

УДК 629.7.05:004.93
УКПП
№ держреєстрації 0117U003934
Інв. №

Міністерство освіти та науки України
Сумський державний університет
(СумДУ)

40007, м. Суми, вул. Римського-Корсакова, 2
тел. (0542) 33-54-79 факс (0542) 33-54-79

ЗАТВЕРДЖУЮ

Проректор з наукової роботи
д-р. фіз.-мат. наук, професор

_____ А.М. Черноус

ЗВІТ

ПРО НАУКОВО-ДОСЛІДНУ РОБОТУ

Інтелектуальна автономна бортова система безпілотного літального апарату
для ідентифікації об'єктів на місцевості

РОЗРОБКА ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ БОРТОВОЇ СИСТЕМИ
БЕЗПІЛОТНОГО АПАРАТУ, ЩО ФУНКЦІОНУЄ В РЕЖИМІ НАВЧАННЯ ВИРІШУВАЛЬНИХ
ПРАВИЛ ТА КЛАСИФІКАЦІЙНОГО АНАЛІЗУ СПОСТЕРЕЖЕНЬ
(проміжний)

Керівник НДР
доцент, канд. техн. наук

В.В. Москаленко

2019

Рукопис завершено 10 грудня 2019 р.

Результати цієї роботи розглянуто на засіданні наукової ради СумДУ, протокол від
26.11.2019 № 4

СПИСОК АВТОРІВ

Керівник НДР,
канд. техн. наук,
провідн. наук. співроб.

(10.12.2019)

Москаленко В. В. (вступ, висновки,
підрозділи 1.1-1.3)

Відповідальний виконавець
канд. техн. наук,
старш. наук. співроб.

(10.12.2019)

Москаленко А. С. (підрозділ 1.3, 2.2)

Виконавці:
канд. техн. наук,
старш. наук. співроб.

(10.12.2019)

Берест О. Б. (підрозділ 2.3)

канд. техн. наук,
старш. наук. співроб.

(10.12.2019)

Мартиненко С. С. (підрозділ 1.3)

канд. фіз.-мат. наук,
старш. наук. співроб.

(10.12.2019)

Журба В. О. (підрозділ 2.2)

канд. техн. наук,
старш. наук. співроб.

(10.12.2019)

Нагорний В. В. (підрозділ 2.3)

мол. наук. співроб.

(10.12.2019)

Окопний Р. П. (підрозділ 2.3)

мол. наук. співроб.

(10.12.2019)

Коробов А. Г. (підрозділ 2.1)

аспірант,

мол. наук. співроб.

(10.12.2019)

Зарецький М. О. (підрозділ 2.3)

РЕФЕРАТ

Звіт про НДР: 67 с., 1 табл., 31 рис., 20 джерел.

БОРТОВА СИСТЕМА БЕЗПІЛОТНОГО ЛІТАЛЬНОГО АПАРАТУ, ЕКСТРАКТОР ОЗНАК, ВИРІШУВАЛЬНІ ПРАВИЛА, ІНФОРМАЦІЙНО-ЕКСТРЕМАЛЬНА ІНТЕЛЕКТУАЛЬНА ТЕХНОЛОГІЯ, МАШИННЕ НАВЧАННЯ, ОПТИМІЗАЦІЯ.

Об'єкт дослідження – процес машинного навчання бортової системи безпілотного апарату та автономної ідентифікації об'єктів на місцевості.

Мета роботи – підвищення функціональної ефективності бортової системи безпілотного літального апарату, що здійснює у автономному режимі локальну навігацію та класифікаційний аналіз спостережень за умов ресурсних та інформаційних обмежень.

Методи дослідження – аналіз літературних джерел, методи розрідженого кодування, методи нейромережевого та інформаційно-екстремального моделювання.

Розроблено діаграми процесів навчання вирішувальних правил та класифікаційного аналізу даних, алгоритми та програми для локалізації і трекінгу об'єктів в полі зору безпілотного пристрою та оптимізації параметрів моделі аналізу даних. Здійснено імітаційне моделювання на відкритих наборах даних. Виконано алгоритмічну і програмну реалізацію детектора рухомих об'єктів на спостереженнях з рухомої камери. При цьому запропонована інкрементальна схема навчання вирішувальних правил. Реалізовано схему активного навчання.

Результати виконання роботи впроваджено в навчальний процес під час підготовки лекційних курсів з навчальних дисциплін «Інтелектуальні системи керування» і «Introduction to Data Science» та під час написання п'яти магістерських робіт. Також отриманий досвід використано для надання послуг, пов'язаних з впровадженням проекту “Minect.ai” в рамках госпдоговірної теми № 09-13 від 07.08.19 на суму 255,00 тис.грн. / 110 000,00 тис. грн.

ЗМІСТ

СКРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	6
ВСТУП	7
1 МОДЕЛІ І МЕТОДИ НАВЧАННЯ БОРТОВОЇ СИСТЕМИ БЕЗПІЛОТНОГО АПАРАТУ	9
1.1 Моделі і методи навчання екстрактора ознак	9
1.2 Моделі і методи побудови вирішувальних правил	17
2 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІЗУ ДАНИХ БОРТОВОЇ СИСТЕМИ БЕЗПІЛОТНОГО АПАРАТУ	40
2.1 Апаратно-програмне забезпечення бортової системи безпілотного апарату.....	40
2.2 Діаграми процесу навчання та екзамену бортової системи безпілотного апарату	49
2.3 Алгоритми функціонування бортової системи в режимі класифікаційного аналізу рухомих об'єктів	56
2.4 Результати фізичного моделювання	59
ВИСНОВКИ.....	63
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	65

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БЛА – безпілотний літальний апарат;

ІЕІ – технологія – інформаційно-екстремальна інтелектуальна технологія;

КФЕ – критерій функціональної ефективності;

СКД – система контрольних допусків.

ВСТУП

Малогабаритні безпілотні апарати часто використовуються у задачах моніторингу території та об'єктів. При цьому безпілотний апарат часто розглядається як мобільний сенсор збору даних для їх подальшої обробки в хмарному сервісі. Однак існує тенденція до розвитку крайових обчислень з метою розвантаження комунікаційних каналів інфокомунікаційного середовища і захисту інформації. Тому багато сучасних розробок пов'язані з розширенням функціональних можливостей та підвищенням рівня автономності безпілотних апаратів. До затребуваних функціональних можливостей безпілотних апаратів відносяться пошук і класифікація об'єктів інтересу та автономна навігація.

Алгоритми детектування об'єктів інтересу і їх класифікаційний аналіз можуть бути корисними в таких практичних задачах, як відео-аналітика систем рятувально-пошукових заходів, охорони периметру, прицілювання і наведення зброї. Автономна навігація може значно полегшити задачу керування безпілотним апаратом під час польоту поруч із будівлями, деревами або в приміщенні.

Використання візуальних сенсорів в задачах автономної навігації та розпізнавання об'єктів на місцевості є перспективним підходом з точки зору інформативності, ваги та ціни. За допомогою відео камери можна одночасно здійснювати оцінку переміщення та отримувати інформацію про зовнішнє середовище. При цьому для прискорення режиму екзамену в системах інтелектуального аналізу візуальних образів розроблено різноманітні апаратні прискорювачі на основі графічних обчислювальних пристроїв та нейропроцесорів. Однак процес навчання таких моделей є досить ресурсомістким, що ускладнює їх удосконалення в автономному режимі. Це обумовлює актуальність розробки моделей і методів навчання системи інтелектуального аналізу візуальних образів для адаптації до нових задач і середовища функціонування за умов ресурсних та інформаційних обмежень.

Розробка, що пропонується, може бути корисною з точки зору зменшення накладних витрат на експлуатацію та переконфігурування безпілотного апарату

до нових умов і задач функціонування. Тобто розробка спрямована на забезпечення ефективного функціонування і зниження вимог до обсягу розмічених навчальних даних та обчислювальних ресурсів за рахунок використання нових моделей і методів навчання. Це дозволить підвищити конкурентоздатність розроблених рішень на ринку безпілотної авіації.

Проміжний звіт складається із вступу, двох розділів, висновків і переліку посилань.

Перший розділ присвячено розробці моделей і методів навчання екстрактора ознакового опису та вирішувальних правил. Розглянуто поєднання принципів розділення пояснюючих факторів та переносу знань для побудови інформативного ознакового опису спостережень за умов обмеженого обсягу розмічених навчальних даних. Також у розділі запропоновано інкрементальні алгоритми синтезу завадозахищених вирішувальних правил.

Другий розділ присвячено реалізації інформаційної технології інтелектуального аналізу візуальних даних бортової системи безпілотної апарату. Виконано опис процесів функціонування бортової системи в режимі навчання і екзамену у вигляді діаграм відображення множин та діаграм стандарту IDEF3. Розглянуто апаратно-програмне забезпечення бортової системи. Запропоновано поєднання алгоритмів детектора і трекара рухомих об'єктів з техніками активного навчання та самонавчання для зниження трудомісткості розмітки даних та забезпечення інтерактивності машинного навчання.

Результати наукових досліджень, одержаних виконавцями проекту, опубліковано в працях [1]–[10].

1 МОДЕЛІ І МЕТОДИ НАВЧАННЯ БОРТОВОЇ СИСТЕМИ БЕЗПЛОТНОГО АПАРАТУ

1.1 Моделі і методи навчання екстрактора ознак

Формування ознакового опису спостережень за умов ресурсних та інформаційних обмежень варто здійснювати на основі принципів, що дозволяють утилізувати всю доступну інформацію як з розміченої і нерозміченої навчальних вибірок, так і з зовнішніх джерел. Вибіркові дані і без розмітки містять багато інформації про їх структуру, яку можна виділити на основі моделей розділення пояснюючих факторів (disentangle explanatory factors). Моделі розділення пояснюючих факторів як правило будують на основі автокодувальників, машин Больцмана, алгоритмів розрідженого кодування та генеративних моделей. Крім цього навчені моделі для аналізу образів в схожих задачах є зовнішніми джерелами, що акумулюють у собі досвід, який можна використати для поточної задачі. Саме в цьому полягає принцип переносу знань (Transfer Learning), згідно якого вагові коефіцієнти навчених мереж можна запозичити як квазіоптимальну стартову точку пошуку для нової задачі. Узагальнену структуру екстрактора ознак, побудованого на таких принципах показано на рис. 1.1.

Серед моделей розділення пояснюючих факторів заслуговують на увагу ті, що основані на ідеях і методах розрідженого кодування. Покладений в їх основу ефект редукції причини (explaining away) дозволяє виявити приховані фактори (першопричини) і забезпечити інформативне і завадозахищене ознакове подання вибірових спостережень навіть за умов обмеженого обсягу даних.

Для підвищення рівня абстрактності пояснюючих факторів екстрактор ознак може містити декілька шарів розрідженого кодування. Реалізація кодера може бути виконана на основі алгоритмів узгодженого переслідування (matching pursuit), ортогонального узгодженого переслідування (orthogonal matching pursuit), мішка переслідувачів (bag of pursuits) чи інших. При цьому для прискорення моделі в режимі екзамену можна обчислювально складний етап оптимізації для пошуку розріджених коефіцієнтів замінити апроксимуючим кодером.

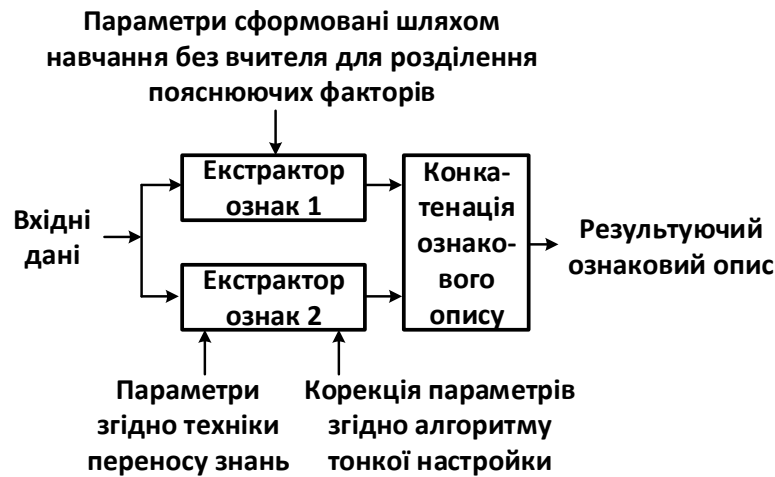


Рисунок 1.1 – Узагальнена структура екстрактора ознак

Навчальна вибірка апроксимуючого кодера формуватиметься як з вхідних даних, що кодуються, так і з результуючого розрідженого коду. На рис. 1.2 показано схему синтезу екстрактора ознак з використанням ідей і методів розрідженого кодування.

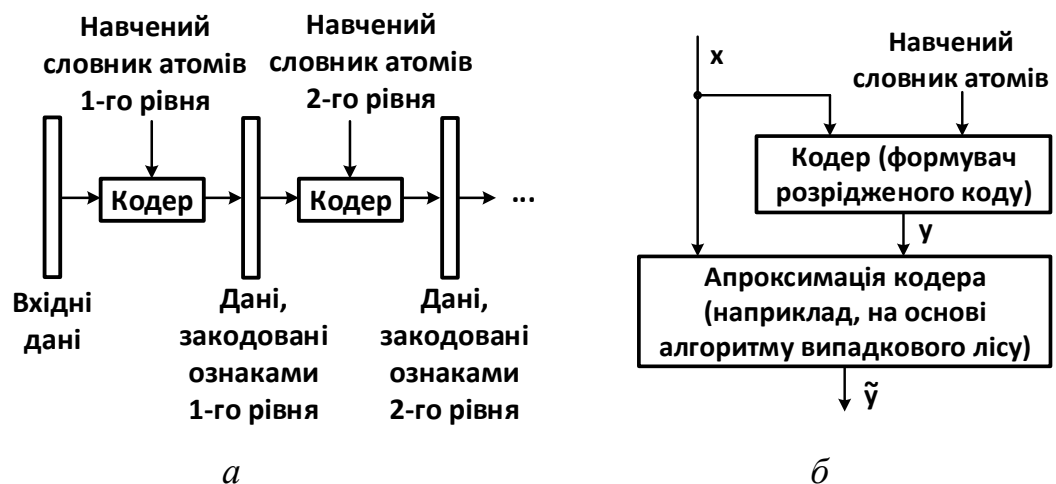


Рисунок 1.2 – Схема побудови екстрактора ознак на основі розрідженого кодування : *а* – послідовність обробки даних в багатошаровому екстракторі;
б – схема навчання апроксимованого кодера

Для розрідженого кодування використовується навчений словник атомів. Атоми мають розмірність таку ж як і будь-яке вхідне спостереження x і їх, у

найпростішому випадку, можна сформувати шляхом кластер-аналізу чи векторного квантування спостережень. В більш складному випадку навчання може здійснюватися за схемою з кінця-в-кінець і навіть з частковим залученням учителя.

Перспективним підходом до реалізації навчання словника атомів без вчителя є використання принципів нейронного газу та правила Ойа, що реалізовано в так званому алгоритмі розріджено кодуючого нейронного газу. Алгоритм нейронного газу характеризується м'якою конкуруючою схемою навчання, що призводить до більш надійної збіжності алгоритму і оптимального розподілу кластерів на вибірці вхідних даних. При цьому використання правила Ойа дозволяє сформувати словник, що забезпечує мінімальну корельованість ознак і завадозахищеність кодування спостережень.

