільшого використання знайшли ентропійна міра Шеннона та інформаційна міра Кульбака. Модифікована інформаційна міра Кульбака для двохальтернативних апріорно рівноймовірних гіпотез має вигляд [6]:

$$E_{K,m}^{(k)} = \log_2 \left( \frac{2 - (\alpha_m^{(k)}(d) + \beta_m^{(k)}(d))}{\alpha_m^{(k)}(d) + \beta_m^{(k)}(d)} \right) * [1 - (\alpha_m^{(k)}(d) + \beta_m^{(k)}(d))], \quad (2.3)$$

де  $\alpha_m^{(k)}(d)$  - помилка першого роду прийняття рішення на  $k$ -му кроці 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, побудованих в радіальному базисі простору Хеммінга.

Оскільки інформаційний критерій (2.3) є функціоналом від точнісних характеристик, то при репрезентативному обсязі навчальної вибірки

необхідно користуватися їх оцінками:

$$\begin{aligned}\alpha_m^{(k)}(d) &= \frac{K_{1,m}^{(k)}}{n_{min}} D_{1,m}^{(k)}(d) = \frac{K_{1,m}^{(k)}}{n_{min}} \alpha_m^{(k)}(d) = \frac{K_{2,m}^{(k)}}{n_{min}}; \\ \beta_m^{(k)}(d) &= \frac{K_{2,m}^{(k)}}{n_{min}} \beta_m^{(k)}(d) = \frac{K_{3,m}^{(k)}}{n_{min}} D_{2,m}^{(k)}(d) = \frac{K_{4,m}^{(k)}}{n_{min}},\end{aligned}\tag{2.4}$$

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

$K_{2,m}^{(k)}$  - кількість подій, які означають належність “чужих” реалізацій класу  $X_m^o$ ;

$n_{min}$  - мінімальний обсяг репрезентативної навчальної вибірки.

Робоча модифікація критерію Кульбака після відповідної підстановки оцінок (2.4) у вираз (2.3) приймає вигляд

$$E_m^{(k)} = \frac{1}{n} \log_2 \left\{ \frac{2n + 10^{-r} - [K_1^{(k)} + K_2^{(k)}]}{[K_1^{(k)} + K_2^{(k)}] + 10^{-r}} \right\} \left[ n - (K_1^{(k)} + K_2^{(k)}) \right],\tag{2.5}$$

де  $10^{-r}$  - достатньо мале число, що вводиться для уникнення поділу на нуль, яке на практиці вибирається в інтервалі  $1 < r \leq 3$ .

Таким чином, вище наведений інформаційний критерій оптимізації параметрів навчання системи розпізнавання в рамках ІЕІ-технології можна розглядати як узагальнений критерій близькості об’єктів, що розпізнаються.

## РОЗДІЛ 3

### ІНФОРМАЦІЙНЕ, АЛГОРИТМІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ БОРТОВОЇ СИСТЕМИ РОЗПІЗНАВАННЯ

#### 3.1 Вхідний інформаційний опис

Основною задачею формування вхідного математичного опису БСР є створення багатовимірної навчальної матриці  $\| y_{m,i}^{(j)} | m = \overline{1, M}; i = \overline{1, N}, j = \overline{1, n} \|$ .

При цьому необхідно розв'язувати такі задачі:

- формування словника ознак розпізнавання;
- алфавіту класів розпізнавання;
- визначення мінімального обсягу репрезентативної навчальної матриці;
- визначення нормованих допусків на ознаки розпізнавання.

Таким чином, формування вхідного математичного опису ІС потребує детального вивчення та аналізу особливостей функціонування джерела інформації, яким, наприклад, у задачах керування може бути розподілений у просто Вхідний математичний опис здатної навчатися БСР подамо у вигляді структури

$$I_{ex} = \langle G, T, \Omega, Z, Y, X; f_1, f_2 \rangle, \quad (3.1)$$

де  $G$  – простір вхідних сигналів (факторів);

$T$  – множина моментів часу зняття інформації;

$\Omega$  – простір ознак розпізнавання;

$Z$  – простір можливих функціональних станів БСВН;

$Y$  – вхідна навчальна матриця;

$X$  – робоча бінарна навчальна матриця, яка в процесі інформаційно-екстремального машинного навчання адаптується до максимальної повної

ймовірності прийняття правильних діагностичних 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  $X$ .

Як джерело інформації розглядається декартовий добуток .

$$W = G \times T \times \Omega \times Z.$$

На рисунку 3.1 показано ділянки дослідницького поля агрокультур, отримані за результатами аерофотозйомки за оптико-електронним каналом спостереження БПЛА.

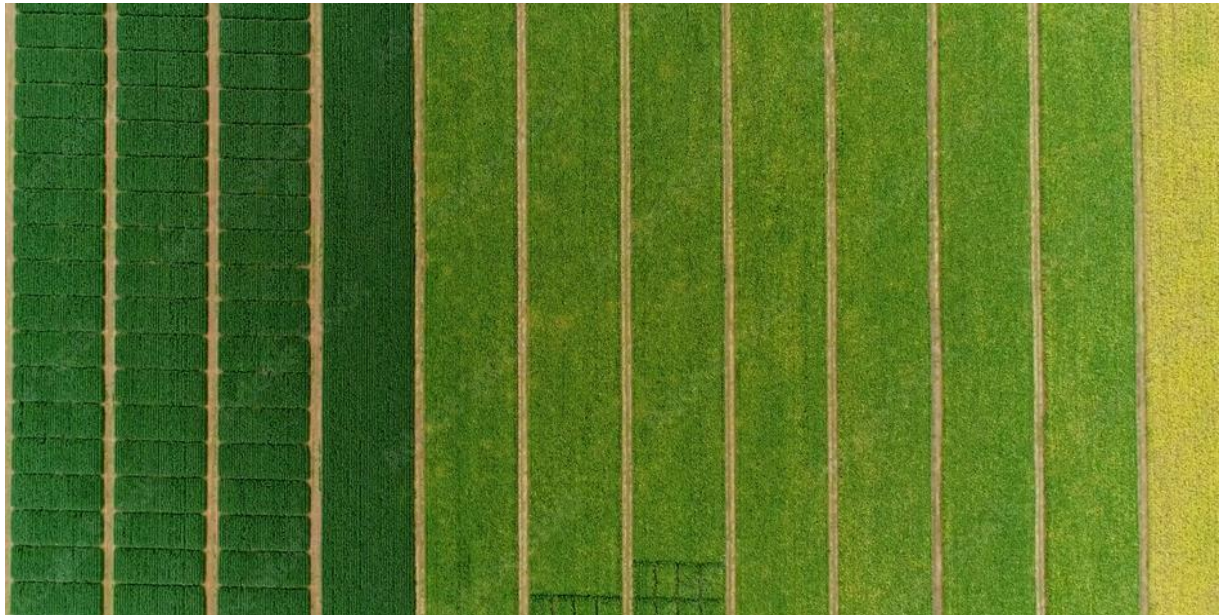


Рисунок 3.1 – Аерофото дослідної ділянки агрокультур

На рисунку 3.1 як класи розпізнавання було обрано ділянки №1 (клас  $X_1^o$  – бобові), №11 (клас  $X_2^o$  – кукурудза) і №12 (клас  $X_3^o$  – соняшник). Для

оперативності обчислення замість зображень ділянок агрокультур далі будемо розглядати їх кадри розміром 60х60 пікселів (рис. 3.2).