На ефективність аналізу образів значною мірою впливає врахування апріорної інформації про топологію вхідних даних. Для 1D, 2D та 3D топологій, що знайшли широке поширення в задачах обробки зображення та послідовностей, аналіз відбувається в рамках локальних рецептивних полів. Тобто вхідні дані скануються вікном (рецептивним полем) в рамках якого відбувається розріджене кодування або згортка зі згортковими фільтрами. На рис. 1.3 показано габарити вхідних даних з відомою топологією та умовні розміри рецептивного поля для формування карти ознак.

Функцію відображення даних в рамках скануючого рецептивного поля в піксель карти ознак назвемо скануючим кодером. На рис. 1.3а показано дані з 1D-топологією у вигляді багатовимірної послідовності даних, що складається з N сигналів, які аналізуються в рамках буферу довжиною в T відліків.

На рис. 1.3б показано дані з 2D-топологією у вигляді одноканального зображення шириною W та висотою H , що сканується кодером по вертикалі і горизонталі рецептивним полем $k \times k$ пікселів. На рис. 1.3в показано дані з 3D-топологією у вигляді D -канального зображення з шириною W та висотою H , що сканується кодером по вертикалі, горизонталі і вглиб рецептивним полем $k \times k \times d$ пікселів.

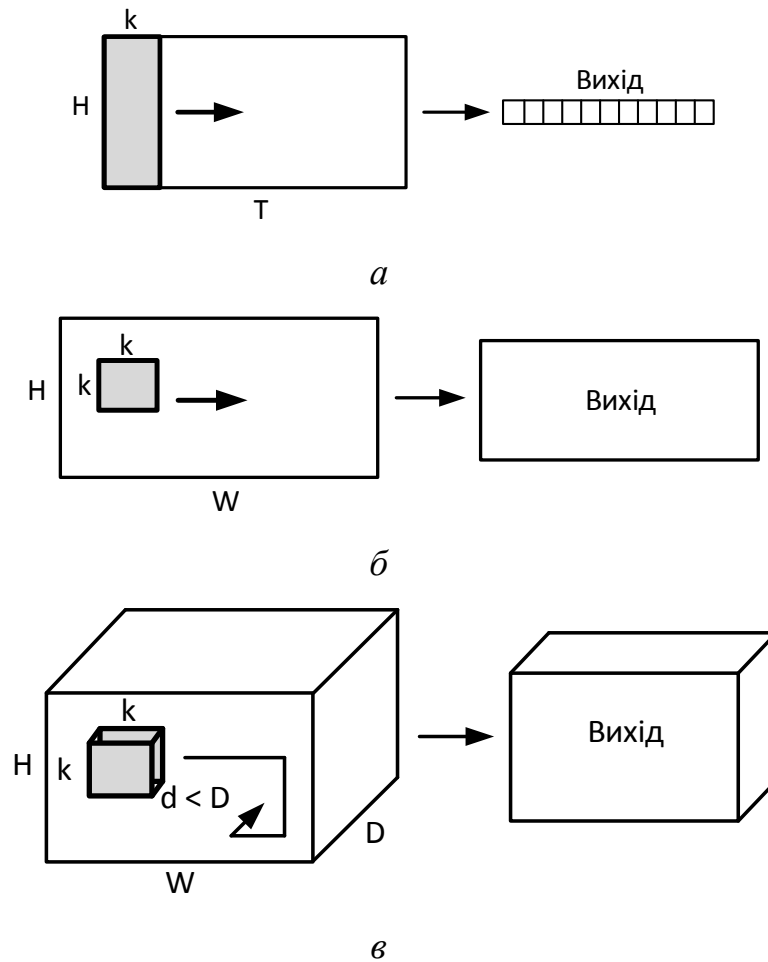


Рисунок 1.3 – Схема формування карти ознак кодуючим вікном :
a – 1D-топологія даних; *б* – 2D-топологія даних; *в* – 3D-топологія даних

У випадку зображення процес його розрідженого кодування полягає в декомпозиції на патчі, що перетинаються, кожен з яких кодується кодером в багатоканальний піксель карти ознак (рис. 1.4).

Оскільки кодер оснований на розрідженому кодуванні, то піксель карти ознак матиме розріджену активацію каналів. Отриману карту ознак можна сприймати як нове багатоканальне зображення, для якого можна повторити процедуру розрідженого кодування. При цьому кількість атомів (кластерів) кодера може бути як більшою за розмірність вхідного патчу, що відповідає надповному базису (*overcomplete dictionary*), так і меншою за розмірність вхідного патчу, що відповідає неповному базису (*undercomplete dictionary*). Дослідження показують, що для задачі класифікаційного аналізу можуть бути ефективними як надповні, так і неповні базиси екстрактора ознак [11].

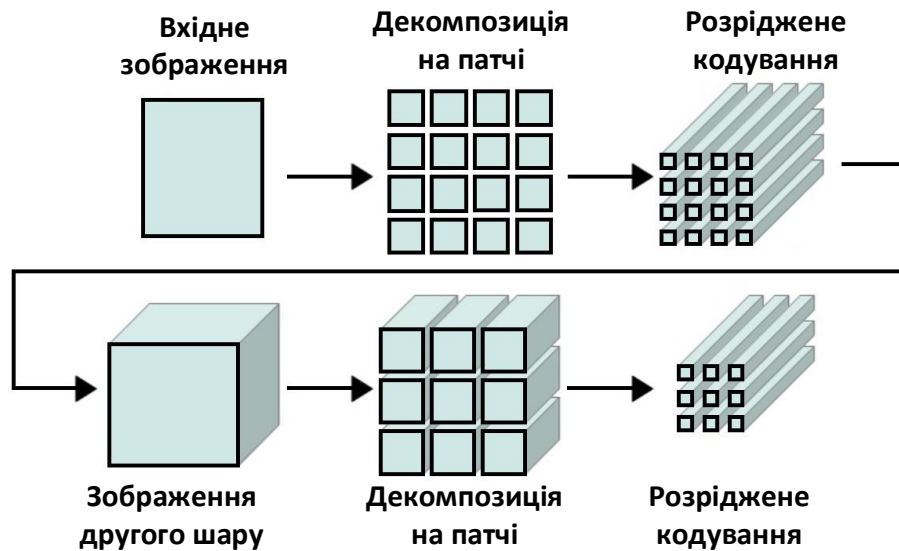


Рисунок 1.4 – Схема аналізу зображення двошаровою моделлю на основі розрідженого кодування

У задачах аналізу часових послідовностей набула популярності каузальна архітектура моделей для роботи з даними, що мають 1D-топологію. В даній архітектурі рецептивне поле кодера охоплює поточні і минулі дані. Тобто результат аналізу кожного кодера залежить тільки від значень сигналу у минулих відліках часу (рис. 1.5). При цьому для збільшення рецептивного поля моделі її необхідно стекувати в глибину.

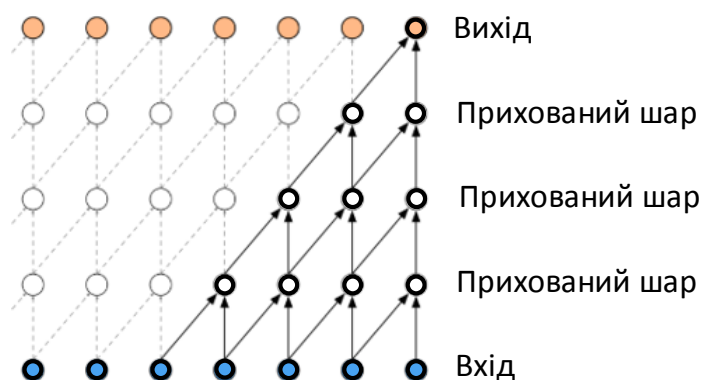


Рисунок 1.5 – Архітектура каузальної моделі для аналізу послідовностей

Для аналізу даних високої розмірності з відомою топологією можна використовувати діряві (dilated) рецептивні поля, які застосовуються до локальних ділянок даних, розмір яких перевищує кількість входів кодера. Тобто частина даних, що накривається рецептивним полем, ігнорується, а аналізу піддаються дані з певним просторовим або часовим кроком (рис. 1.6).

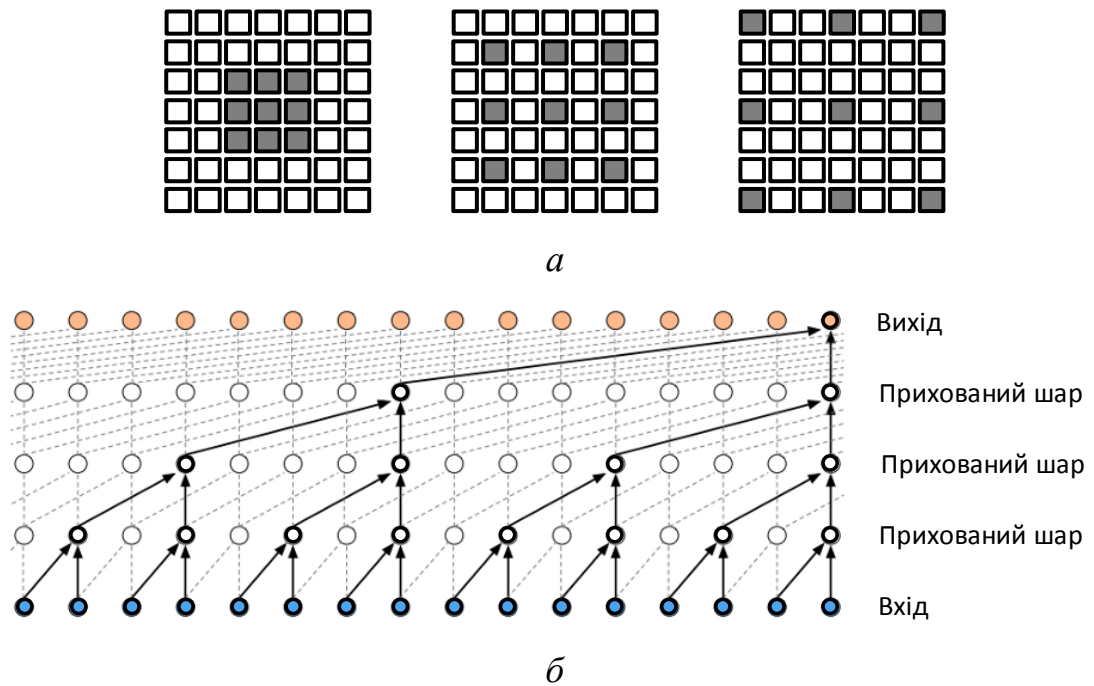


Рисунок 1.6 – Діряві рецептивні поля моделі аналізу даних :

a – 2D-топология; *б* – 1D-топология казуальної моделі

Стекування кодерів з дірявими рецептивними полями дозволяє досягти великого рецептивного поля моделі з використанням невеликої кількості шарів і параметрів. При цьому не відбувається втрат інформації, оскільки кодер здійснює сканування вхідних даних в просторі або часі. Крім того наявність пропусків дозволяє знизити чутливість до високочастотних складових шуму і, як результат, підвищити завадозахищеність моделі.

Для дослідження залежностей на різних рівнях деталізації в рамках окремого шару моделі використовують поєднання кодерів з рецептивними полями різного розміру (рис. 1.7).



Рисунок 1.7 – Приклад використання кодерів з рецептивними полями різного розміру для формування ознакового опису різного рівня деталізації

Малі рецептивні поля кодерів дозволяють здійснювати екстракцію детальної (fine-grained) інформації, а великі рецептивні поля кодерів забезпечують екстракцію грубої (coarse-grained), як правило, контекстної інформації. Наприклад в популярних Inception-модулях згорткових мереж використовуються згорткові фільтри з ядрами 1x1, 3x3 та 5x5 для сприйняття різномасштабної просторової інформації (від детального до загубленого рівнів).

Одним із шляхів ефективної регуляризації глибоких моделей аналізу даних є застосування методу Dropout, що зменшує ймовірність ефекту перенавчання і забезпечує навчання за схемою “з-кінця-в-кінець” (end-to-end). Його суть полягає у вимиканні випадковим чином частини нейронів нейронного шару під час навчання та усередненні результатів в режимі екзамену. Шар нейронів на кожному окремому кроці навчання розглядається як ансамбль експериментів Бернуллі. Множина вимкнених нейронів на кожній ітерації навчання є випадковою величиною, що розподілена за біноміальним законом.

Таким чином метод Dropout реалізує псевдоансамблювання підмереж однієї нейронної мережі. При цьому основний ефект його застосування досягається за рахунок усунування прояву взаємоадаптації (co-adaptation) нейронів. Взаємоадаптація нейронів полягає в адаптації одних нейронів для компенсації

помилки інших нейронів, однак результати даного ефекту не узагальнюються на дані, які не брали участі у навчанні. Крім очевидних переваг застосування методу Dropout існують і недоліки. Наприклад, застосування методу Dropout призводить до збільшення в 2-3 рази часу необхідного для навчання внаслідок досить зашумленого сигналу оновлення параметрів.

У випадку апіорної невизначеності виправданим є протокол інкрементального ускладнення моделі до моменту перенавчання. Тому важливе місце поряд зі схемою навчання “з-кінця-в-кінець” (end-to-end) посідає і схема навчання “шар-за-шаром” (layer-by-layer). Ця схема може використовуватися як для навчання без вчителя моделі розділення пояснюючих факторів, так і при додаванні чи тонкій настройці запозичених шарів в рамках техніки переносу знань (Transfer Learning). При цьому за умов ресурсних та інформаційних обмежень альтернативою до методу Dropout можуть бути різноманітні техніки, основані на звичайному ансамблюванні та $L1$ або $L2$ регуляризації.

Серед методів ансамблювання в задачі виділення компактного ознакового подання заслуговує на увагу алгоритм випадкового лісу, побудований з використанням методу предиктивних кластеризуючих дерев (Predictive Clustering Trees). Синтез дерев рішень в рамках методу предиктивних кластеризуючих дерев полягає в рекурсивному зменшенні внутрішньокластерної дисперсії з кожним розщепленням вузлів дерева. У результаті шлях вхідного вектора x від вершини до листка містить цінну інформацію про структуру даних. При цьому ансамбль таких дерев рішень дозволяє сформувати ознакове подання, що стійке до шуму. Екстракція компактного ознакового опису в даному випадку може здійснюватися шляхом конкатенації шляхів рішень, закодованих двійковим кодом (рис. 1.8).

Аналіз рис. 1.8 показує, що вузли кожного дерева пронумеровані, а кількість вузлів в дереві рішень рівна довжині двійкового коду. Тоді кодування вхідного спостереження полягає в присвоєнні одиничних значень тим бітам, що відповідають вузлам, через які проходить шлях прийняття рішення від кореню до листка відповідного дерева [12].