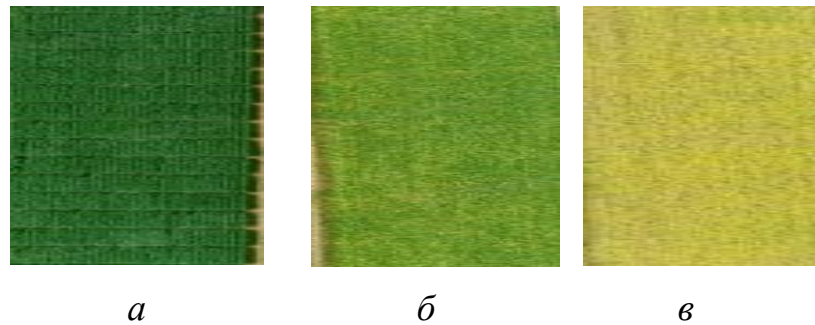


Рисунок 3.2 – Зображення кадрів зон інтересу:  $a$  – клас  $X_1^o$ ;

$б$  – клас  $X_2^o$ ;  $в$  – клас  $X_3^o$

Для заданого алфавіту  $\{X_m^o | m = \overline{1,3}$  класів розпізнавання було сформовано вхідну навчальну матрицю типу “об’єкт-властивість”  $\|y_{m,i}^{(j)}\|, i = \overline{1,N}, j = \overline{1,n}$ , де  $N, n$  – кількість ознак розпізнавання та структурованих векторів ознак класів розпізнавання відповідно, яка має вигляд.

$$\|y_{m,i}^{(j)}\| = \begin{pmatrix} y_{m,1}^{(1)} & y_{m,2}^{(1)} & \dots & y_{m,1}^{(1)} & \dots & y_{m,N}^{(1)} \\ y_{m,1}^{(2)} & y_{m,2}^{(2)} & \dots & y_{m,1}^{(2)} & \dots & y_{m,N}^{(2)} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ y_{m,1}^{(j)} & y_{m,2}^{(j)} & \dots & y_{m,1}^{(j)} & \dots & y_{m,N}^{(j)} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ y_{m,1}^{(n)} & y_{m,2}^{(n)} & \dots & y_{m,1}^{(n)} & \dots & y_{m,N}^{(n)} \end{pmatrix}. \quad (3.2)$$

У матриці (3.2) рядок  $\{y_{m,i}^{(j)} | i = \overline{1,N}\}$  визначає  $j$ -у реалізацію, а стовпчик  $\{y_{m,i}^{(j)} | j = \overline{1,n}\}$  – навчальну випадкову вибірку значень  $i$ -ї ознаки.

Оскільки зображення агрокультур (рис.3.2) відносяться до типу “текстура”, то структуровані вектори ознак розпізнавання формувалися



шляхом порядкового сканування зображень в декартовій системі координат

### **3.2. Машинне навчання бортової системи автономного безпілотного літального апарату з оптимізацією контрольних допусків на ознаки розпізнавання**

У рамках ІЕІ-технології адаптація вхідного математичного опису здатної навчатися системи до її максимальної інформаційної спроможності здійснюється шляхом оптимізації параметрів машинного навчання за інформаційним критерієм. Реалізація базового алгоритму машинного навчання в загальному випадку не гарантує високої достовірності розпізнавання наземних навігаційних орієнтирів, оскільки стартові контрольні допуски на ознаки розпізнавання зазвичай є неоптимальними. Таким чином, виникає необхідність збільшення глибини машинного навчання за допомогою оптимізації системи контрольних допусків, які істотно впливають як на геометричні параметри контейнерів класів розпізнавання, так і на точнісні характеристики класифікаційних рішень. Нехай задано вектор параметрів машинного навчання для алфавіту класів розпізнавання  $\{X_m^o\}$

$$g_m = \langle x_m, d_m, \delta \rangle, \quad (3.3)$$

де  $x_m$  – усереднений структурований вектор ознак класу розпізнавання

$X_m^o$ ;

$d_m$  – радіус гіперсферичного контейнера класу розпізнавання  $X_m^o$ , який відновлюється в радіальному базисі простору ознак розпізнавання;

$\delta$  – параметр поля контрольних допусків на ознаки розпізнавання.

Параметр  $\delta$  дорівнює половині симетричного поля контрольних допусків на ознаки розпізнавання, як це показано на рисунку 3.3.

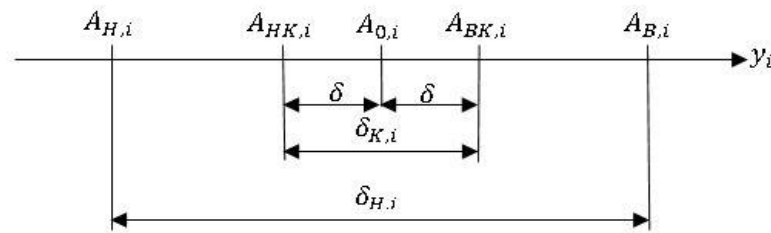


Рисунок 3.3 – Поле допусків на ознаку розпізнавання

На рисунку 3.3 взято такі позначення:  $A_{0,i}$  – номінальне значення ознаки  $y_i$ ;  $A_{Н,i}$  – нижній нормований (експлуатаційний) допуск;  $A_{В,i}$  – верхній нормований допуск;  $A_{НК,i}$  – нижній контрольний допуск;  $A_{БК,i}$  – верхній контрольний допуск;  $\delta_{К,i}$  – поле контрольних допусків;  $\delta_{Н,i}$  – поле нормованих допусків..

Двобічне симетричне поле контрольних допусків через параметр  $\delta$  визначається за формулою

$$\delta_{К,i} = 2\delta \frac{A_{В,i} - A_{Н,i}}{a}, \quad (3.4)$$

де  $a$  – кількість градацій контрольного поля допусків, яка для всіх ознак розпізнавання є однаковою.

Область значень радіуса контейнера класу розпізнавання  $X_m^o$  задається нерівністю

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

де  $d(x_m \oplus x_c)$  – міжцентрова відстань найближчих класів розпізнавання

$X_m^o$  і  $X_c^o$ , яка визначається як кодова відстань між відповідними усередненими векторами ознак  $x_m$  і  $x_c$ .

На практиці при  $a=100$  параметр  $\delta$  може визначатися як кількість відсотків відхилення  $i$ -ї ознаки розпізнавання від її номінального значення  $A_{0,i}$ .

Функціональну категорійну модель інформаційно-екстремальної машинного навчання автономної БСР з оптимізацією контрольних допусків на ознаки розпізнавання показано на рисунку 3.4.

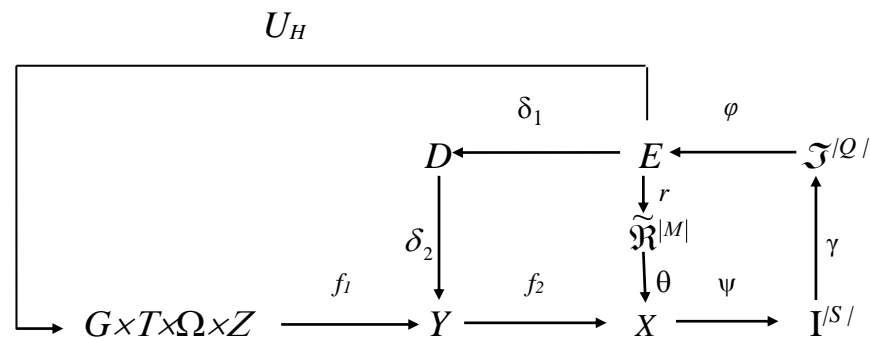


Рисунок 3.4 – Функціональна категорійна модель машинного навчання з оптимізацією контрольних допусків

На рисунку 3.4 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  $\theta$  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  $\gamma_1 : x_m^{(j)} \in X_m^o$ . Оператор  $\gamma$  визначає множину точнісних характеристик  $\mathfrak{Z}^{|Q|}$ , де  $Q = S^2$ , а

оператор  $\varphi$  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. Водночас оператор  $\delta_1$  на кожному кроці машинного навчання змінює контрольне поле допусків на ознаки розпізнавання, а оператор  $\delta_2$  оцінює належність ознак розпізнавання заданому контрольному полю допусків. Оператор  $U_H$  регламентує процес машинного навчання.