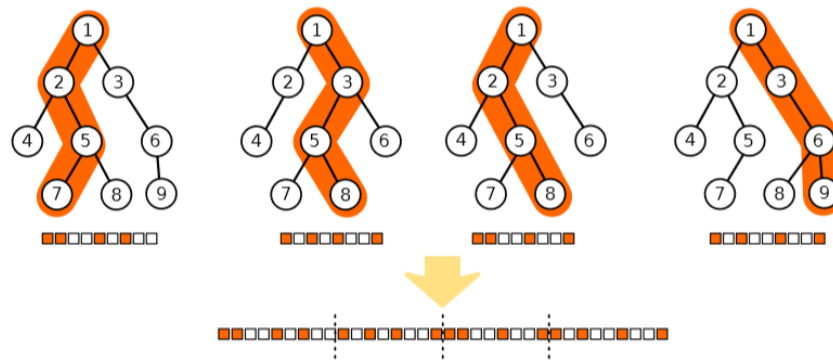


Рисунок 1.8 – Індукція ознакового опису ансамблем випадкових дерев рішень

Таким чином, формування ознакового опису полягає в застосуванні моделей і методів, що дозволяють утилізувати всю доступну апріорну інформацію для підвищення ефективності навчання. До апріорної інформації відноситься інформація про топологію даних, інформація про доменну область застосування системи (для пошуку сторонніх моделей, акумульовані знання яких можна запозичити згідно техніки переносу), а також інформація про структуру даних у вигляді множини пояснюючих факторів. Існує велика кількість варіантів щодо вибору моделі виділення пояснюючих факторів з вибірки нерозмічених даних, велика кількість конфігурацій рецептивних полів нейронів або кодерів та способів компактного і завадозахищеного подання ознакового опису найвищого рівня. Під час синтезу моделі виділення пояснюючих факторів перевагу пропонується віддавати моделям і методам, оснований на принципах розрідженого кодування. При цьому настройка параметрів і гіперпараметрів моделі екстрактора ознак ґрунтується на максимізації функціональної ефективності вирішальних правил.

1.2 Моделі і методи побудови вирішувальних правил

У працях [13, 14] було показано, що побудова вирішувальних правил в рамках геометричного підходу забезпечує завадозахищеність і зниження вимог до обсягу розмічених навчальних даних. Однією з найбільш вдалих реалізацій

даного підходу є так звані сіамські нейронні мережі. Сіамська мережа складається з двох екстракторів ознак зі спільними параметрами (рис. 1.9).

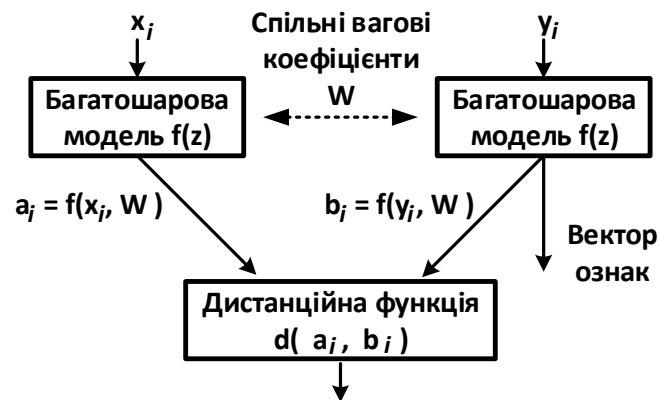


Рисунок 1.9 – Структура сіамської моделі для навчання ієрархічного екстрактора ознак з учителем

Параметри екстрактора ознак вважаються оптимальними, якщо для пари семантично схожих зразків вихід мережі має близьке до нуля значення, а для пари семантично відмінних зразків – значення виходу, що близьке до одиниці. Тобто навчання сіамських мереж можна вважати навчанням метрики подібності (metric learning). Навчена сіамська мережа може виконувати роль дистанційної метрики для лінійних алгоритмів побудови вирішальних правил, таких як метод k -найближчих сусідів. На цьому принципі оснований алгоритми навчання з декількох поглядів (few-shot learning). Також дані мережі можуть розглядатися як адаптивне ядро для ядерних алгоритмів побудови вирішальних правил. При цьому задача побудови функції близькості (similarity function) є еквівалентною до задачі побудови простору інваріантних ознак. Тому навчений екстрактор можна використовувати і окремо, без близнюка.

Навчальна вибірка сіамської нейронної мережі складається з семантично подібних пар зразків S та семантично відмінних пар зразків D , тобто

$$S = \{(x_1^{(i)}, x_2^{(i)}) : x_1^{(i)} \text{ та } x_2^{(i)} \text{ є семантично близькими з маркуванням } y^{(i)} = 0\},$$

$$D = \{(x_1^{(i)}, x_2^{(i)}) : x_1^{(i)} \text{ та } x_2^{(i)} \text{ є семантично відмінними з маркуванням } y^{(i)} = 1\}.$$

Функція втрат для сіамської мережі може мати вигляд

$$L(x_1^{(i)}, x_2^{(i)}) = y(x_1^{(i)}, x_2^{(i)}) \log p(x_1^{(i)}, x_2^{(i)}) + (1 - p(x_1^{(i)}, x_2^{(i)})) \log(1 - p(x_1^{(i)}, x_2^{(i)})) + \lambda^T |w|^2,$$

де $(x_1^{(i)}, x_2^{(i)})$ – пара зразків міні-паketу навчальної вибірки;

$y(x_1^{(i)}, x_2^{(i)})$ – вектор маркування міні-паketу;

λ^T – регуляризаційні ваги.

Для збільшення міжкласового зазору та зменшення внутрішньокласової дисперсії в просторі ознак, що формується екстрактором сіамської мережі функція втрат може набути вигляд

$$\begin{aligned} L(x_1^{(i)}, x_2^{(i)}) &= (1 - y(x_1^{(i)}, x_2^{(i)})) L_{sim}(x_1^{(i)}, x_2^{(i)}) + y(x_1^{(i)}, x_2^{(i)}) L_{dis}(x_1^{(i)}, x_2^{(i)}) = \\ &= (1 - y(x_1^{(i)}, x_2^{(i)})) \frac{1}{n} \sum_{i=1}^n p(x_1^{(i)}, x_2^{(i)}) + y(x_1^{(i)}, x_2^{(i)}) \frac{1}{n} \sum_{i=1}^n \max(0, m - p(x_1^{(i)}, x_2^{(i)})), \end{aligned}$$

де m – параметр міжкласового зазору, що може бути обчислений як $m = N/M$, де N, M – кількість ознак та кількість класів відповідно за умови, що $N > M$.

Багатокласову задачу завжди можна звести до серії двохкласових. Методи розв'язання задач двохкласової класифікації добре розроблені. При цьому часто обчислювально ефективніше працювати не з монолітним багатокласовим класифікатором, а з еквівалентною множиною двохкласових класифікаторів. До методів зведення багатокласової класифікації до серії двохкласових належать двійкове кодування кодами, що коректують помилки (Error Correcting Output Codes, ECOC), “кожен проти всіх” (one-against-all), “кожен проти кожного” (one-against-one), “турнір на вибування” (survivor), дихотомія (dichotomy), “кожен сам за себе” (everyone for himself) [14].

Під час використання кодів, що коректують помилки, номер класу записують у вигляді k -значного двійкового числа. Для цього здійснюється навчання k -класифікаторів, кожен з яких розпізнає один з k розрядів номера класу. За результатами розпізнавання вхідного вектора кожним з k класифікаторів однозначно відновлюють номер класу, до якого він належить. Якщо окремі класифікатори помиляються, то номер класу відновлюють шляхом заміни отриманого номеру номером класу, що найближчий до отриманого за метрикою Хеммінга (рис. 1.10) [13].

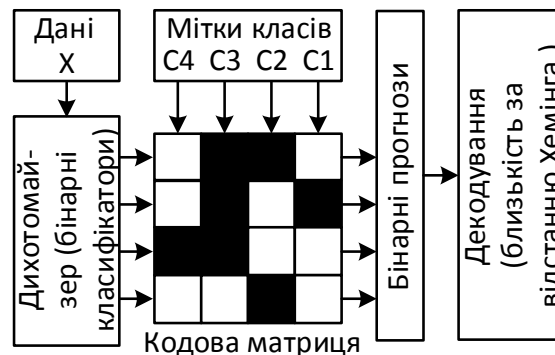


Рисунок 1.10 – Ілюстрація до схеми багатокласової класифікація з використанням двійкового кодування номера класу кодами, що виправляють помилки

На рис. 1.10 показано кодову матрицю, де чорні і білі комірочки позначають мітки 0 або 1 для зразків відповідних класів на етапі навчання бінарних класифікаторів. В режимі екзамену ця матриця використовується для порівняння прогнозованого коду з кодами класів. Ефективна кодова матриця повинна забезпечувати найбільшу відстань Хеммінга як між рядками кодової матриці, так і між стовбцями. Відстань між рядками забезпечує можливість самовиправлення помилок, а відстань між стовбцями забезпечує некорельованість результатів кожного з бінарних класифікаторів, що навчаються. При цьому існує багато способів формування номера класу для кодової матриці, однак саме коди, що виправляють помилки, (самокоректуючі коди) забезпечують кодування з

необхідними властивостями. Поширеним способом формування самокоректуючих кодів є використання матриць і кодів Адамара [14].

Основним недоліком методу ЕСОС є ігнорування структури класів під час побудови кодової матриці без можливості оптимізації коду кожного класу в процесі навчання. Крім того відсутність кодового радіусу для кожного номера класу, що вказує на кратність помилок, які можуть бути виправлені, ускладнює виявлення викидів або новизни в даних.

Синтез класифікаційних вирішальних правил пропонується здійснювати в рамках так званої інформаційно-екстремальної інтелектуальної технології (ІЕІ-технології), яка ґрунтується на таких принципах :

- двійкове кодування класів розпізнавання самокоректуючими кодами (Error-Correcting Output Coding) з урахуванням внутрішньої структури класів розпізнавання;
- трансформація простору ознак для зменшення відстані між зразками однакових класів та збільшення відстані між зразками різних класів (подібно до сіамських нейронних мереж) в рамках двійкового простору Хеммінга;
- оптимізація в інформаційному розумінні вирішальних правил, що відновлюються в радіальному базисі простору Хеммінга, з метою врахування габаритів розподілу кожного класу і підвищення стійкості до шуму і новизни в даних;
- інкрементальне нарощування складності моделі;
- використання обчислювально ефективних операцій як будівельних блоків моделі.

В рамках ІЕІ-технології процес навчання моделі класифікаційного аналізу полягає в реалізації ітераційної процедури оптимізації генотипних та фенотипних параметрів функціонування, що впливають на функціональну ефективність навчання. Одним з основних генотипних параметрів навчання моделі класифікаційного аналізу є система порогів для двійкового кодування спостережень. У найпростішому випадку ця система може бути побудована у

вигляді полів контрольних допусків, а в більш складному випадку у вигляді ансамблю дерев рішень.

Система полів контрольних допусків (СКД) на ознаки розпізнавання, що визначає координати спостережень в бінарному просторі ознак, напряму впливає на геометричні параметри замкнених роздільних гіперповерхонь, які прийнято називати контейнерами класів розпізнавання. Оптимізація СКД спрямована на таку зміну розподілу векторів спостережень в бінарному просторі, яка дозволяє оптимально в інформаційному розумінні описати розподіли класів контейнерами найпростішої (гіперсферичної) форми. Тобто алгоритм навчання повинен забезпечити у найгіршому випадку мінімальний перетин контейнерів класів, а в найкращому випадку – максимальний відступ між межами контейнерів класів (рис. 1.11).

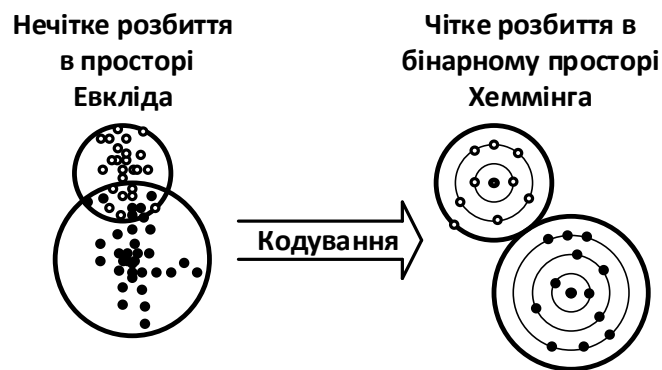


Рисунок 2.11 – Ілюстрація для пояснення ідеї трансформації простору ознак в рамках ІЕІ-технології

На рис. 1.11 показано межі гіперсферичних контейнерів і розподіл спостережень двох класів, де зразки одного з класів показані у вигляді чорних точок, а зразки іншого класу показані у вигляді незафарбованих кілець.

На рис. 1.12 показано поле контрольних допусків на значення i -ї ознаки $x_{m,i}^{(j)}$, $i = \overline{1, N}$, межі яких відраховуються від усередненого значення ознаки в базовому класі $X_B^o \in \{X_m^o\}$, який обирається розробником інформаційного забезпечення. В діагностичних системах базовому класу X_B^o відповідає клас, що

характеризує нормальний стан, щоб решту класів можна було розглядати як різноманітні відхилення від норми [15].

На рис. 1.12 прийнято такі позначення: $\bar{x}_{Б,i}$ – усереднене значення ознаки в базовому класі; $A_{\min,i}$, $A_{\max,i}$ – нижній та верхній нормовані допуски відповідно, які задають область значень i -ї ознаки і відповідних контрольних допусків; $A_{H,l,i}$, $A_{B,l,i}$ – нижня та верхня межі контрольних допусків l -го інтервалу; $\delta_{K,l,i}$ – поле контрольних допусків l -го інтервалу.

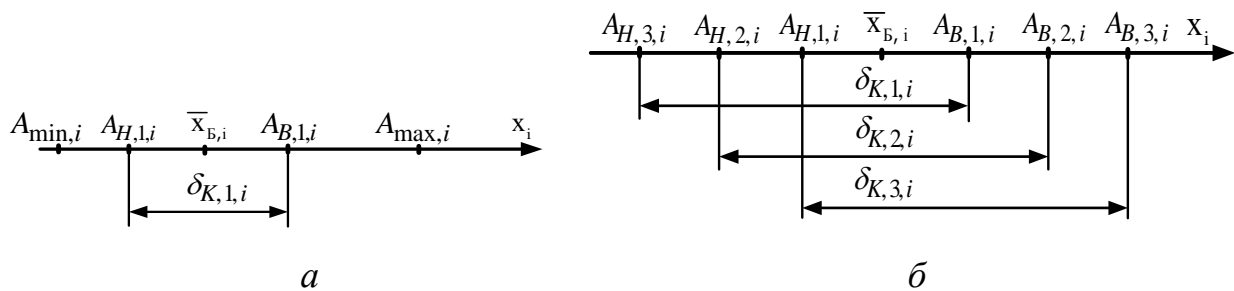


Рисунок 1.12 – Поле контрольних допусків на значення ознаки :

a – одноінтервальне поле; b – триінтервальне поле

Інформаційно-екстремальне машинне навчання основане на адаптивному двійковому кодуванні ознак розпізнавання шляхом порівняння значення i -ї ознаки з відповідними нижнім $A_{H,l,i}$ та верхнім $A_{B,l,i}$ контрольними допусками l -го інтервалу, які розраховуються за формулами

$$A_{H,l,i} = \bar{x}_{Б,i} \left[1 - \frac{\delta_i}{\delta_{\max}} \right]^l,$$

$$A_{B,l,i} = \bar{x}_{Б,i} \left[1 + \frac{\delta_i}{\delta_{\max}} \right]^l,$$

де δ_i – параметр поля контрольних допусків;

δ_{\max} – максимальне значення параметра поля контрольних допусків.

У випадку використання СКД формування бінарної навчальної матриці $\{b_{m,f}^{(j)} \mid f = \overline{1, L^* N}; l = \overline{1, L}; i = \overline{1, N}; j = \overline{1, n}; m = \overline{1, M}\}$ для L -інтервальної СКД здійснюється за правилом

$$b_{m, L^* i - L + l}^{(j)} = \begin{cases} 1, & \text{якщо } A_{H, L-l+1, i} \leq x_{m, i}^{(j)} \leq A_{B, l, i}; \\ 0, & \text{інакше} \end{cases} \quad (1.1)$$

Межі контрольних допусків ділять область можливих значень ознаки розпізнавання на 2^*L+1 областей, кожній з яких відповідає окремий двійковий код i -ї ознаки, що складається з L розрядів. Кодова відстань між кодами сусідніх областей рівна одній кодовій одиниці, а кодова від-стань між кодами областей, розміщених через одну чи більше областей, рівна двом і більше кодовим одиницям. Запропонована схема кодування (1.1) дозволяє збільшити різноманітність двійкових векторів-реалізацій та враховувати напрям відхилення розподілу векторів-реалізацій образів від базового класу, який відповідає найбільш бажаному функціональному стану.

Одним із шляхів реалізації оперативного визначення оптимальних параметрів СКД є застосування популяційних алгоритмів пошукової оптимізації. Ці алгоритми не потребують початкових наближень і дозволяють знайти оптимальне рішення за невелику кількість ітерацій. При цьому одна ітерація популяційного алгоритму потребує n_a запусків базового алгоритму інформаційно-екстремального машинного навчання, де n_a – кількість агентів популяції [13]. Базовий алгоритм інформаційно-екстремального навчання полягає в оптимізації геометричних параметрів контейнерів класів при незмінній СКД для оцінки функціональної ефективності моделі.

Одним з найпростіших в реалізації популяційних алгоритмів пошукової оптимізації є алгоритм рою частинок (Particle Swarm Optimization) [16]. Ефективність кожної частинки популяційного алгоритму, тобто близькість до глобального оптимуму, вимірюється за допомогою наперед визначеної фітнес-

функції J . Роль фітнес-функції виконує функція інформаційного критерію ефективності машинного навчання. Кожна j -та частинка крім її позиції P_j зберігає наступну інформацію: V_j – поточна швидкість частинки, P_{bestj} – краща персональна позиція частинки. Краща персональна позиція j -ї частинки – це позиція j -ї частинки, у якій значення фітнес функції для частинки було максимальним на поточний момент часу. Крім цього, з метою пошуку глобального екстремуму фітнес-функції найкраща частинка шукається в усьому рої, а позиція позначається як $Gbest$. Розглянемо основні кроки реалізації алгоритму рою частинок для оптимізації вектора параметрів поля СКД.

1. Ініціалізація рою частинок (агентів):
 - а) ініціалізація кількості частинок n_a ;
 - б) ініціалізація розмірності кожної частинки N та ініціалізація меж зміни i -ї координати j -ї частинки $\delta_{j,i}$;
 - в) ініціалізація початкових позицій частинок $P_j[0] := 100 \cdot U(0,1)$, де $U(0,1)$ – генератор випадкових чисел з діапазону $(0,1)$;
 - г) ініціалізація початкових швидкостей частинок $V_j(0) := 0$;
 - д) ініціалізація максимальної швидкості частинок в $V_{\max,i}$;
 - е) ініціалізація вагових коефіцієнтів для формули швидкості, тобто ваги інерції w та констант прискорення c_1 і c_2 .
2. Інкремент номера ітерації: $k := k + 1$.
3. Інкремент номера частинки: $j := j + 1$.
4. Інкремент номера координати в позиції: $i := i + 1$.
5. Розрахунок нового стану частинки:
 - а) розрахунок i -ї компоненти швидкості для j -ї частинки за правилами

$$V_{j,i}[k+1] := wV_{j,i}[k] + c_1a_{1,i}[k] * (P_{bestj,i}[k] - P_{j,i}[k]) + c_2a_{2,i}[k] * (Gbest_j - P_{j,i}[k]);$$

$$V_{j,i}[k+1] := \begin{cases} V_{j,i}[k+1] & \text{if } V_{j,i}[k+1] < V_{\max,i}, \\ V_{\max,i} & \text{if } \text{else}, \end{cases}$$

де $a_1[k] = U(0,1)$, $a_2[k] = U(0,1)$;

б) оновлення позиції частинки $P_j[k+1] := P_j[k] + V_j[k+1]$;

в) обчислення цільової функції $J_j[k+1]$;

г) оновлення значень найкращої персональної $Pbest$ та глобальної $Gbest$ позицій агентів пошуку

$$Pbest_j[k+1] := \begin{cases} Pbest_j[k], & \text{if } J(P_j[k+1]) \leq J(Pbest_j[k]); \\ P_j[k+1], & \text{if } \text{else}; \end{cases}$$

$$Gbest[k+1] := \arg \max_j \{J(Pbest_j[k+1])\}.$$

6. Перевірка умови зупину: якщо $k < K_{\max}$, де K_{\max} – максимальна кількість ітерацій пошуку, і $J(Gbest[k+1]) < 1,0$, то перехід до кроку 2, інакше – до кроку 7.

7. Зупин.

Використання ансамблю дерев рішень є ще одним способом двійкового кодуванням ознак. При цьому двійкове кодування вектора x_j здійснюється шляхом конкатенації результатів прийняття рішень деревами ансамблю T_1, \dots, T_L . При цьому кожне дерево рішень формує бінарний код, де кожен ненульовий біт відповідає вузлу на шляху прийняття рішень від кореня дерева до термінального вузла). Древа рішень будуються за принципом баггінгу або бустінгу (рис. 1.13) предиктивних кластеризуючих дерев рішень (Predictive Clustering Tree) [16]. В результаті вхідна навчальна вибірка кодується в бінарну навчальну матрицю $\{b_{m,s,i} \mid i = \overline{1, N_2}; s = \overline{1, n_z}; m = \overline{1, Z}\}$, де N_2 – кількість індукованих двійкових ознак; n_m – кількість навчальних зразків, що відповідають класу X_m^o .

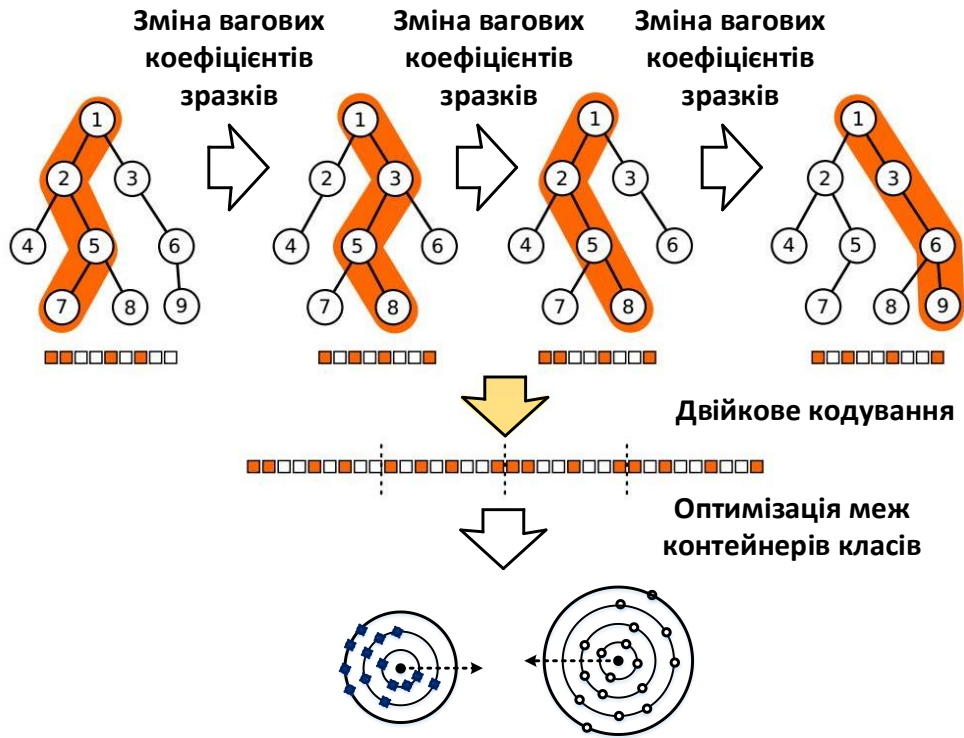


Рисунок 1.13 – Схема двійкового кодування спостережень з використанням бустінгу кластеризуючих дерев рішень

Інформаційно-екстремальний класифікатор в режимі екзамену визначає належність вхідного вектору x з відповідним двійковим поданням b до одного з класів $\{X_m^o \mid m = \overline{1, M}\}$ відповідно до максимального значення функції належності $\mu_m(b)$ на основі процедури $\arg \max_m \{\mu_m(b)\}$. При цьому функція належності $\mu_m(b)$ до контейнера з опорним вектором b_m^* та радіусом d_m^* , обчислюється за однією з формул :

$$\mu_m(b) = \exp \left(- \sum_{i=1}^{N_2} b_i \oplus b_{m,i}^* / d_m^* \right). \tag{1.2}$$

$$\mu_m(b) = 1 - \sum_{i=1}^{N_2} b_i \oplus b_{m,i}^* / d_m^* \tag{1.3}$$

Таким чином, інформаційно-екстремальний класифікатор, який оцінює належність j -го набору даних $x^{(j)}$ з N_1 ознаками до одного з класів Z , виконує кодування ознак за допомогою композиції дерев рішень і побудову вирішальних правил в радіальному базисі двійкового простору Хеммінга. На вхід алгоритму надходить максимально допустима кількість кодуючих дерев K , навчальний набір $D = \{x^{(j)}, y^{(j)} \mid j = \overline{1, n}\}$, де n розмір набору даних, а $y^{(j)}$ це мітка j -го навчального зразка, що належить множині $\{X_z^\circ \mid z = \overline{1, Z}\}$.

Навчання інформаційно-екстремального класифікатора з використанням бустінгу виконується наступним чином :

- 1) Ініціалізація вагових коефіцієнтів $w^{(j)} = 1/n$.
- 2) Для $k = 1, \dots, K$ виконуються наступні кроки.
- 3) Генерація D_k набору навчальних даних з усього набору D з використанням функції розподілу ймовірності $P(X = x^{(j)}) = w^{(j)}$.
- 4) Навчання дерева рішень T_k на наборі D_k .
- 5) Бінарне кодування вектору $x^{(j)}$ з набору даних D шляхом об'єднання шляхів рішення в деревах T_1, \dots, T_k . Результатом роботи цього етапу є бінарна матриця $\{b_{z,i}^{(s)} \mid i = \overline{1, N_2}; s = \overline{1, n_z}; z = \overline{1, Z}\}$, де N_2 кількість індукованих бінарних ознак та n_z кількість реалізацій відповідного класу X_z° , що задовольняє рівність $n = \sum_z n_z$.
- 6) Побудова інформаційно-екстремальних вирішальних правил в радіальному базисі бінарного простору Хеммінга та обчислення інформаційного критерію

$$E_z^* = \max_{\{d\}} E_z(d), \quad (1.4)$$

де $\{d\} = \left\{ 0, 1, \dots, \left(\sum_i b_{z,i} \oplus b_{c,i} - 1 \right) \right\}$ – набір концентричних радіусів з центром b_z ;

b_z, b_c – опорні вектори розподілу даних в класі X_z° та сусідньому до нього класі X_c° відповідно, які можна розрахувати за правилом

$$b_{z,i} = \begin{cases} 1, \text{if } \frac{1}{n_z} \sum_{s=1}^{n_z} b_{z,i}^{(s)} > \frac{1}{Z} \sum_{c=1}^Z \frac{1}{n_c} \sum_{s=1}^{n_c} b_{c,i}^{(s)}; \\ 0, \text{otherwise.} \end{cases} \quad (1.5)$$

де E_z – критерій ефективності побудови вирішальних правил в режимі навчання для класу X_z° [8], який розраховується як нормалізована модифікація інформаційної міри Кульбака [9]

$$E_z = \frac{1 - (\alpha_z + \beta_z)}{\log_2(2 + \varepsilon) - \log_2(\varepsilon)} \log_2 \left[\frac{2 - (\alpha_z + \beta_z) + \varepsilon}{(\alpha_z + \beta_z) + \varepsilon} \right], \quad (1.6)$$

де α_z, β_z – частота хибно позитивних та хибно негативних спрацювань вирішальних правил класу X_z° ; ε будь-яке мале додатнє число, яке вводиться, для уникнення невизначеності при діленні на нуль.

Задля підвищення ефективності навчання, загальноприйнятим шляхом є зведення проблеми багатокласової класифікації до серії двокласових класифікацій за принципом «один проти всіх». Для уникнення проблеми незбалансованості класів розпізнавання, обумовленої переважанням у навчальному наборі даних негативних зразків, здійснюється введення синтетичного класу, який є альтернативним для X_z° . Синтетичний клас представлено n_z векторами з інших класів, які найбільш близькі до опорного вектора b_z , де n_z – обсяг навчального набору даних класу X_z° .

7) Тестування одержаних інформаційно-екстремальних вирішальних правил на наборі даних D та розрахунок коефіцієнту помилок для кожної реалізації з D . При цьому, в режимі екзамену прийняття рішення про належність вектора b до одного з класів розпізнавання з алфавіту $\{X_z^\circ \mid z = \overline{1, Z}\}$ здійснюється за максимальним значенням функції належності $\mu_b(b)$ відповідно до виразу $\arg \max_z \{\mu_z(b)\}$. У цьому випадку функція належності $\mu_z(b)$ бінарного подання b вхідного вектора даних x до класу X_z° , оптимальний контейнер якого має опорний вектор b_z^* та радіус d_z^* , розраховується за формулою (1.2).

8) Оновлення вагових коефіцієнтів $\{w^{(j)}\}$ пропорційно до помилок розпізнавання вектора $x^{(j)}$:

$$w^{(j)} = 1 - \mu_{m'}(x^{(j)}), m' = y^{(j)};$$

$$w^{(j)} = \frac{w^{(j)}}{\sum_j w^{(j)}}.$$

9) Якщо $|E_k^* - E_{k-1}^*| < \varepsilon$ то вихід з циклу.

У найпростішому випадку регресійні вирішальні правила будуються у вигляді лінійної моделі, вагові коефіцієнти якої знаходять згідно методу найменших квадратів. Однак в багатозадачних моделях зі спільним ознаковим описом більш ефективним вважається використання регресійної моделі з, як мінімум, одним прихованим шаром.