Двох циклічний алгоритм машинного навчання системи з оптимізацією контрольних допусків на ознаки розпізнавання згідно з категорійною моделлю (рис. 3.4) має вигляд

$$\{\delta_{K,i}^* \mid i = \overline{1, N}\} = \arg \max_{G_\delta} \{ \max_{G_E \cap \{k\}} \bar{E}^{(k)} \}, \quad (3.5)$$

де

$$\bar{E}^{(k)} = \frac{1}{M} \sum_{m=1}^M E_m^{(k)}$$

усереднене за алфавітом класів розпізнавання значення інформаційного критерію оптимізації, обчислене на  $k$ -му кроці машинного навчання;

$G_\delta$  – область допустимих значень контрольних допусків на ознаки розпізнавання.

Оптимізація контрольних допусків на ознаки розпізнавання може здійснюватися за такими основними схемами:

- алгоритм паралельної оптимізації, за яким контрольні допуски оптимізуються для всіх ознак одночасно;
- алгоритм послідовної оптимізації, за яким контрольні допуски оптимізуються послідовно для кожної ознаки розпізнавання за фіксованих (стартових) значень інших ознак;

– алгоритм оптимізації за зведеним полем допусків, який доцільно застосовувати як послідовно-паралельний алгоритм за наявності різних шкал вимірювання для окремих груп ознак розпізнавання.

Розглянемо інформаційно-екстремальне машинне навчання БСВН з гіперсферичним класифікатором, в якому відновлення контейнерів класів розпізнавання відбувається шляхом паралельної оптимізації контрольних допусків на ознаки розпізнавання. Вхідною інформацією для алгоритму машинного навчання є масив навчальної матриці  $\{y_{m,i}^{(j)}\}$  і система полів нормованих допусків  $\{\delta_{H,i}\}$  на ознаки розпізнавання, яка задає область значень відповідних контрольних допусків.

Розглянемо основні етапи реалізації алгоритму машинного навчання СВА з паралельною оптимізацією контрольних допусків на ознаки розпізнавання, за якою на кожному кроці навчання змінюються контрольні допуски для всіх ознак розпізнавання одночасно:

- 1) ініціалізація лічильника кроків зміни параметра  $\delta$  поля контрольних допусків:  $\delta := 0$ ;
- 2)  $\delta := \delta + 1$ ;
- 3) обчислюються для всіх ознак розпізнавання нижні  $A_{HK,i}$  і верхні  $A_{BK,i}$  контрольні допуски на ознаки розпізнавання:

$$A_{HK,i} = \bar{y}_i - \delta; \quad A_{BK,i} = \bar{y}_i + \delta. \quad (3.6)$$

4) реалізується базовий алгоритм, за яким для кожного значення параметра  $\delta$  визначаються згідно з процедурою (2.22) оптимальні значення радіусів контейнерів класів розпізнавання;

5) якщо  $\delta < \delta_H / 2$ , то виконується пункт 2, інакше – пункт 6;

6) обчислюється усереднене за алфавітом класів розпізнавання максимальне значення інформаційного критерію  $\bar{E}^*$ ;

7) визначається оптимальний параметр поля контрольних допусків, який забезпечує максимальне значення усередненого критерію  $\bar{E}^*$  :

$$\delta^* = \arg \bar{E}^* ;$$

8) обчислюються за формулами (3.6) відповідні оптимальні контрольні допуски на ознаки розпізнавання

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

9) у базі знань автономної БСР запам'ятовуються оптимальні параметри машинного навчання, задані структурованим вектором (3.3);

10) ЗУПИН.

Визначені на етапі паралельної оптимізації контрольні допуски на ознаки розпізнавання є квазіоптимальними, оскільки вони були обчислені за однакову кількість кроків навчання для всіх ознак. Для визначення оптимальних контрольних допусків на ознаки розпізнавання в методах інформаційно-екстремального машинного навчання здійснюється їх послідовна оптимізація. Водночас визначені на етапі паралельної оптимізації контрольні допуски беруться як стартові для алгоритму послідовної оптимізації. Оскільки під час оптимізації  $i$ -ї ознаки інші наступні ознаки мають неоптимальні контрольні допуски, то послідовна оптимізація потребує у цьому разі проведення ітераційних прогонів до того часу, поки значення інформаційного критерію оптимізації не буде змінюватися. Це призводить до зменшення оперативності алгоритму машинного навчання.

Отже одержані за результатами інформаційно-екстремального машинного навчання оптимальні параметри функціонування автономної БСВН дозволяють побудувати в просторі ознак розпізнавання вирішальні правила для прийняття класифікаційних рішень в режимі екзамену.

### 3.3 Функціонування автономної бортової системи в режимі екзамену

За одержаними в процесі машинного навчання оптимальними геометричними параметрами контейнерів класів розпізнавання будуються вирішальні правила, які в продукційній формі подамо у вигляді

$$(\forall X_k^o \in \tilde{\mathfrak{R}}^{|M|})(\forall X_l^o \in \tilde{\mathfrak{R}}^{|M|}) \left( f [(\mu_m > 0) \& (\mu_m = \max_{\{m\}} \{\mu_m\})] \right. \\ \left. \text{then } x^{(j)} \in X_m^o \text{ else } x^{(j)} \notin X_m^o \right), \quad (3.7)$$

де  $x^{(j)}$  – вектор, що розпізнається;

$\mu_m$  – функція належності вектору  $x^{(j)}$  контейнеру класу розпізнавання  $X_m^o$ .

У виразі (3.7) функція належності для гіперсферичного контейнера класу розпізнавання  $X_m^o$  визначається за формулою

$$\mu_m = 1 - \frac{d(x_m^* \oplus x^{(j)})}{d_m^*}, \quad (3.8)$$

де  $x_m^*$  – оптимальний усереднений двійковий вектор ознак;

$d_m^*$  – оптимальний радіус гіперсферичного контейнера.

Таким чином, при функціонуванні БСР в режимі екзамену визначається за вирішальними правилами (3.7) належність реалізації розпізнавального класу до одного з класів із заданого алфавіту. Водночас вирішальні правила через малу обчислювальну трудомісткість відрізняються високою оперативністю.

Оцінювання функціональної ефективності інформаційно-екстремального машинного навчання здійснюється під час функціонування

БСР в режимі екзамену, алгоритм якого аналогічний алгоритму функціонування системи безпосередньо в робочому режимі. У рамках ІЕІ-технології типову категорійну модель у вигляді орієнтованого графа відображень множин, що застосовуються на етапі екзамену, показано на рисунку 3.5.

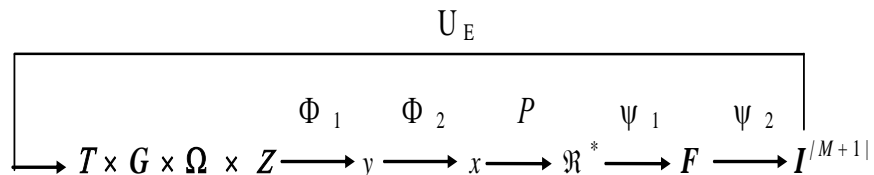


Рисунок 3.5 – Категорійна модель функціонування автономної БСВН в режимі екзамену

У категорійній моделі (рис. 3.5) оператор  $\Phi_1$  формує екзаменаційний вектор ознак  $y$  класу, що розпізнається, аналогічний за структурою вектору з навчальної матриці. Оператор  $\Phi_2$  за заданими на етапі машинного навчання контрольними допусками формує двійковий вектор  $x$ , а оператор  $P$  відображає цей вектор на побудоване на етапі машинного навчання оптимальне розбиття  $\mathfrak{R}^*$  класів розпізнавання. Оператор  $\Psi_1$  для кожного вектора ознак, що розпізнається, обчислює значення побудованих на етапі машинного навчання вирішальних правил і формує терм-множину  $F$ , а оператор  $\Psi_2$  за максимальним значенням вирішального правила відносить вектор  $x$ , до одного з класів заданого алфавіту  $\{X_m^o\}$ . У цьому разі множина гіпотез  $I^{|M+1|}$  містить додаткову гіпотезу  $\gamma_{M+1}$ , що характеризує некласифіковане рішення, тобто вектор, який розпізнається, не належить до жодного класу із заданого алфавіту класів розпізнавання. Призначенням оператора  $U_E$  є регламентація процесу екзамену.

Розглянемо схему реалізації алгоритму екзамену:

- 1) ініціалізація лічильника класів розпізнавання  $m := 0$ ;



- 2)  $m := m + 1$ ;
- 3) ініціалізація лічильника кількості реалізацій  $j := 0$ ;
- 4)  $j := j + 1$ ;
- 5) обчислення функції належності (2.24);
- 6) порівняння: якщо  $j \leq n$ , то виконується пункт 4, інакше – пункт 7;
- 7) обчислення усередненого значення функції належності (2.24)

$$\bar{\mu}_m = \frac{1}{n} \sum_{j=1}^n \mu_{m,j}; \quad (3.9)$$

- 8) порівняння: якщо  $m \leq M$ , то виконується пункт 2, інакше – пункт 9;
- 9) обчислення максимального значення функції (3.9)

$$\bar{\mu}_m^* = \max_{\{m\}} \bar{\mu}_m;$$

10) визначення класу розпізнавання за максимальним значенням функції (3.9);

11) якщо для всіх класів розпізнавання максимальні значення функції (3.9) від'ємні, то экзаменаційна реалізація не класифікується;

12) ЗУПИН.

Таким чином, побудовані в процесі машинного навчання в рамках геометричного підходу вирішальні правила є чіткими і відрізняються незначною обчислювальною трудомісткістю і тому характеризуються високою оперативністю, що є важливим фактором при функціонуванні БПЛА в режимі відеонавігації.

Розглянуті вище категорійні моделі відбивають притаманні людині перетворення інформації та інформаційні потоки, які мають місце за когнітивних процесів формування та прийняття класифікаційних рішень. Тому їх можна розглядати як узагальнені структурні схеми алгоритмів

інформаційно-екстремального машинного навчання і функціонування БСР в режимі екзамену.

### 3.4 Короткий опис програмного забезпечення

Для виконання дипломної роботи була вибрана мова програмування C# (C Sharp). Це об'єктно-орієнтована мова програмування. Розроблено в 1998-2001 роках групою інженерів компанії Microsoft під керівництвом Андерса Хейлсберг і Скотта Вільтаумота. Синтаксис C# близький до C++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML.

Таблиця 3.1 – Основні константи та змінні програми

Ім'я константи або змінної	Короткий опис
<code>verticalLength=40;</code>	Кількість реалізацій
<code>countClasses=3;</code>	Кількість класів розпізнавання
<code>sourceData.Length;</code>	Кількість ознак розпізнавання
<code>bool parallel; bool consistent;</code>	Змінні, за допомогою яких активізуються елементи програми, що виконують збір даних про хід навчання системи
<code>double[] delta;</code>	Масив параметрів СКД, що використовуються при її формуванні і оптимізації.
<code>double[] limitUp; double[] limitDown;</code>	Система контрольних допусків на ознаки розпізнавання
<code>double[] EtalVecBinA; double[] EtalVecBinB; double[] EtalVecBinC;</code>	Еталонні (усереднені) вектори ознак розпізнавання
<code>double radius;</code>	Оптимальні радіуси контейнерів класів розпізнавання

<code>double[,] classA, double[,] binA;</code>	Навчальна матриця та бінарна навчальна матриця відповідно
<code>double[] k1_A; double[] k2_A; double[] k1_B; double[] k2_B; double[] k1_C; double[] k2_C;</code>	Значення точносних характеристик та КФЕ при оптимальних параметрах навчання
<code>double[] dck_xkA; double[] dck_xkB; double[] dck_xkC;</code>	Масив кодових відстаней від еталонного вектора до реалізацій
<code>double[] dck_xnA; double[] dck_xnB; double[] dck_xnC;</code>	Кодова відстань між центрами класів навчання
<code>int i, j, k;</code>	Лічильники
<code>meanValues;</code>	Масив середніх значень матриці

Крім того було розроблено наступні методи (Табл. 3.8):

Таблиця 3.8 – Основні методи програми

<b>Ім'я методу</b>	<b>Короткий опис</b>
<code>double[,] readFile</code>	Метод для завантаження навчальної матриці
<code>double[,] BinMatrix</code>	Розрахунок бінарної навчальної матриці
<code>double findCountXOR</code>	Розрахунок оптимальних значень радіусів контейнерів класів розпізнавання
<code>double[] FindEtalVecBin</code>	Розрахунок еталонних векторів
<code>void alg</code>	Навчання інтелектуальної системи за базовим алгоритмом
<code>void main</code>	Оптимізація СКД за паралельним алгоритмом
<code>void SaveLimit</code>	Збереження отриманих результатів.
<code>double[] findCountXORforEachLinesMatrix</code>	Розрахунок кодових відстаней між центрами класів та реалізаціями.
<code>double[] findLimit</code>	Знаходження допусків

double FindMaxCount	Пошук максимуму
double FindMinCount	Пошук мінімуму

### 3.5 Результати комп'ютерного моделювання

На рисунку 3.6 показано графік залежності інформаційного критерію оптимізації (2.5) від параметра  $\delta$  (delta), отриманий у процесі інформаційно-екстремального машинного навчання БСР з оптимізацією контрольних допусків за процедурою (3.5).

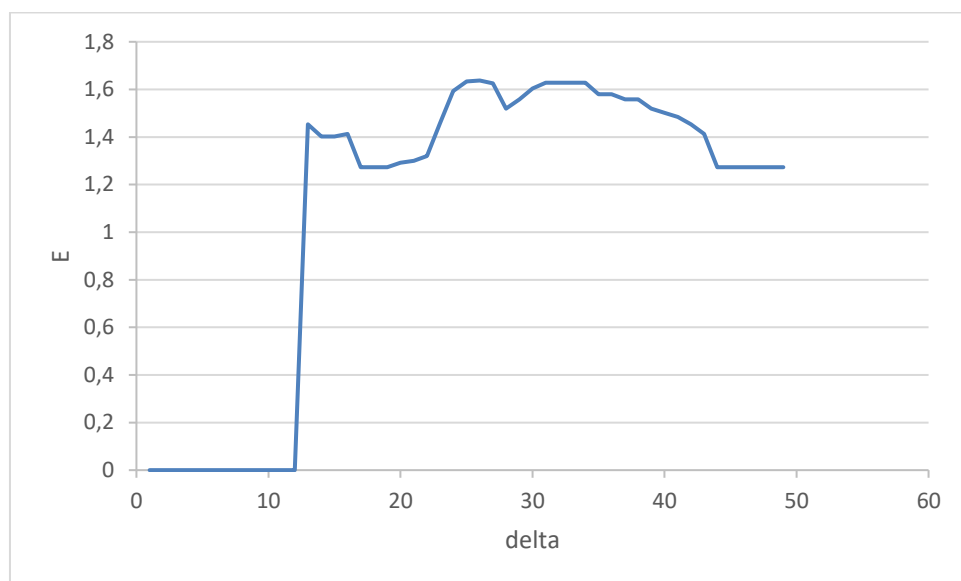


Рисунок 3.6 – Графік залежності інформаційного критерію Кульбака від параметра поля контрольних допусків на ознаки розпізнавання

Аналіз рисунку 3.6 показує, що оптимальне значення параметра поля контрольних допусків дорівнює  $\delta^* = 26$  (у градаціях яскравості зображень) при максимальному значенні інформаційного критерію  $E^* = 1,62$ .

Для побудови вирішальних правил (3.8) необхідно знання оптимальних геометричних параметрів контейнерів класів розпізнавання із заданого алфавіту. На рисунках 3.7 – 3.9 показано графіки залежності інформаційного критерію оптимізації (2.5) від радіусів гіперсферичних

контейнерів класів розпізнавання.