Розглянемо регресійну модель у вигляді нейромережі прямого поширення з одним прихованим шаром. При цьому набір навчальних даних описується множиною $\{(x^{(j)}, y^{(j)}) \mid x^{(j)} \in R^N, y^{(j)} \in R^M, 1 \leq j \leq n\}$, де $x^{(j)} = (x_1^{(j)}, x_2^{(j)}, \dots, x_N^{(j)})^T$. Вихідна змінна $y^{(j)} \in R^M$ відповідає вектору цілі,

тобто є міткою навчального зразка. Нейромережа з R адитивними прихованими вузлами та функцією активації $\varphi(x)$ може бути подана у вигляді системи рівнянь

$$\sum_{r=1}^R \beta_r \varphi(w_r^T x^{(j)} + b_r) = o^{(j)}, 1 \leq j \leq n$$

де $w_r = (w_{r,1}, w_{r,2}, \dots, w_{r,N})^T$ – вектор вагових коефіцієнтів, який зв'язує вхідний шар з r -м прихованим вузлом;

b_r – зміщення r -го прихованого вузла;

$\beta_r = (\beta_{r,1}, \beta_{r,2}, \dots, \beta_{r,M})$ – вектор вагових коефіцієнтів, що зв'язує вихідний шар з r -м прихованим вузлом; $o^{(j)}$ – вихід мережі для вхідного вектору $x^{(j)}$;

$\varphi(x)$ – функція активації.

Нейромережа з R прихованими вузлами може відтворити ці n зразків з заданою точністю, якщо всі параметри можуть вільно коригуватись, тобто існують, β_r , w_r і b_r . Вище наведені вирази можуть бути компактно переписані як матрична рівність

$$H\beta = Y, \quad (1.7)$$

де

$$H = \begin{bmatrix} \varphi(w_1^T x^{(1)} + b_1) & \dots & \varphi(w_R^T x^{(1)} + b_R) \\ \dots & \dots & \dots \\ \varphi(w_1^T x^{(n)} + b_1) & \dots & \varphi(w_R^T x^{(n)} + b_R) \end{bmatrix}_{n \times R},$$

$$\beta = \begin{bmatrix} \beta_1 \\ \dots \\ \beta_R \end{bmatrix}_{R \times M}, \quad Y = \begin{bmatrix} y^{(1)} \\ \dots \\ y^{(n)} \end{bmatrix}_{n \times M},$$

де H – матриця виходу прихованого шару.

Навчання даної мережі можливо здійснити з використанням алгоритму зворотного поширення помилки. Однак нестабільність результатів, чутливість до гіперпараметрів навчання та повільна збіжність обумовлюють непридатність даного алгоритму за умов ресурсних обмежень. При цьому важливим гіперпараметром є кількість нейронів прихованого шару, що найважче піддається оцінці. Одним з рішень є інкрементальне ускладнення моделі до моменту досягнення необхідної точності за навчальною та тестовою вибірками. Для прискорення навчання можна скористатися принципами машин екстремального навчання (Extreme Learning Machine), де вагові коефіцієнти прихованого шару ініціалізуються шляхом генерації випадкових чисел або ортогональних випадкових матриць. Вагові коефіцієнти вихідного шару визначаються шляхом псевдоінверсії, що може бути обчислена на основі алгоритму ортогоналізації Грамма-Шмідта. Збіжність ітераційного алгоритму ортогоналізації Грамма-Шмідта для пошуку рішення задачі найменших квадратів було доведено у праці Li Ying [8]. Розглянемо основні кроки навчання машини екстремального навчання з інкрементальним збільшенням кількості прихованих вузлів і визначенням вагових коефіцієнтів вихідного шару на основі алгоритму ортогоналізації Грамма-Шмідта:

1. Встановити максимальну кількість ітерацій L_{\max} , початкову залишкову помилку $E = [y^{(1)}, \dots, y^{(j)}]^T$ та допустиму помилку E_0 .
2. Для $L = 1 \dots L_{\max}$ виконуються наступні кроки.
3. Збільшити кількість прихованих вузлів на один : $r = r + 1$.
4. Випадковим чином згенерувати один прихований нейрон та розрахувати вектор його виходу h_r .
5. Якщо $r = 1$ тоді $v_r = h_r$ інакше

$$v_r = h_r - \frac{\langle v_1, h_r \rangle}{\langle v_1, v_1 \rangle} v_1 - \frac{\langle v_2, h_r \rangle}{\langle v_2, v_2 \rangle} v_2 - \dots - \frac{\langle v_{r-1}, h_r \rangle}{\langle v_{r-1}, v_{r-1} \rangle} v_{r-1} \quad (1.8)$$

6. Якщо $\|v_r\| \geq \varepsilon$, то виконувати обчислення вихідної ваги для нового прихованого вузла $\beta_r = v_r^T E / (v_r^T v_r)$ та розрахувати нову залишкову помилку $E = E - v_r \beta_r$ інакше $r = r - 1$.

7. Якщо $\|E\| < E_0$ або помилка на тестовій вибірці не зменшується протягом заданої кількості ітерацій, то вихід з основного циклу роботи.

Приведений алгоритм повинен успішно працювати у випадку добре підготованого ознакового опису, інакше може знадобитися велика кількість прихованих вузлів. Тому слід розглянути і алгоритми навчання багат шарових регресійних моделей, які дозволяють зменшити загальну кількість нейронів. При цьому для економії оперативної пам'яті слід розглянути можливість навчання в міні-пакетному режимі або в режимі онлайн (стохастичний алгоритм). Крім алгоритму зворотного поширення помилки, що оснований на алгоритмі градієнтного спуску, можливі альтернативні варіанти. Найпростішим варіантом є використання псевдоінверсії Мура-Пенроуза для вирішення лінійних рівнянь типу $Ax=b$ на основі сингулярного розкладу матриці. У випадку навчання в міні-пакетному чи онлайн режимах обчислення псевдоінверсії є досить швидкою операцією, оскільки розмір матриць буде досить невеликим. Це дозволяє знайти необхідну корекцію параметрів нейрона для досягнення бажаного виходу і одночасно оновити ціль для попереднього нейронного шару [17]. Однак в даному випадку існує потреба у використанні функцій активації $activation(\square)$, які мають обернене відображення з такою ж областю визначення. До таких функцій можуть належати ReLU-подібні функції типу LeakyReLU, Softplus, S-ReLU та інші.

На рис. 2.13 показано псевдокод алгоритму прямого поширення сигналу для нейронного шару з K нейронів. На вхід нейронного шару надходить N -вимірний вектор спостереження $x = [x_1, \dots, x_i, \dots, x_N]$. Вихід нейронного шару формує K -вимірний вектор $o = [o_1, \dots, o_k, \dots, o_K]$. При цьому кожен i -й вхід нейрона має ваговий коефіцієнт w_i та коефіцієнт зміщення b_i .

для k від 1 до K :

для i від 1 до N :

$$o_k = o_k + x_i * w_i + b_i$$

$$o_k = \text{activation}(o_k)$$

Рисунок 2.13 – Псевдокод алгоритму прямого поширення сигналу для нейронного шару

для k від 1 до K :

для i від 1 до N :

$$y_{k,i} = \text{activation}^{-1}(y_k) / N$$

$$o_{k,i} = x_i * w_{k,i} + b_{k,i}$$

$$\begin{bmatrix} \Delta w_{k,i} \\ \Delta b_{k,i} \end{bmatrix} = [x_i \quad 1]^{\dagger} \times [(o_{k,i} - y_{k,i})]$$

$$\tilde{w}_{k,i} = w_{k,i} - \alpha \Delta w_{k,i}$$

$$\tilde{b}_{k,i} = b_{k,i} - \alpha \Delta b_{k,i}$$

$$\tilde{x}_{k,i} = \frac{y_{k,i} - \tilde{b}_{k,i}}{\tilde{w}_{k,i}}$$

для i від 1 до N :

$$\tilde{x}_i = \frac{1}{K} \sum_{k=1}^K \tilde{x}_{k,i}$$

Рисунок 2.14 – Псевдокод алгоритму зворотного поширення сигналу для нейронного шару мережі

На рис. 2.14 показано псевдокод алгоритму зворотного поширення сигналу для корекції вагових коефіцієнтів і оновлення цілі для попереднього шару нейромережі. На вхід алгоритму зворотного поширення сигналу в заданому нейронному шарі надходить вхідний вектор нейронного шару $x = [x_1, \dots, x_i, \dots, x_N]$ та очікуваний (бажаний) вектор вихідного сигналу нейронного шару

$y = [y_1, \dots, y_k, \dots, y_K]$. При цьому вихід нейрона формується під впливом кожного його входу, тому цільове значення $y_{k,i}$ для i -го входу k -го нейрона є складовою результату застосування оберненої функції активації до цільової змінної y_k . Для простоти обчислень цільова змінна рівномірно розподіляється між кожним входом нейрона, тобто $y_{k,i} = \text{activation}^{-1}(y_k) / N$.

Якщо ввести бажаний ваговий коефіцієнт $\tilde{w}_{k,i}$ та бажане зміщення $\tilde{b}_{k,i}$ i -го входу k -го нейрона, які забезпечують бажану складову виходу $y_{k,i}$, тоді можна скласти наступну систему рівнянь :

$$\begin{cases} x_i * w_{k,i} + b_{k,i} = o_{k,i} \\ x_i * \tilde{w}_{k,i} + \tilde{b}_{k,i} = y_{k,i} \end{cases} \Rightarrow \begin{cases} x_i * \tilde{w}_{k,i} + \tilde{b}_{k,i} = y_{k,i} \\ \tilde{w}_{k,i} = w_{k,i} - \Delta w_{k,i} \\ \tilde{b}_{k,i} = b_{k,i} - \Delta b_{k,i} \end{cases} \Rightarrow \\ \Rightarrow x_i * \Delta w_{k,i} + \Delta b_{k,i} = (o_{k,i} - y_{k,i}),$$

де $\Delta w_{k,i}$, $\Delta b_{k,i}$ – необхідний сигнал корекції для отримання бажаної складової вихідного сигналу $y_{k,i}$.

Вище наведені рівняння дозволяють записати задачу тренування параметрів нейрона у формі, що дозволяє застосувати псевдоінверсію Мура-Пенроуза.

$$\begin{bmatrix} x_i & 1 \end{bmatrix} \times \begin{bmatrix} \Delta w_{k,i} \\ \Delta b_{k,i} \end{bmatrix} = \begin{bmatrix} (o_{k,i} - y_{k,i}) \end{bmatrix} \Rightarrow \\ \begin{bmatrix} \Delta w_{k,i} \\ \Delta b_{k,i} \end{bmatrix} = \begin{bmatrix} x_i & 1 \end{bmatrix}^\dagger \times \begin{bmatrix} (o_{k,i} - y_{k,i}) \end{bmatrix}$$

Для забезпечення стабільності навчання і уникнення помітного впливу шумового навчального зразка $x = [x_1, \dots, x_i, \dots, x_N]$ на результат навчання задається достатньо мале значення швидкості навчання α . Значення α залежить від

кількості епох навчання, обсягу навчальних даних і для більшості практичних задач може бути обране з діапазону $[0,01-0,0001]$.

Скореговані значення вагового коефіцієнта $\tilde{w}_{k,i}$ та зміщення $\tilde{b}_{k,i}$ можуть бути використані для обчислення скорегованого вхідного сигналу $\tilde{x}_{k,i}$, який забезпечуватиме бажане значення складової виходу $y_{k,i}$. При цьому цільовий сигнал попереднього шару нейронної мережі може бути обчислений як усереднене за нейронами даного шару значення скорегованих входів

$$\tilde{x}_i = \frac{1}{K} \sum_{k=1}^K \tilde{x}_{k,i}.$$

Таким чином, у найпростішому випадку регресійні вирішальні правила можуть бути сформовані у вигляді лінійної комбінації ознак розпізнавання, але в більш складному випадку існує потреба в багат шаровій архітектурі. Синтез цих правил полягатиме у вирішенні задачі найменших квадратів на основі алгоритму градієнтного спуску, псевдо-інверсії Мура-Пенроуза або ортогоналізації Грамма-Шміда. При цьому безградієнтні алгоритми навчання менш чутливі до гіперпараметрів.

У задачах глобальної оптимізації набули популярності мета-евристичні пошукові алгоритми, що не висувають умов диференційованості чи опуклості досліджуваної функції. Найбільш ефективними з точки зору точності рішень вважаються популяційні алгоритми, до яких належать алгоритм рою бджіл (Artificial Bee Colony Algorithm), алгоритм косяку риб (Fish School Search Algorithm), алгоритм зграї птахів (Bird Swarm Algorithm), алгоритм рою частинок (Particle Swarm Optimization Algorithm), гравітаційний алгоритм (Gravitational Search Algorithm), еволюційний алгоритм (Evolutionary Algorithm), алгоритм диференційної еволюції (Differential Evolution Algorithm) та інші [18].

Істотним недоліком популяційних алгоритмів є значні ресурсні потреби для вирішення задач з великою кількістю параметрів. Наприклад, в задачі оптимізації

глибоких нейронних мереж популяційному алгоритму знадобиться зберігати в пам'яті популяцію таких мереж з модифікованими параметрами. Сучасні глибокі мережі займають великий обсяг пам'яті і потребують суттєві обчислювальні ресурси для обчислення фітнес-функції. Однак в задачі тонкої настройки моделі область пошуку (exploration space) дещо звужена. Тому можливий ряд модифікацій алгоритму, обмежень розміру популяції та кроку модифікації рішень, які можуть забезпечити практичну придатність популяційних алгоритмів для тонкої настройки глибоких моделей. Одним з характерних представників даного підходу є еволюційний алгоритм з обмеженою оцінкою (limited evaluation evolutionary algorithms). Крім того, зменшити область пошуку можна і за рахунок роздільної оптимізації шарів (рис.1.6б) [17].

Еволюційний алгоритм з обмеженою оцінкою може здійснювати оцінку популяції на обмеженій кількості вхідних навчальних зразків подібно до алгоритму стохастичного градієнтного спуску, який здійснює корекцію вагових коефіцієнтів мережі після обробки міні-паketу (mini-batch). Використання міні-паketу замість одного зразка зменшує ймовірність значного впливу на результат навчання шумових зразків. При цьому в еволюційному алгоритмі передбачено модифікацію фітнес-функції для уникнення втрати на етапі селекції агентів популяції, що є більш ефективними у глобальному сенсі, але менш пристосовані до конкретного вхідного міні-паketу. Під час обчислення фітнес-функції враховується як ефективність агентів для поточного міні-паketу, так і ефективність предків окремих агентів на минулих міні-пакетах.

В еволюційному алгоритмі з обмеженою оцінкою використовуються окремо фітнес-функція для випадку репродукції схрещуванням (1.9) і окремо фітнес-функція для асексуальної репродукції (1.10) відповідно :

$$f' = \frac{f_{p_1} + f_{p_2}}{2}(1-d) + f, \quad (1.9)$$

$$f' = f_{p_1}(1-d) + f, \quad (1.10)$$

де f' – скоригована індивідуальна фітнес-функція агента популяції;

f – індивідуальна фітнес-функція агента популяції для поточного міні-паketу;

f_{p_1} – фітнес-функція першого предка;

f_{p_2} – фітнес-функція другого предка;

d – константне значення затухання впливу історії предків.