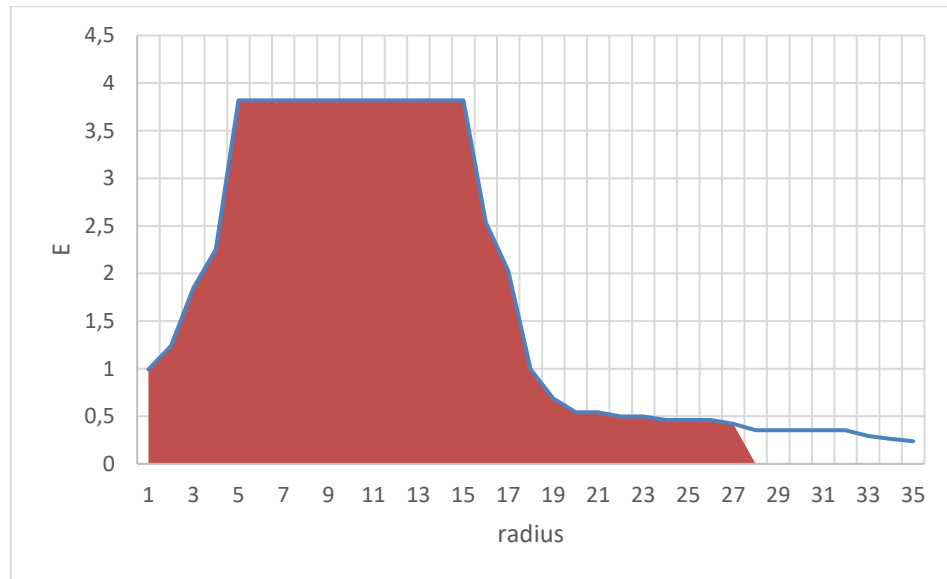


Рисунок 3.7 – Графік залежності критерію Кульбака від радіусу контейнеру першого класу розпізнавання з оптимальним значенням дельта

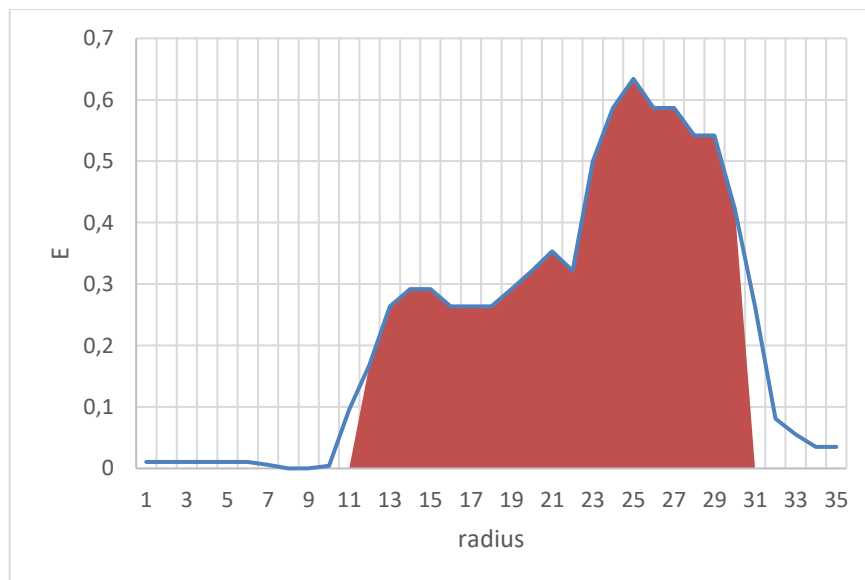


Рисунок 3.8 – Графік залежності критерію Кульбака від радіусу контейнера другого класу розпізнавання з оптимальним значенням дельта

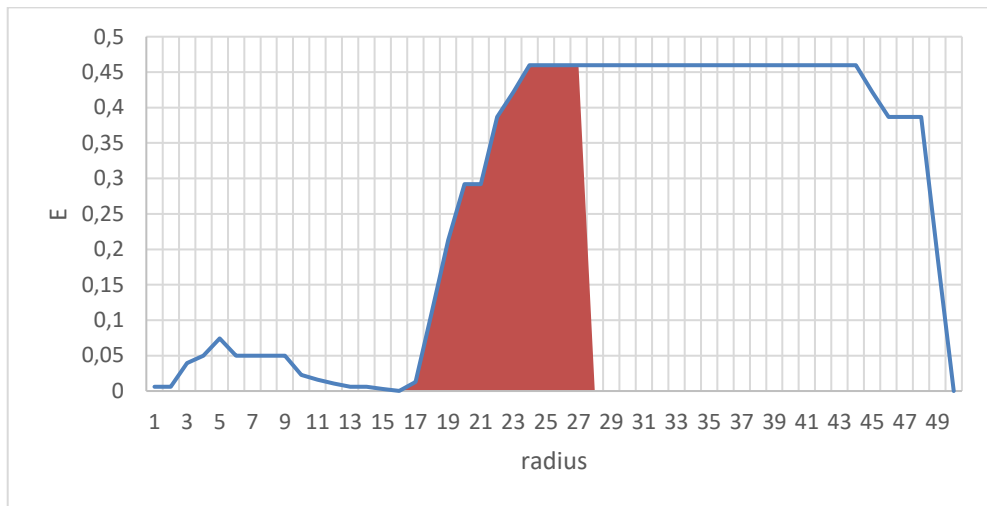


Рисунок 3.9 – Графік залежності критерію Кульбака від радіусу контейнера третього класу розпізнавання з оптимальним значенням дельта

Таким чином за результатами інформаційно-екстремального машинного навчання БСР отримано значення інформаційного критерію оптимізації параметрів машинного навчання, які підтверджують розділимість класів розпізнавання. Основними результатами машинного навчання є:

1) Оптимальні радіуси контейнерів класів розпізнавання (у кодових одиницях):

1 клас: 15

2 клас: 25

3 клас: 27

2) Максимальні значення інформаційного критерію:

1 клас: 3.818

2 клас: 0.634

3 клас: 0.460

3) Оптимальний параметр поля контрольних допусків:

Delta: 26

З метою перевірки функціональної ефективності було реалізовано алгоритм екзамену.

На рисунку 3.10 показано вікно результатів екзамену.

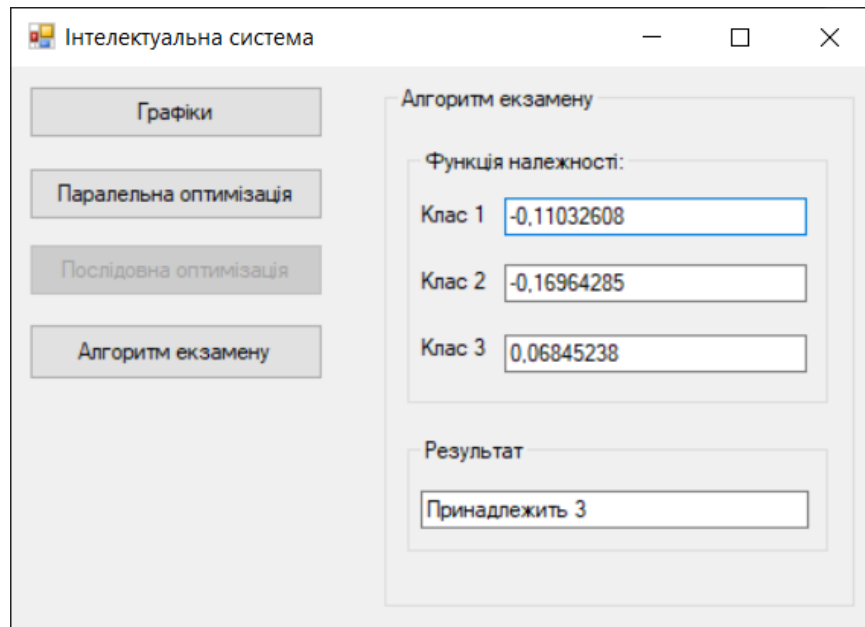


Рисунок 3.10 – Інтерфейс програми екзамену

Аналіз значень, представлених на рисунку 3.10 показує, що подана для екзамену реалізація належить класу розпізнавання  $X_3^o$ . Оскільки розпізнавався структурований вектор ознак першого класу, то результати екзамену підтвердили достовірність розпізнавання та надійність запропонованого алгоритму.

## ВИСНОВКИ

1. Проаналізовано сучасний стан застосування безпілотних літальних апаратів для спостереження за розвитком агрокультур. Встановлено, що актуальною задачею є сегментація регіону вирощування агрокультур з метою визначення виду рослин. При цьому особливу увагу приділено вивченню основних понять, принципів, математичних моделей інформаційно-екстремального машинного навчання БСР, функціонування таких систем в різних режимах та оцінці їх ефективності.

2. Розроблено алгоритми інформаційно-екстремального машинного навчання системи розпізнавання з паралельною оптимізацією контрольних допусків та алгоритм функціонування системи в режимі екзамену. Як інформаційний критерій оптимізації параметрів машинного навчання БСР використовувався модифікований критерій Кульбака.

3. Результати комп'ютерного моделювання, проведеного в режимі екзамену, підтвердили працездатність розробленого програмного забезпечення. При цьому отримано достатньо високу достовірність розпізнавання.

4. Для підвищення функціональної ефективності інформаційно-екстремального машинного навчання БСР для сегментації агрокультур доцільно збільшення глибини машинного навчання шляхом оптимізації додаткових параметрів функціонування системи, включаючи параметри формування вхідного інформаційного опису



## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Єпіфанов І. Н. Проблематика використання безпілотних літальних апаратів (дронів) в логістиці [Електронний ресурс]. Режим доступу: <https://scientificarticle.ru/images/PDF/2016/9/.pdf>.
2. Використання дронів в логістиці [Електронний ресурс]. Режим доступу: <https://www.microdrones.com/en/industry-experts/unmanned-cargo/>.
3. Томлін, К. Дана (1990). Геоінформаційні системи та картографічне моделювання. Серія «Прентіс Холл» із географічної інформатики. Прентіс Холл. Процитовано 5 січня 2017.
4. Субботін С. Нейро-нечіткий мережевий синтез та спрощення на прецедентах у задачах діагностики та розпізнавання образів / С. Субботін // Оптична пам'ять та нейронні мережі (Інформаційна оптика). – 2013. – Вип. 22, № 2. – С. 97 – 103.
5. Довбиш А. С. Основи проектування інтелектуальних систем: Навчальний посібник / А. С. Довбиш. – Суми: Видавництво СумДУ, 2009. – 171 с.
6. Довбиш А.С. Інформаційно-екстремальний алгоритм машинного навчання системи розпізнавання транспортних засобів / А. С. Довбиш, Ю. В. Симоновський, О. В. Коробченко, М. А. Летюга // Вісник НТУ «ХП», 2016. – № 45 (1217). – С. 17–22
7. Довбиш А. С., Будник М. М., П'ятаченко В. Ю., Мироненко М. І. Інформаційно-екстремальне машинне навчання бортової системи розпізнавання транспортних засобів. Кібернетика та системний аналіз, 2020, 56(4), стор. 534-543. DOI:10.1007/s10559-020-00269-y.
8. Саймон Д. Алгоритмы эволюционной оптимизации. — М.: ДМК Пресс, 2020. — 940 с.
9. Ян Леун. Как учится машина. Революция в области нейронных сетей и глубокого обучения, 2021.
10. Банцаф, Вольфганг; Нордін, Пітер; Келлер, Роберт; Франконе, Френк (1998). Генетичне програмування – Вступ

11. Краснопоясовський А.С. Інформаційний синтез інтелектуальних систем керування: підхід, що ґрунтується на методі функціональностатистичних випробувань.– Суми, 2004.– 261

12. Проценко О., Савченко Т., Мироненко М., Приходченко О. Інформаційне та екстремальне машинне навчання для бортової системи розпізнавання наземних об'єктів. Матеріали - 2020 IEEE 11-та Міжнародна конференція з надійних систем, послуг і технологій, DESSERT 2020. Сторінки: 213-218

13. Мукунд Сундарараджан, Анкур Талі, Цікі Ян. Аксіоматична атрибуція для глибоких мереж [2017]. <https://arxiv.org/abs/1703.01365>

14. Карен Симонян, Андреа Ведальді, Ендрю Зіссерман. Візуалізація моделей класифікації зображень і карт помітності [2013]. <https://arxiv.org/abs/1312.6034>

15. Шон Сю, Субхашіні Венугопалам, Мукунд Сундарараджан. Атрибуція в масштабі та просторі. <https://archive.org/pdf/2004.03383.pdf>

## ДОДАТОК

### Form1.cs

```

public partial class Form1 : Form {
    Form2 form2 = new Form2();
    public double ro = 0.5d;
    public int verticalLength = 40;
    public double[] sourceData;
    double[] isTrue;
    public static double Length;
    public double[,] classA;
    public double[,] classB;
    public double[,] classC;
    public static int countClasses = 3;

    public static bool parallel = false;
    public static bool consistent = false;

    CreateClassA createClassA = new CreateClassA();
    CreateClassB createClassB = new CreateClassB();
    CreateClassC createClassC = new CreateClassC();
    Functions functions = new Functions();
    UsingFiles usingFiles = new UsingFiles();

    public Form1() {
        InitializeComponent();
        Length = sourceData.Length;
    }
    public void callCreateWriteClasses() {
        classA = createClassA.create();
        classB = createClassB.create(classA);
        classC = createClassC.create(classA);
        usingFiles.writeFile(classA, "classA");
        usingFiles.writeFile(classB, "classB");
        usingFiles.writeFile(classC, "classC");

        mainMethod();
    }

    public void mainMethod() {}

    private void Form1_Load(object sender, EventArgs e) {
        Text = "Інтелектуальна система";
        FormBorderStyle = FormBorderStyle.FixedSingle;
        Write.txt1 = textBox1;
        Write.txt2 = textBox2;
        Write.txt3 = textBox3;
        Write.txt4 = textBox4;
        Write.pbr1 = progressBar1;
        Write.lb14 = label4;
        Height = 335;
        groupBox4.Visible = false;
        button1.Visible = false;
        button2.Enabled = false;

        if (File.Exists(usingFiles.url+"classA"+usingFiles.format) &&
            File.Exists(usingFiles.url+"classB"+usingFiles.format) &&
            File.Exists(usingFiles.url+"classC"+usingFiles.format)) {
            classA = usingFiles.readFile("classA");
            classB = usingFiles.readFile("classB");
            classC = usingFiles.readFile("classC");
        }
    }
}

```

```

        mainMethod();
    }

    else {
        callCreateWriteClasses();
    }
}

private void button1_Click(object sender, EventArgs e) {
    callCreateWriteClasses();
}

async void consistentAsync() {
    consistent = true;
    parallel = false;
    button2.Enabled = false;
    DrawGraph drawGraph = new DrawGraph();
    drawGraph.Text = "Послідовна оптимізація КД";
    groupBox4.Visible = true;
    groupBox4.Text = "Процес послідовної оптимізації КД";
    Height = 440;
    OptimizationKD optimizationKD = new OptimizationKD(classA, classB, classC);
    await Task.Run(() => {
        optimizationKD.main("consistent");
    });
    Height = 335;
    groupBox4.Visible = false;
    drawGraph.chart2.Series.Clear();
    drawGraph.chart3.Series.Clear();
    drawGraph.chart4.Series.Clear();

    drawGraph.GetGraph(drawGraph.chart1, optimizationKD.E_consistent_all,
"Послідовна оптимізація КД", 4);
    drawGraph.Show();
}

private void button2_Click(object sender, EventArgs e) {
    consistentAsync();
}

private void button3_Click(object sender, EventArgs e) {
    parallel = true;
    consistent = false;
    button2.Enabled = true;

    DrawGraph drawGraph = new DrawGraph();
    drawGraph.Text = "Паралельна оптимізація КД";

    OptimizationKD optimizationKD = new OptimizationKD(classA, classB, classC);
    optimizationKD.main("parallel");

    drawGraph.chart2.Series.Clear();
    drawGraph.chart3.Series.Clear();
    drawGraph.chart4.Series.Clear();
    drawGraph.GetGraph(drawGraph.chart1, optimizationKD.E, "Паралельна
оптимізація КД", 4, optimizationKD.k1, optimizationKD.k2);
    drawGraph.Show();
}

private void button4_Click(object sender, EventArgs e) {
    button2.Enabled = false;
    if (parallel == false && consistent == false) {
        DrawGraph drawGraph = new DrawGraph();
        drawGraph.Text = "Графіки залежності критерію Кульбака від радіусів