Агенти популяції прямо кодують параметри моделі $W \in \mathbb{R}^C$, де C – кількість параметрів моделі. Операції репродукції та мутації напряму модифікують вектор параметрів кожного агента. Оператор репродукції схрещуванням може бути рівномірний (uniform crossover), під час якого нащадок $W_{u,i}$ утворюється шляхом випадкової копії елементів обох батьківських агентів (1.11). Також оператор репродукції схрещуванням може бути поданий у вигляді процедури арифметичного усереднення (Arithmetic crossover) параметрів від батьківських агентів (1.12) для формування дочірнього агента W_a . Формули операторів репродукції показано нижче :

$$W_{u,i} = \begin{cases} W_{1,i}, & \text{з ймовірністю } 0,5; \\ W_{2,i}, & \text{в протилежному випадку;} \end{cases} \quad (1.11)$$

$$W_a = \frac{1}{2}(W_1 + W_2) . \quad (1.12)$$

Оператор мутації здійснює додавання шуму з Гаусовим розподілом $N(0,1)$ до параметрів батьківського агента $W_m = W_1 + \sigma N(0,1)$, де σ – сила мутації. При цьому сила мутації σ є важливим гіперпараметром, який бажано зменшувати з кожною епохою, подібно до того, як в алгоритмі градієнтного спуску зменшується швидкість навчання. На рис. 2.15 показано псевдокод еволюційного алгоритму з обмеженою оцінкою.

Доки $iter < max_iter$ виконувати :

- Вибірка міні – пакету навчальних зразків*
- Оцінка кожного агента популяції*
- Модифікація фітнес – функції для кожного агента з урахуванням ефективності предків*
- Селекція методом рулетки*
- Формування нащадків шляхом схрещування або мутації*
- Зменшення сили мутації шляхом множення на константу послаблення*

$iter \leftarrow iter + 1$

Рисунок 2.15 – Псевдокод еволюційного алгоритму з обмеженою оцінкою

Для реалізації тонкої настройки моделі пропонується першу популяцію формувати шляхом застосування оператора мутації до існуючого рішення. При цьому один з агентів першої популяції пропонується сформувати шляхом простого копіювання існуючого (вхідного) рішення.

Таким чином, у випадку виконання диференційованості функції, якою описується модель аналізу даних, ефективним методом тонкої настройки є використання алгоритмів зворотного поширення помилки. У випадку порушення даної умови, що часто буває під час синтезу гібридних моделей, пошук слід здійснювати метаевристичними популяційними або траєкторними алгоритмами. Ці алгоритми здатні вирішувати оптимізаційну задачу типу “чорної шухляди” і гармонізувати різномірні частини гібридної моделі під час тонкої настройки.

2 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІЗУ ДАНИХ БОРТОВОЇ СИСТЕМИ БЕЗПІЛОТНОГО АПАРАТУ

2.1 Апаратно-програмне забезпечення бортової системи безпілотного апарату

На основі отриманих в дисертаційній роботі наукових результатів розроблено структурні та функціональні схеми, інформаційне та програмне забезпечення інтелектуальної бортової системи малогабаритного БПЛА для автономного аеро-відео-моніторингу місцевості. Адаптація системи до нових умов функціонування і задач здійснюється на основі машинного навчання, що формує вирішальні правила інтелектуальної бортової системи.

Загальну функціональну схему БПЛА типу квадрокоптер з бортовою системою автономного аеровідеомоніторингу місцевості показано на рис. 2.1.

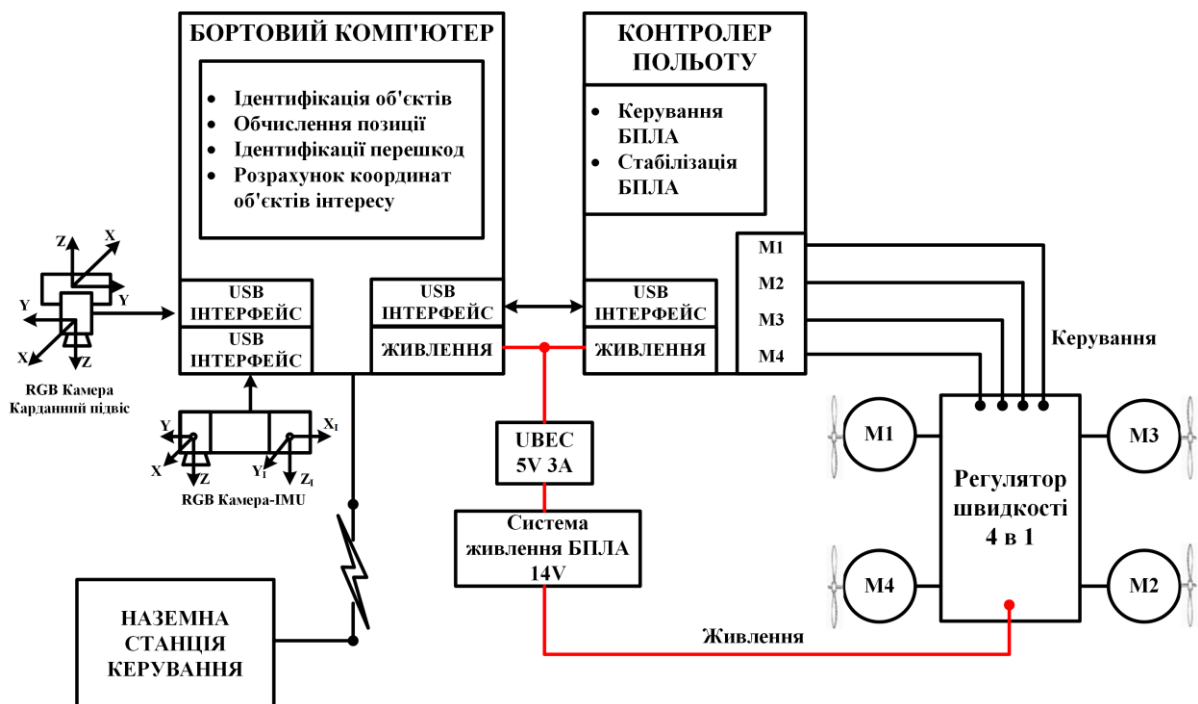


Рисунок 2.1 – Функціональна схема БПЛА для аеровідеомоніторингу місцевості, що навчається

Відповідно до схеми наведеної на рис. 2.1 функції зовнішньої комунікації, формування команд для контролера польоту, автономної навігації та додаткові функції аналізу даних бере на себе модуль бортового комп'ютера. Отримані за допомогою бортових камер візуальні знімки та дані від інерційних вимірювальних пристроїв подаються до бортового комп'ютера через послідовний інтерфейс де, після їх попередньої обробки, виконується аналіз даних : розрахунок поточного положення апарату, ідентифікація об'єктів інтересу на вхідних зображеннях та прогнозування можливих перешкод. Після проведення аналізу зображень та знаходження об'єктів інтересу виконуються розрахунок координат знайдених об'єктів з використанням інформації про поточне положення БПЛА, що отримане від модуля обчислення позиції. Після цього результат роботи може бути переданий через модуль зв'язку до наземної станції, або збережений в бортовій пам'яті БПЛА. Інформація про поточні координати і орієнтацію камери БПЛА та можливі перешкоди на шляху руху надходять до модулю коригування траєкторії польоту. Цей модуль виконує відповідний розрахунок та формує завдання для польотного контролера на корекцію руху БПЛА та інформує користувача про можливу зміну траєкторії руху БПЛА за необхідністю. Контролер польоту здійснює керування двигунами БПЛА відповідно до траєкторії руху отриманої від бортового комп'ютера. Одночасно з цим на контролер польоту здійснює неперервну стабілізацію апарату для компенсації поривів вітру і інших зовнішніх впливів. Наземна станція керування представляє собою пристрій, на який покладено функції модуля візуалізації інформації та людино-машинної взаємодії з БПЛА на відстані по бездротовому каналу зв'язку. При цьому наземна станція керування дозволяє оцінити якість роботи системи та за необхідності задіяти режим навчання/тонкої настройки бортової системи БПЛА для адаптації до нових умов середовища функціонування.

Враховуючи, що система аеровідеомоніторингу місцевості є розподіленою і передбачає взаємодію та обмін даними між її складовими частинами, то архітектура програмного забезпечення системи побудована відповідно до принципів об'єктно-орієнтованого та функціонального проектування. При цьому

обмін інформацією між модулями реалізований на основі клієнт-серверної архітектури.

На рис. 4.2 показано структурну схему програмного забезпечення бортової системи БПЛА для автономного аеро-відео-моніторингу місцевості.

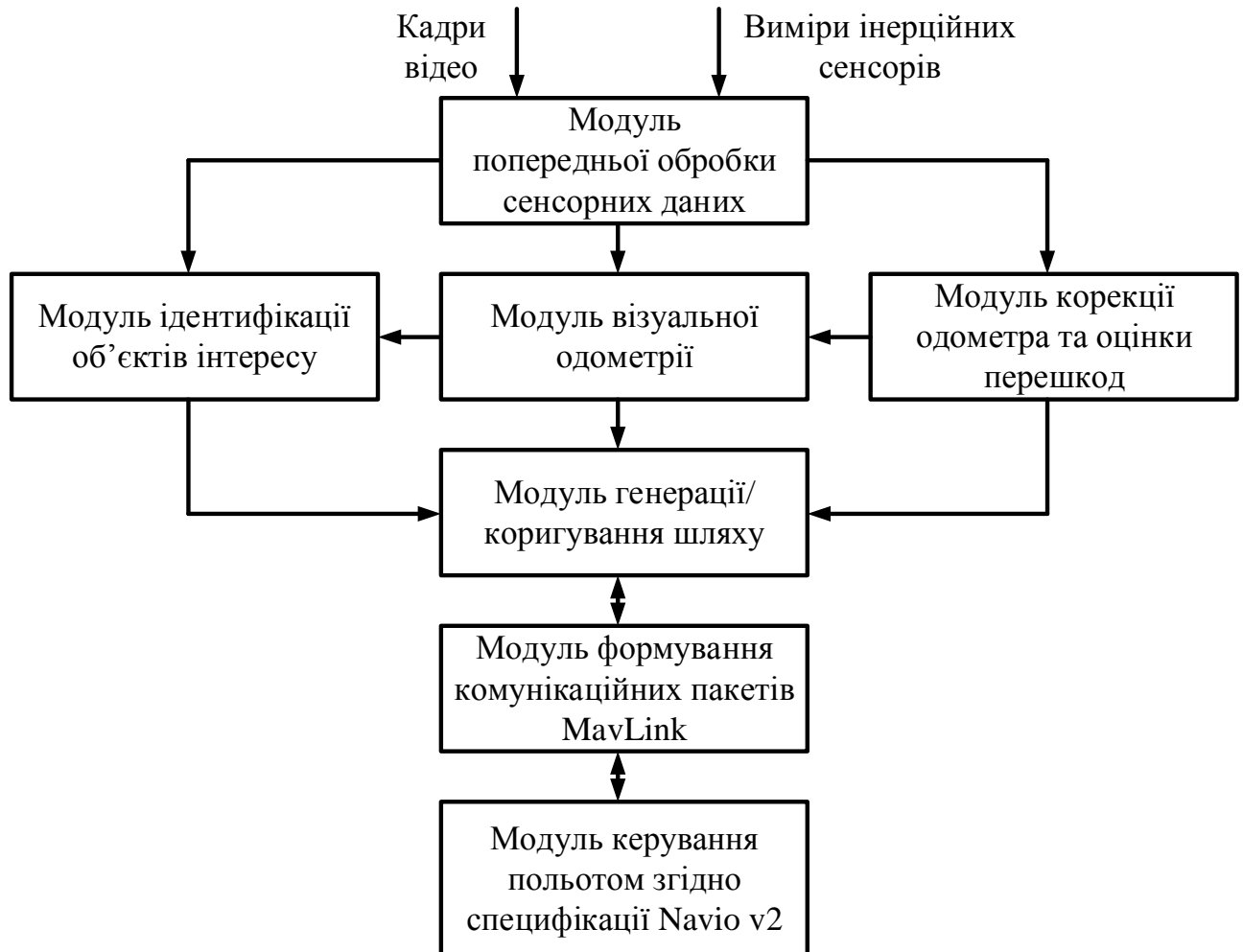


Рисунок 2.2 – Структурна схема програмного забезпечення інтелектуальної бортової системи безпілотної літальної апарату

Аналіз рис. 2.2 показує, що до складу програмного забезпечення інтелектуальної бортової системи аналізу даних повинні входити модуль ідентифікації об'єкта інтересу для розпізнавання і визначення його координат. При цьому, модуль візуальної одометрії обчислює локальну позицію апарату в системі NED (North-East-Down) для її перерахунку в глобальні координати, що дозволяє визначити локацію об'єкта інтересу у відповідних системах координат.

Модуль корекції одометра та оцінки перешкод здійснює паралельну корекцію результатів оцінки переміщення та пропонує повороти для уникнення прогнозованих перешкод. Решта модулів є компонентами операційної системи робота, що вільно поширюється під ліцензією LGPL (GNU Lesser General Public License).

Інструментом і засобом розгортання запропонованої інформаційної технології є бібліотеки лінійної алгебри та графових обчислень. При цьому було використано функції лінійної алгебри з пакету SciPy і NumPy з нативною залежністю від ATLAS LAPACK і BLAS. Реалізацію графових обчислень здійснено з використанням фреймворку Tensorflow, оскільки це забезпечує здатність до розгортання в багатьох популярних обчислювальних платформах і середовищах.

Перевірка отриманих теоретичних результатів та тестування розроблених алгоритмів було реалізовано на базі літального апарату типу квадрокоптер. Правильно зібраний і мінімально налаштований квадрокоптер здатний зависати у повітрі практично без участі пілота. Одним з найважливіших компонентів квадрокоптера є контролер польоту, що здійснює неперервну стабілізацію апарату, протидіючи поривам вітру і неоднорідностям повітряних мас. Функції зовнішньої комунікації, формування команд для контролера польоту, автономної навігації та додаткові функції аналізу даних як правило бере на себе бортовий комп'ютер.