```

контейнерів класів розпізнавання";

```

        AlgMachine algMachine = new AlgMachine(classA,classB,classC);
        algMachine.alg(20d);
        algMachine.algFind_K_KFE();

        drawGraph.GetGraph(drawGraph.chart1, algMachine.E_A, "Кульбак", 1,
algMachine.k1_A, algMachine.k2_A);
        drawGraph.GetGraph(drawGraph.chart2, algMachine.E_B, "Кульбак", 2,
algMachine.k1_B, algMachine.k2_B);
        drawGraph.GetGraph(drawGraph.chart3, algMachine.E_C, "Кульбак", 3,
algMachine.k1_C, algMachine.k2_C);
        drawGraph.Show();

    } else if (parallel == true) {
        DrawGraph drawGraph = new DrawGraph();
        drawGraph.Text = "Графіки одержані при оптимальному значенні дельта";

        AlgMachine algMachine = new AlgMachine(classA,classB,classC);
        algMachine.alg(DrawGraph.op_delta);
        algMachine.algFind_K_KFE();

        drawGraph.GetGraph(drawGraph.chart1, algMachine.E_A, "Кульбак", 1,
algMachine.k1_A, algMachine.k2_A);
        drawGraph.GetGraph(drawGraph.chart2, algMachine.E_B, "Кульбак", 2,
algMachine.k1_B, algMachine.k2_B);
        drawGraph.GetGraph(drawGraph.chart3, algMachine.E_C, "Кульбак", 3,
algMachine.k1_C, algMachine.k2_C);
        drawGraph.Show();

        parallel = false;

    } else if (consistent == true) {
        DrawGraph drawGraph = new DrawGraph();
        drawGraph.Text = "Графіки після послідовної оптимізації";

        AlgMachine algMachine = new AlgMachine(classA,classB,classC);
        algMachine.alg("consistent");
        algMachine.algFind_K_KFE();

        drawGraph.GetGraph(drawGraph.chart1, algMachine.E_A, "Кульбак", 1,
algMachine.k1_A, algMachine.k2_A);
        drawGraph.GetGraph(drawGraph.chart2, algMachine.E_B, "Кульбак", 2,
algMachine.k1_B, algMachine.k2_B);
        drawGraph.GetGraph(drawGraph.chart3, algMachine.E_C, "Кульбак", 3,
algMachine.k1_C, algMachine.k2_C);
        drawGraph.Show();

        consistent = false;

    }

}

private void button5_Click(object sender, EventArgs e) {
    button2.Enabled = false;
    AlgExam algExam = new AlgExam(classA, classB, classC, classC);
}
}

```

## AlgMachine.cs

```

class AlgMachine {
    public double[,] classA;
    public double[,] classB;
    public double[,] classC;
    public double[] meanValues;
    public static double[] limitUp;
    public static double[] limitDown;

    static double[] limitUpStart;
    static double[] limitDownStart;

    public static double[] limitUpSave;
    public static double[] limitDownSave;
    public static int prog = 0;
    double[,] binA;
    double[,] binB;
    double[,] binC;
    double[] meanBinA;
    protected double[] EtalVecBinA;
    double[] meanBinB;
    protected double[] EtalVecBinB;
    double[] meanBinC;
    protected double[] EtalVecBinC;
    double[] dck_xkA; // к своим
    double[] dck_xnA; // к чужим
    double[] dck_xkB; // к своим
    double[] dck_xnB; // к чужим
    double[] dck_xkC; // к своим
    double[] dck_xnC; // к чужим
    public double[] k1_A;
    public double[] k2_A;
    public double[] k1_B;
    public double[] k2_B;
    public double[] k1_C;
    public double[] k2_C;
    public double[] E_A;
    public double[] E_B;
    public double[] E_C;
    public double k1_sum;
    public double k2_sum;
    public double E_sum;
    Functions functions = new Functions();
    public AlgMachine(double[,] classA, double[,] classB, double[,] classC) {
        this.classA = classA;
        this.classB = classB;
        this.classC = classC;
    }

    public void alg(double delta) {
        meanValues = functions.findMean(classA);
        limitUp = functions.findLimit(meanValues, "Up", delta);
        limitDown = functions.findLimit(meanValues, "Down", delta);
        algMain();
    }

    public void start() {
        meanValues = functions.findMean(classA);
        limitUpStart = functions.findLimit(meanValues, "Up", DrawGraph.op_delta);
        limitDownStart = functions.findLimit(meanValues, "Down", DrawGraph.op_delta);
        limitUp = new double[meanValues.Length];
        limitDown = new double[meanValues.Length];
    }
}

```

```

limitUpSave = new double[meanValues.Length];
limitDownSave = new double[meanValues.Length];

for (int i = 0; i < meanValues.Length; i++) {
    limitUp[i] = limitUpStart[i];
    limitDown[i] = limitDownStart[i];

    limitUpSave[i] = limitUpStart[i];
    limitDownSave[i] = limitDownStart[i];
}

}

public void SaveLimit() {
    for (int i = 0; i < Form1.Length; i++) {
        limitUpSave[i] = limitUp[i];
        limitDownSave[i] = limitDown[i];
    }
}

public void alg(int oznaka, int i, double optDelta, int progon) {
    if (prog != progon) {
        SaveLimit();
    }

    if (oznaka != 0) {
        limitUp[oznaka - 1] = limitUpSave[oznaka - 1] + optDelta;
        limitDown[oznaka - 1] = limitDownSave[oznaka - 1] - optDelta;
    }

    limitUp[oznaka] = limitUpSave[oznaka] + i;
    limitDown[oznaka] = limitDownSave[oznaka] - i;

    algMain();

    prog = progon;
}

public void alg(string TYPE) {
    if (TYPE == "consistent") {
        meanValues = functions.findMean(classA);
        limitUp = new double[meanValues.Length];
        limitDown = new double[meanValues.Length];

        for (int i = 0; i < Form1.Length; i++) {
            limitUp[i] = limitUpSave[i];
            limitDown[i] = limitDownSave[i];
        }
    }
    algMain();
}

private void algMain() {
    binA = functions.BinMatrix(classA, limitDown, limitUp);
    binB = functions.BinMatrix(classB, limitDown, limitUp);
    binC = functions.BinMatrix(classC, limitDown, limitUp);

    meanBinA = functions.findMean(binA);
    EtalVecBinA = functions.FindEtalVecBin(meanBinA);
    meanBinB = functions.findMean(binB);
    EtalVecBinB = functions.FindEtalVecBin(meanBinB);
    meanBinC = functions.findMean(binC);
}