Як бортовий комп'ютер було обрано популярний одноплатний комп'ютер Raspberry Pi 3B на базі Arm процесора, який може комплектуватися відносно недорогим польотним контролером Navio 2, або ArduPilot Mega. Крім того для реалізації функцій візуальної навігації бортовий комп'ютер може бути укомплектований додатковим одноплатним комп'ютером Raspberry Pi 3B або енергоефективним USB модулем нейрообчислень на основі Movidius Neural Compute Stick від компанії Intel. Компоновочна схема квадрокоптера, що побудованого на базі Raspberry Pi 3B, та Navio 2 показано на рис. 2.3

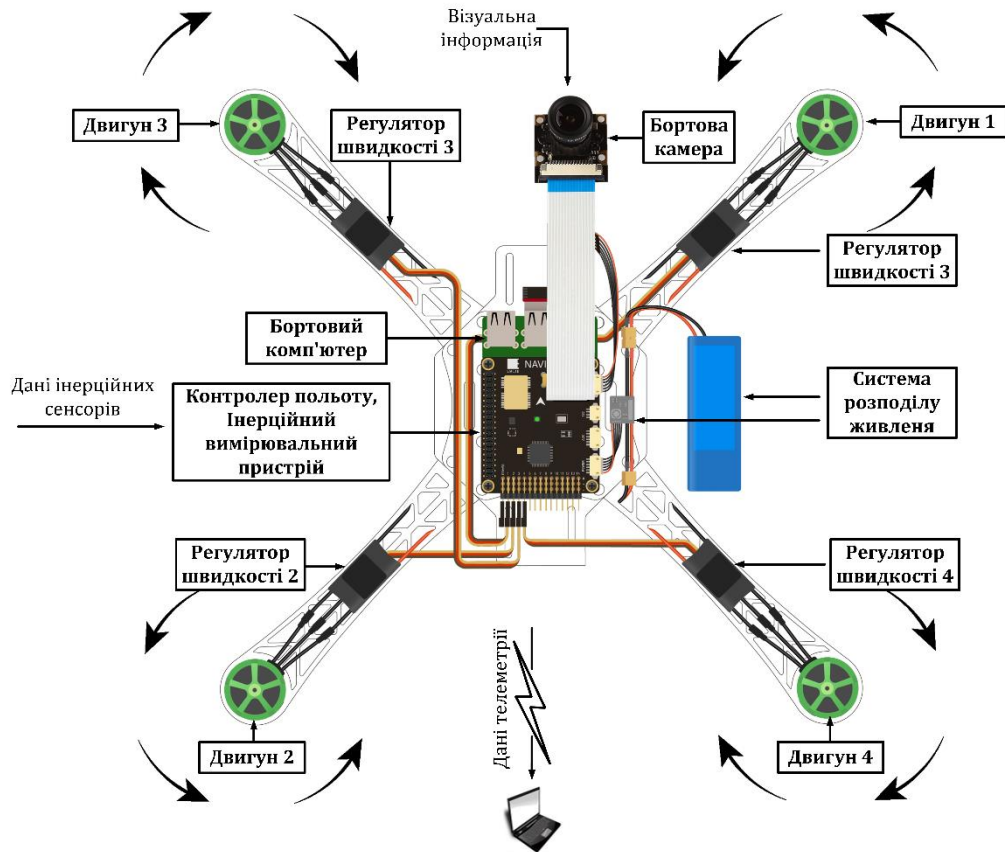


Рисунок 2.3 – Структурна схема апаратного забезпечення БПЛА

У задачах визначення поточного положення БПЛА та координат об'єктів інтересу важливим є отримання внутрішніх параметрів камери (значень оптичного центру, фокусної відстані, коефіцієнтів дисторсії) [18]. Для визначення внутрішніх параметрів необхідно виконати процедуру калібрування камери Існує велика кількість програмних продуктів призначених для калібрування камери [15]. У роботі пропонується виконати калібрування камери за допомогою програмного продукту Matlab Calibration Toolbox [18]. Під час калібрування було використано 20 зображень каліброчної таблиці формату А1 зроблених із зафіксованого положення камери. Кожний блок на каліброчній таблиці мав розміри 30 x 30 мм. В ході калібрування каліброчна таблиця встановлювалась під різними кутами до площини зображення та проводилась відеофіксація цих положень камерою, що підлягає калібруванню (рис. 2.4). Результати обчислення каліброчних параметрів наведено у табл. 2.1.

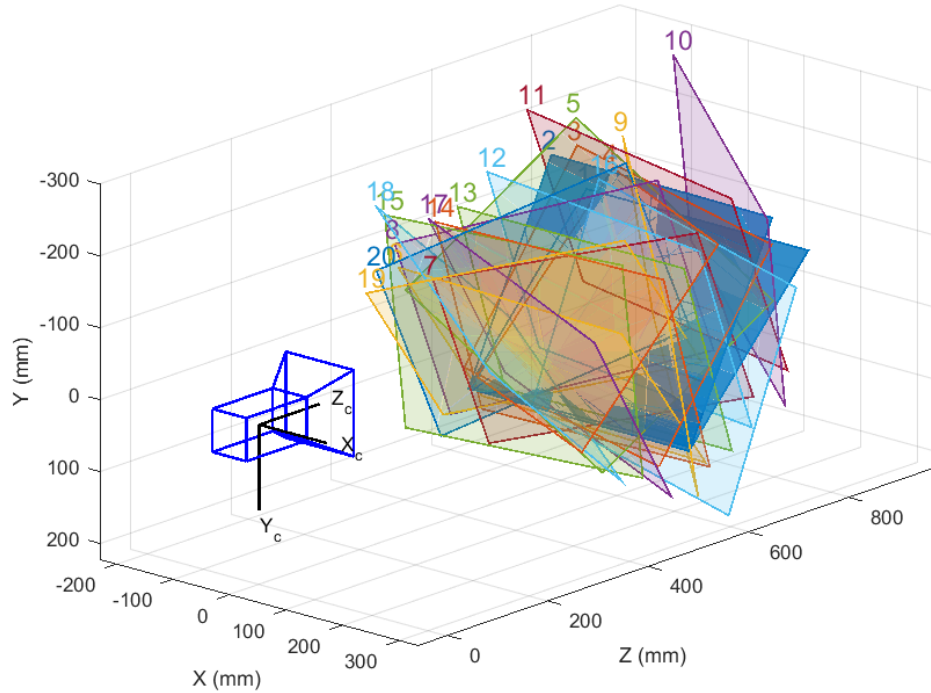


Рисунок 2.4 – Положення візуального шаблону в системі координат камери, що підлягає калібруванню

Таблиця 2.1. Внутрішні параметри калібрування камери Arducam CMOS MT9V034

Параметр	Значення параметру
Фокусна відстань f , пікс	$[461.315, 460.692] \pm [0.176, 0.176]$
Проекція оптичного центру, пікс	$[377.969, 268.412] \pm [0.300, 0.298]$
Піксельна похибка, пікс	$[0.1756, 0.1358]$
Радіальні коефіцієнти, пікс	$[-0.1036, -0.0161] \pm [0.0044, 0.0169]$
Тангенсійні коефіцієнти, пікс	$[0.0139, -0.0037] \pm [0.0243, 0.0117]$

Аналіз табл. 2.1 показує, що значення фокусної відстані подано в пікселях по горизонталі і вертикалі. Можна сказати, що сенсор даної камери має майже квадратні пікселі, оскільки співвідношення практично дорівнює одиниці. Також було з'ясовано, що реальний оптичний центр лінзи не співпадає із центром

зображення, що може вводити похибку при побудові лінії горизонту і візуальної оцінці переміщення.

Для забезпечення оптимального використання бортових ресурсів всі обчислювально складні алгоритми було реалізовано з використання векторних операцій прискорення. Такі операції полягають у об'єднанні декількох однакових операцій до однієї, при цьому набір даних до якого застосовується така операція є вектором. Дані операції підтримують роботу з наступними типами даних: integer 8, 16, 32, 64 bit, float 32 bit.

Реалізація автономної навігації передбачається на одному чи двох одноплатних 64-х бітних комп'ютерів Raspberry Pi 3+. Це комп'ютер має процесор CPU Broadcom BCM2837B0, Cortex-A53 (ARMv8) з частотою 1.4 ГГц і обсяг оперативної пам'яті рівні 1 Гбайт. Офіційна операційна система Raspbian є 32-х бітною, а неофіційні 64-х бітні операційні системи мають численні проблеми з драйверами вводу-виводу. Тому була виконана компіляція Debian Stretch для 64-х бітного процесора SoC Broadcom BCM2837 Cortex-53 chip. Проте інтелектуальний коректор візуальної одометрії ефективно розгортається з використанням USB модуля нейрообчислень Movidius Neural Compute Stick від компанії Intel. Основою даного модуля є процесор машинного зору (VPU) Intel Movidius Myriad X з інтегрованим нейропроцесором (NPU).

Враховуючи різномірний характер даних, які використовуються для аналізу для реалізації взаємодії між компонентами системи моніторингу було використано фреймворк Robotic Operation System (ROS). Застосування ROS дозволило створити гнучку, та зручну для функціонального масштабування платформу та покращити модульність системи [19]. ROS забезпечує стандартні служби операційної системи, такі як апаратну абстракцію, низькорівневий контроль пристроїв, реалізацію часто використовуваних функцій, передачу повідомлень між процесами та керування пакетами. ROS засновано на архітектурі графів де обробка інформації відбувається в вузлах (nodes), які можуть отримувати та передавати між собою повідомлення (messages) за визначеною темою (topics), а також підписатись на необхідну тему повідомлень. Враховуючи

дану концепцію можна описати структурну схему взаємодії апаратно-програмних компонентів системи аеровідеомоніторингу з використанням програмного пакету ROS у вигляді графу (рис. 2.5).

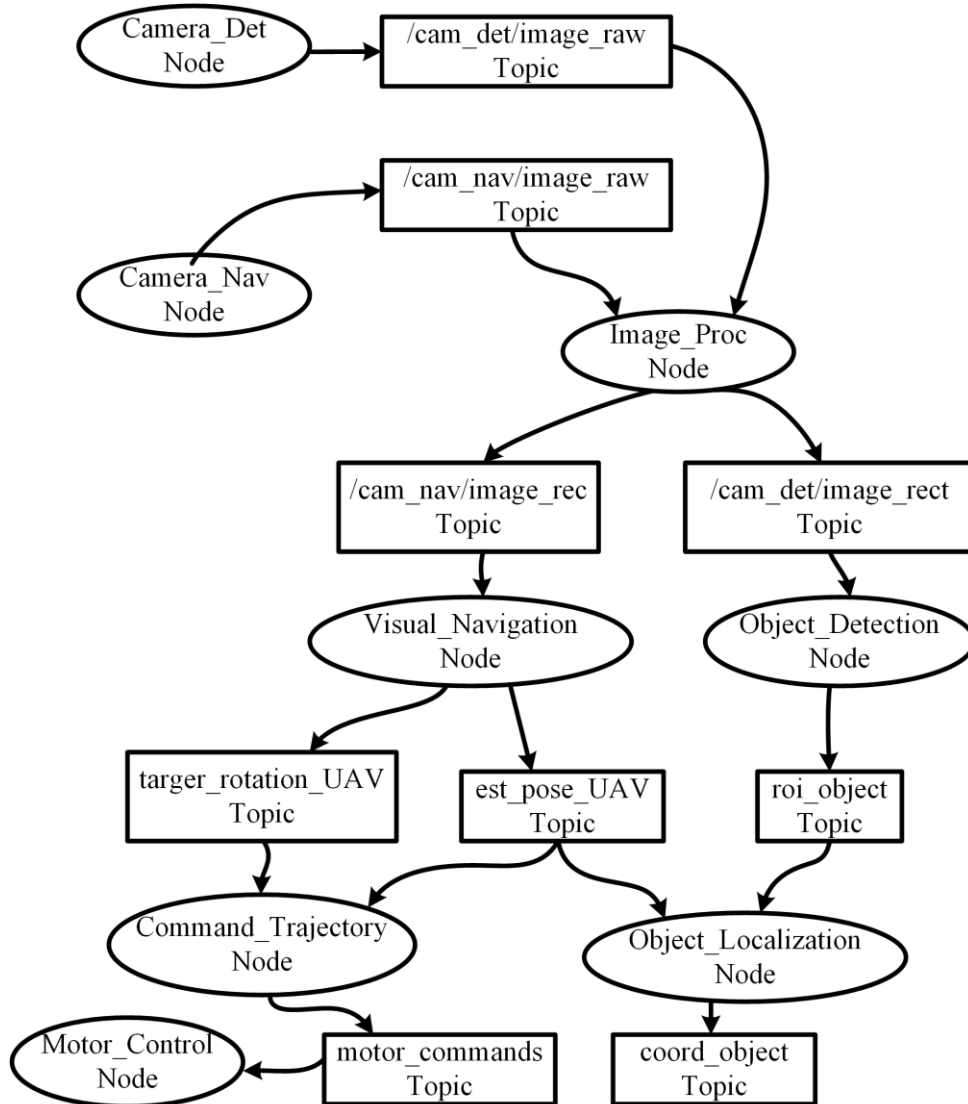


Рисунок 2.5 – Граф взаємодії компонентів системи аеромоніторингу з використання програмного пакету ROS

Відповідно до наведеної схеми на рис. 2.5 вузли Camera_Det та Camera_Nav отримують поточні візуальні дані з визначених цифрових камер. Вузол Image_Proc призначено для низькорівневої обробки зображення та зменшення ефекту дисторсії. Розрахунок поточного положення за зображенням виконується

покладено на `Visual_Navigation`. Розпізнавання об'єктів інтересу виконується у вузлі `Object_Detection`. Вузол `Object_Localization` використовуючи інформацію, отриману від повідомлень теми `roi_object` (координати розпізнаних об'єктів на зображенні) та `est_pose` (вирахована поточна позиція системи в просторі), виконує розрахунок координат об'єкта інтересу відносно БПЛА в системі координат NED. Одночасно вузол `Command_Trajectory` виконує формування команд на зміну траєкторії польоту відповідно до завдання, або для уникнення зіткнення з розпізнаними перешкодами. `Motor_Control` реалізує керування БПЛА відповідно до польотного завдання. Процес відбувається безперервно до отримання команди на припинення роботи.

Таким чином, програмно-апаратне забезпечення БПЛА містить поширені і недорогі компоненти, призначені для побудови безпілотних і робототехнічних систем. При цьому використання відкритого програмного забезпечення дозволяє здійснювати функціональне масштабування і доробку системи з мінімальними витратами ресурсів. Це забезпечує можливість швидкого прототипування, виробництва і впровадження розробки в практику.

Реалізація наземної станції керування була виконана на мові програмування Java у вигляді додатку для операційної системи Android. Архітектура програмного забезпечення станції керування відповідає сучасним принципам побудови об'єктно-орієнтованих додатків [19]. Використання шаблону Model-View-Controller (MVC), дозволило створити гнучку, та зручну для функціонального масштабування платформу. Принципи інкапсуляції використовуються при розділенні MVC-компонентів системи моніторингу на рівні інтерфейсів та пакетів, що покращує модульність програми і підтримку її частин. Графічний інтерфейс системи моніторингу місцевості показано на рис. 2.6.

На рисунку рис. 2.6а показано роботу системи в режимі ідентифікації об'єктів інтересу, де більшу частину займає візуальне відображення розпізнаних об'єктів. На рис. 2.6б відображено загальну структуру меню вибору режимів роботи системи. На рис. 2.6в показано роботу системи в режимі адаптації до розпізнавання нового алфавіту класів.

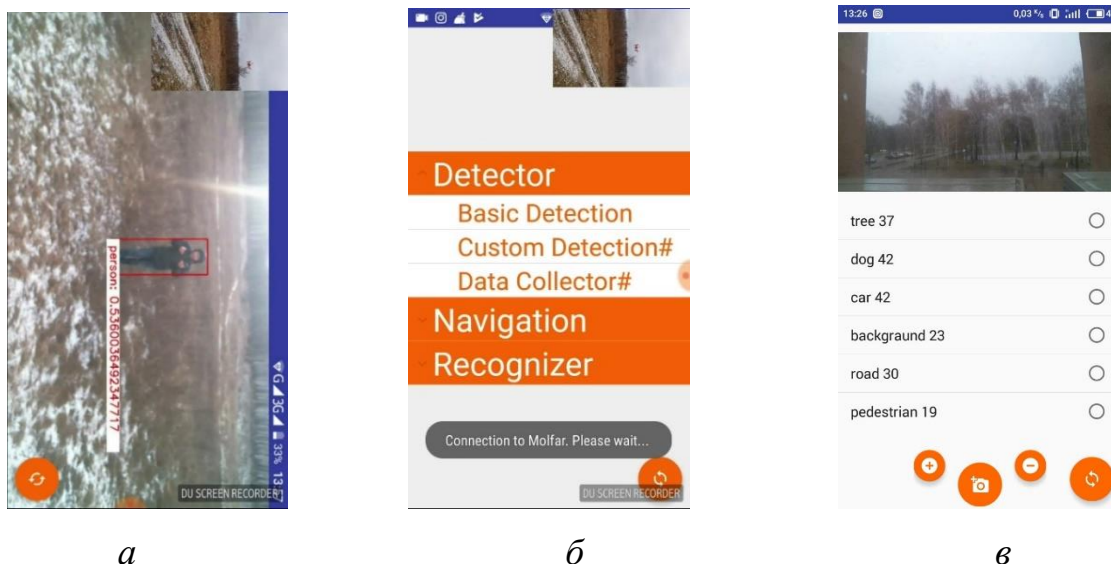


Рисунок 2.6 – Графічний інтерфейс роботи системи моніторингу :
а – в режимі розпізнавання об’єктів; *б* – головне меню вибору режимів; *в* –
 режим донавчання новим класам розпізнавання

Таким чином, для проектування апаратно-програмного забезпечення інтелектуальної бортової системи БПЛА використано бібліотеки з відритим кодом, відкриті інтерфейси і фреймворки, що полегшує розвиток і функціональне масштабування розробки.