```

```

EtalVecBinC = functions.FindEtalVecBin(meanBinC);

double[] distancesA = new double[2];
double[] distancesB = new double[2];
double[] distancesC = new double[2];
double[] minDistances = new double[3];
int[] index = new int[3];

distancesA[0] = functions.findCountXOR(EtalVecBinA, EtalVecBinB); //0 9 0
distancesA[1] = functions.findCountXOR(EtalVecBinA, EtalVecBinC);

distancesB[0] = functions.findCountXOR(EtalVecBinB, EtalVecBinA); //9 0 9
distancesB[1] = functions.findCountXOR(EtalVecBinB, EtalVecBinC);

distancesC[0] = functions.findCountXOR(EtalVecBinC, EtalVecBinA); //0 9 0
distancesC[1] = functions.findCountXOR(EtalVecBinC, EtalVecBinB);

minDistances[0] = functions.FindMinCount(distancesA);
minDistances[1] = functions.FindMinCount(distancesB);
minDistances[2] = functions.FindMinCount(distancesC);
index[0] = Array.IndexOf(distancesA, minDistances[0]);
index[1] = Array.IndexOf(distancesB, minDistances[1]);
index[2] = Array.IndexOf(distancesC, minDistances[2]);

dck_xkA = functions.findCountXORforEachLinesMatrix(EtalVecBinA, binA);
switch (index[0]) {
    case 0:
        dck_xnA = functions.findCountXORforEachLinesMatrix(EtalVecBinA,
binB);
        break;
    case 1:
        dck_xnA = functions.findCountXORforEachLinesMatrix(EtalVecBinA,
binC);
        break;
}
dck_xkB = functions.findCountXORforEachLinesMatrix(EtalVecBinB, binB);
switch (index[1]) {
    case 0:
        dck_xnB = functions.findCountXORforEachLinesMatrix(EtalVecBinB,
binA);
        break;
    case 1:
        dck_xnB = functions.findCountXORforEachLinesMatrix(EtalVecBinB,
binC);
        break;
}
dck_xkC = functions.findCountXORforEachLinesMatrix(EtalVecBinC, binC);
switch (index[2]) {
    case 0:
        dck_xnC = functions.findCountXORforEachLinesMatrix(EtalVecBinC,
binA);
        break;
    case 1:
        dck_xnC = functions.findCountXORforEachLinesMatrix(EtalVecBinC,
binB);
        break;
}
}
public void algFind_K_KFE() {
    k1_A = functions.findK(dck_xkA, meanValues.Length);
    k2_A = functions.findK(dck_xnA, meanValues.Length);
    k1_B = functions.findK(dck_xkB, meanValues.Length);
    k2_B = functions.findK(dck_xnB, meanValues.Length);
    k1_C = functions.findK(dck_xkC, meanValues.Length);
    k2_C = functions.findK(dck_xnC, meanValues.Length);
}

```



```

        E_A = functions.find_KFE(k1_A, k2_A, meanValues.Length);
        E_B = functions.find_KFE(k1_B, k2_B, meanValues.Length);
        E_C = functions.find_KFE(k1_C, k2_C, meanValues.Length);
    }
}
OptimizationKD.cs

class OptimizationKD : AlgMachine {
    private double prevE;

    public double PrevE {
        get {
            return prevE;
        }

        set {
            prevE = value;
        }
    }

}

    public double[] E;
    public double[] k1;
    public double[] k2;
    public double[] KFE_max = new double[3];
    int[] radius = new int[3];
    double k1_A_mid;
    double k2_A_mid;
    double k1_B_mid;
    double k2_B_mid;
    double k1_C_mid;
    double k2_C_mid;

    public double[] E_consistent_1;
    public double[] E_consistent_2;
    public double[] E_consistent_3;
    public double[] E_consistent_all;
    public double[] E_optMaxIndex = new double[3];

    double E_max = 0;
    int E_max_index;
    Functions functions = new Functions();
    Form1 form1 = new Form1();

    public OptimizationKD(double[,] classA, double[,] classB, double[,] classC) :
base(classA, classB, classC) {
    }

    public void main(string TYPE) {
        E = new double[classA.GetLength(0)];
        k1 = new double[classA.GetLength(0)];
        k2 = new double[classA.GetLength(0)];

        if (TYPE == "parallel") {
            for (int i = 0; i <= 100; i++) {
                base.alg(i);
                repeat(i);
            }
        } else if (TYPE == "consistent") {
            E_consistent_1 = new double [form1.sourseData.Length];
            E_consistent_2 = new double [form1.sourseData.Length];
            E_consistent_3 = new double [form1.sourseData.Length];
            E_consistent_all = new double [form1.sourseData.Length*3];
            double progI = 0;

```

```

double opt = 0;
int index = 0;
base.start();
for (int progon = 0; progon < 3; progon++) {
    for (int oznaka = 0; oznaka < form1.sourceData.Length; oznaka++) {
        double progress = (100 * progI) / (form1.sourceData.Length * 3);
        Write.pbr1.Invoke(new Action(() =>{
            Write.pbr1.Value = Convert.ToInt32(progress);
            Write.lbl14.Text = Convert.ToString(Math.Round(progress, 5) +
"%");

        }));
        progI++;
        index = E_max_index;

        opt = E_max;
        E_max = 0;

        for(int delta = 0; delta < 50; delta++) {
            base.alg(oznaka, delta, index, progon);
            repeat(delta);
        }

        switch (progon) {
            case 0:
                E_consistent_1[oznaka] = E_max;
                break;
            case 1:
                E_consistent_2[oznaka] = E_max;
                break;
            case 2:
                E_consistent_3[oznaka] = E_max;
                break;
        }
    }

    limitUp[form1.sourceData.Length - 1] =
limitUpSave[form1.sourceData.Length - 1] + E_max_index;
    limitDown[form1.sourceData.Length - 1] =
limitDownSave[form1.sourceData.Length - 1] - E_max_index;
}

base.SaveLimit();

for (int i = 0; i < form1.sourceData.Length * 3; i++) {
    if (i < form1.sourceData.Length) {
        E_consistent_all[i] = E_consistent_1[i];
    } else if (i < form1.sourceData.Length * 2) {
        E_consistent_all[i] = E_consistent_2[i -
form1.sourceData.Length];
    } else if (i < form1.sourceData.Length * 3) {
        E_consistent_all[i] = E_consistent_3[i - (form1.sourceData.Length
* 2)];
    }
}

private void repeat(int i) {
    base.algFind_K_KFE();

    KFE_max[0] = functions.FindMaxCount(E_A);
    KFE_max[1] = functions.FindMaxCount(E_B);
    KFE_max[2] = functions.FindMaxCount(E_C);
}

```

```

radius[0] = functions.findWorkArea(k1_A, k2_A);
radius[1] = functions.findWorkArea(k1_B, k2_B);
radius[2] = functions.findWorkArea(k1_C, k2_C);

k1_A_mid = functions.isFindWorkArea(E_A, KFE_max[0], k1_A, radius[0]);
k2_A_mid = functions.isFindWorkArea(E_A, KFE_max[0], k2_A, radius[0]);
k1_B_mid = functions.isFindWorkArea(E_B, KFE_max[1], k1_B, radius[1]);
k2_B_mid = functions.isFindWorkArea(E_B, KFE_max[1], k2_B, radius[1]);
k1_C_mid = functions.isFindWorkArea(E_C, KFE_max[2], k1_C, radius[2]);
k2_C_mid = functions.isFindWorkArea(E_C, KFE_max[2], k2_C, radius[2]);
KFE_max[0] = radius[0] == 0 ? 0 : KFE_max[0];

KFE_max[1] = radius[1] == 0 ? 0 : KFE_max[1];

KFE_max[2] = radius[2] == 0 ? 0 : KFE_max[2];

k1_sum = k1_A_mid + k1_B_mid + k1_C_mid;
k2_sum = k2_A_mid + k2_B_mid + k2_C_mid;
E_sum = KFE_max[0] + KFE_max[1] + KFE_max[2];

k1[i] = k1_sum / 3d;
k2[i] = k2_sum / 3d;
E[i] = E_sum / 3d;

if (radius[0] == 0 || radius[1] == 0 || radius[2] == 0) {
    if (prevE < E[i]) {
        E[i] = 0;
    }
}
prevE = i != 0 ? E[i - 1] : 0;

if (k1[i] >= 0.5 && k2[i] < 0.5) {
    if (E[i] > E_max) {
        E_max = E[i];
        E_max_index = i;
    }
}
}
}

```