2.2 Діаграми процесу навчання та екзамену бортової системи безпілотної апарату

У загальному випадку модель будь-якої інтелектуальної моделі аналізу даних можна подати у вигляді діаграми відображення множин, що задіяні в процесах трансформації даних та прийняття рішень (рис. 2.7).

На рис. 2.7 прийнято такі позначення:

T – множина моментів часу зняття інформації;

G – простір вхідних сигналів (факторів), які діють на систему;

Ω – простір ознак розпізнавання;

Z – простір можливих функціональних станів, у яких перебуває система;

Y – вибіркова множина (вхідна навчальна матриця);

$\Phi: G \times T \times \Omega \times Z \times V \rightarrow Y$ – оператор формування вибіркової навчальної множини Y ;

θ – оператор відображення вибіркової множини Y в простір вторинних ознак, у якому формуються вирішальні правила \mathfrak{R} ($\mathfrak{R} \subset \Omega$);

ψ – оператор прийняття рішень;

I – множина результатів перевірки статистичних тестів;

ξ_1 – оператор зворотного зв'язку для корекції параметрів вирішальних правил;

ξ_2 – оператор зворотного зв'язку для корекції параметрів формування вторинного ознакового опису;

$U: I \rightarrow G \times T \times \Omega \times Z$ – оператор регламентації процесу функціонування інтелектуальної системи аналізу вхідних даних.

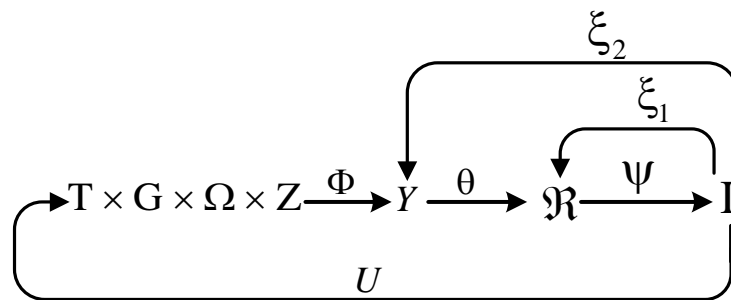


Рисунок 2.7 – Узагальнена модель інтелектуальної моделі аналізу даних у вигляді відображення множин

Для деталізації процесів оптимізації екстрактора ознак контури, утворені операторами ξ_1 та ξ_1 , розщеплюються на композицію нових контурів. На рис. 2.8 показано модель процесу навчання ієрархічного екстрактора ознак для інтелектуального аналізу спостережень у вигляді діаграми відображення множин.

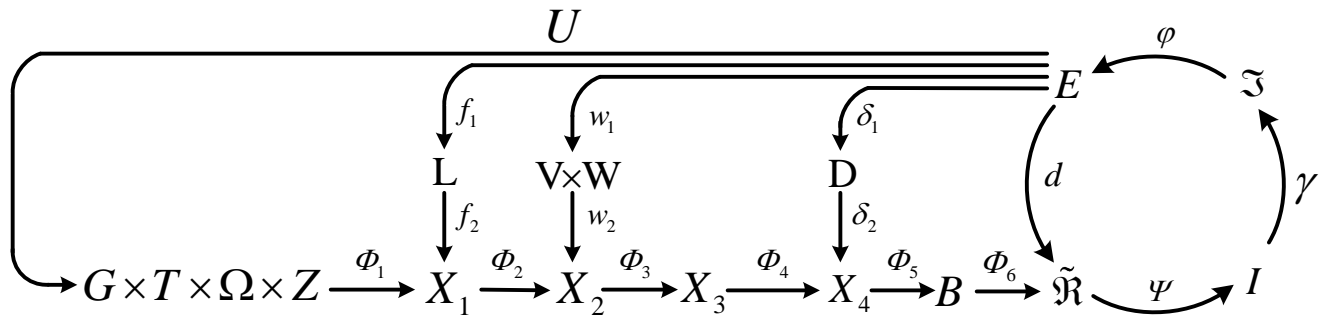


Рисунок 2.8 – Модель навчання ієрархічного екстрактора ознак для аналізу вхідних даних у вигляді діаграми відображення множин

Діаграма, показана на рис. 2.8, містить вхідний математичний опис системи аналізу вхідних даних у вигляді структури

$$\langle G, T, \Omega, Z, X_1, X_2, X_3, X_4, B; \Phi_1, \Phi_2, \Phi_3, \Phi_4, \Phi_5, \Phi_6 \rangle,$$

де X_1 – вибірка необроблених спостережень;

X_2 – спостереження, доповнені розрідженим кодом пояснюючих факторів;

X_3 – спостереження, додатково доповнені ознаковим описом, сформованим запозиченими шарами моделі згідно техніки переносу знань (Transfer Learning);

X_4 – спостереження, що описуються лише індукованими ознаками (внаслідок відкидання початкового оригінального ознакового опису);

B – спостереження, закодовані ознаковим описом найвищого рівня, в рамках якого відбувається побудова вирішальних правил;

Φ_1 – оператор формування вибірки даних;

Φ_2 – оператор екстракції пояснюючих факторів для доповнення вибірки;

Φ_3 – оператор екстракції ознакового опису на основі запозичених шарів відповідно до принципу переносу знань (Transfer Learning) для додаткового доповнення вибірки;

Φ_4 – оператор видалення початкового оригінального ознакового опису, що повертає лише індукований ознаковий опис;

Φ_5 – оператор формування ознакового опису найвищого рівня.

На рис. 2.8 оператор $\Phi_6: B \rightarrow \tilde{\mathfrak{R}}$ відображає вибірккові дані, закодовані ознаковим описом найвищого рівня, в множину параметрів, що описує конфігурацію вирішальних правил. Оператор прийняття рішень $\psi: \tilde{\mathfrak{R}} \rightarrow \mathbb{I}$ перевіряє статистичні гіпотези. Оператор $\gamma: \mathbb{I} \rightarrow \mathfrak{S} \mid q \mid$ шляхом оцінки статистичних гіпотез формує множину точнісних характеристик \mathfrak{S} . Оператор $\phi: \mathfrak{S} \rightarrow E$ обчислює множину значень критерію функціональної ефективності, який є функціоналом точнісних характеристик. Контур оптимізації геометричних параметрів вирішальних правил $\tilde{\mathfrak{R}}$ шляхом пошуку максимуму критерію навчання замикається оператором $d: E \rightarrow \tilde{\mathfrak{R}}^{|M|}$.

У процесі оптимізації параметрів і гіперпараметрів моделі беруть участь такі множини: L – множина гіперпараметрів моделі для виділення пояснюючих факторів; V – множина масок на верхні шари запозиченої нейронної мережі в рамках техніки переносу знань; W – множина значень корекції параметрів запозиченої мережі для її тонкої настройки; D – множина значень параметрів регуляризуючого шару екстракції ознакового опису найвищого рівня. Композиція операторів $\psi \circ \gamma \circ \phi \circ d$ утворює контур оптимізації геометричних параметрів вирішальних правил. Композиція операторів $\delta_1 \circ \delta_2 \circ \Phi_5 \circ \Phi_6$ утворює контур оптимізації параметрів регуляризуючого шару екстракції компактного ознакового подання найвищого рівня. Композиція операторів $w_1 \circ w_2 \circ \Phi_3 \circ \Phi_4$ утворює контур селекції та тонкої настройки шарів запозиченої нейромережі згідно принципу переносу знань. Композиція операторів $f_1 \circ f_2 \circ \Phi_2$ утворює контур оптимізації гіперпараметрів моделі виділення пояснюючих факторів.

Таким чином, формування ознакового опису полягає в застосуванні моделей і методів, що дозволяють утилізувати всю доступну апріорну інформацію для підвищення ефективності навчання. До апріорної інформації відноситься

інформація про топологію даних, інформація про доменну область застосування системи (для пошуку сторонніх моделей, акумульовані знання яких можна запозичити згідно техніки переносу), а також інформація про структуру даних у вигляді множини пояснюючих факторів. Існує велика кількість варіантів щодо вибору моделі виділення пояснюючих факторів з вибірки нерозмічених даних, велика кількість конфігурацій рецептивних полів нейронів або кодерів та способів компактного і завадозахищеного подання ознакового опису найвищого рівня. Під час синтезу моделі виділення пояснюючих факторів перевагу пропонується віддавати моделям і методам, оснований на принципах розрідженого кодування. При цьому настройка параметрів і гіперпараметрів моделі екстрактора ознак ґрунтується на максимізації функціональної ефективності вирішальних правил.

Модуль ідентифікації об'єктів інтересу та модуль корекції одометра та оцінки перешкод можуть працювати в режимі екзамену та навчання для адаптації до нових умов функціонування. Дані модулі обчислюють власний ознаковий опис спостережень для максимального врахування контексту кожної задачі. Функціональна модель запропонованої інформаційної технології автономного аеро- відео- моніторингу в режимі екзамену показано на рис. 2.9.

Розроблена інформаційна технологія дозволяє здійснювати формування ознакового опису та прийняття рішень за умов обмеженого обсягу ресурсів з врахуванням просторово-часової контекстної інформації. Для реалізації модуля ідентифікації об'єктів на місцевості результуючий ознаковий опис формується в блоці А1 шляхом об'єднання декількох карт ознак різного рівня в одну. При цьому блок 2 та блок 3 здійснюють класифікацію та локалізацію об'єкта інтересу на зображенні відповідно, де груба оцінка координат відбувається відповідно до індексу пікселя карти ознак. Для реалізації модуля корекції візуальної одометрії та оцінки перешкод блок А0 та А1 можуть бути суміщені в єдиний процес обчислення ознакового опису за серією вхідних кадрів. При цьому блок А2 та блок А3 здійснюють класифікацію перешкод на класи повороту для уникнення зіткнень та уточнення оцінки здійсненого просторового переміщення відповідно.

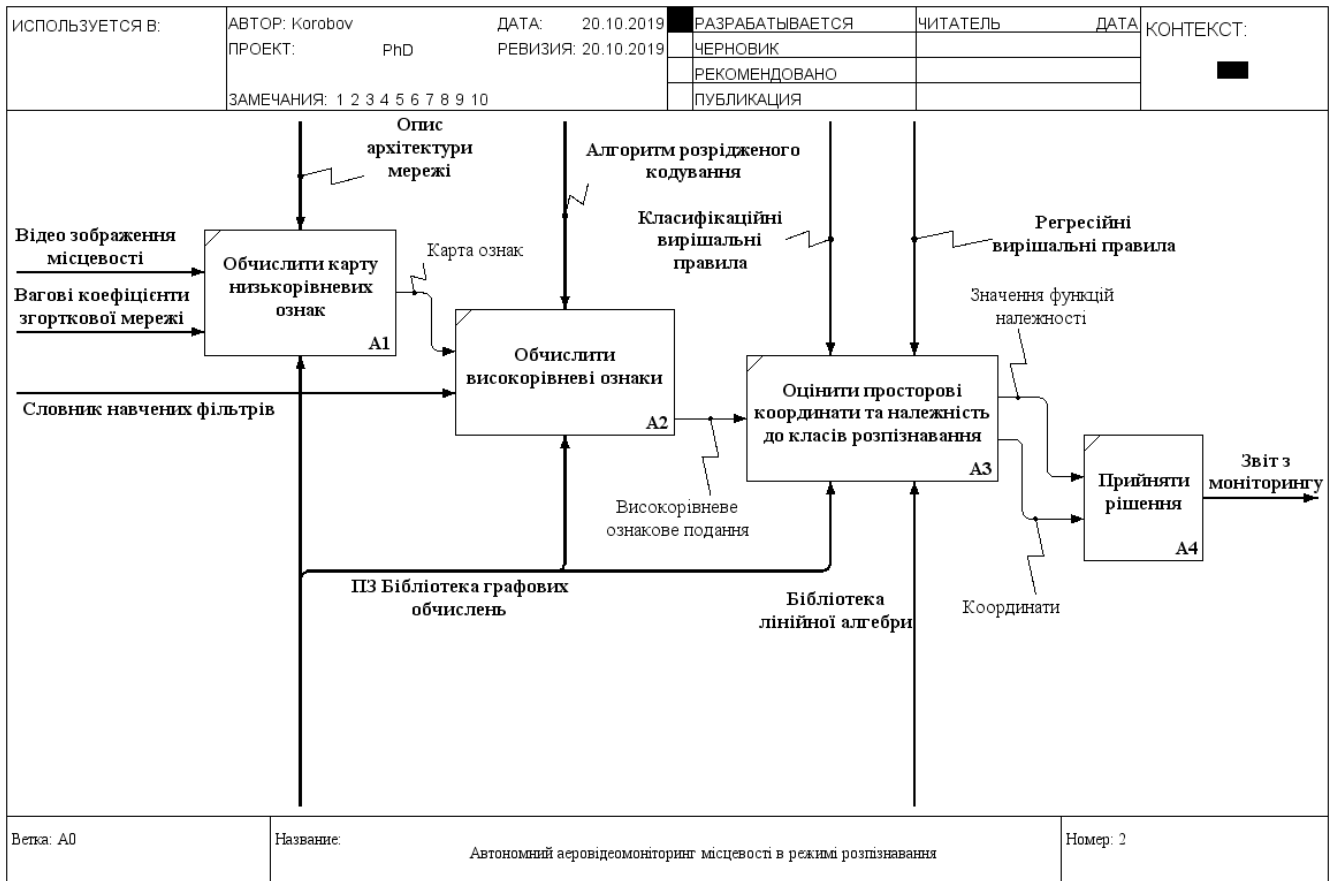


Рисунок 2.9 – Функціональна модель інформаційної технології автономного аеро-відео-моніторингу місцевості в режимі розпізнавання

Функціональну модель інформаційної технології автономного аеро-відео-моніторингу місцевості в режимі навчання показано на рис. 2.10, де відображено процеси обробки інформації згідно запропонованим у роботі протоколам машинного навчання бортової системи.

Як показано на рис. 2.10, низькорівнева карта ознак проектується на початку з використанням принципів переносу знань, а високорівневі шари екстрактора ознак після попереднього навчання можуть коректуватися в процесі максимізації комплексного критерію ефективності моделі відповідно до алгоритмів, що використовуються в блоці A3. При цьому кожне оновлення екстрактора ознак потребує процесу оновлення обчислювально ефективних вирішальних правил в блоці A2. Проте у випадку однорівневого екстрактора ознак блок A0 та A1 об'єднуються в один блок.

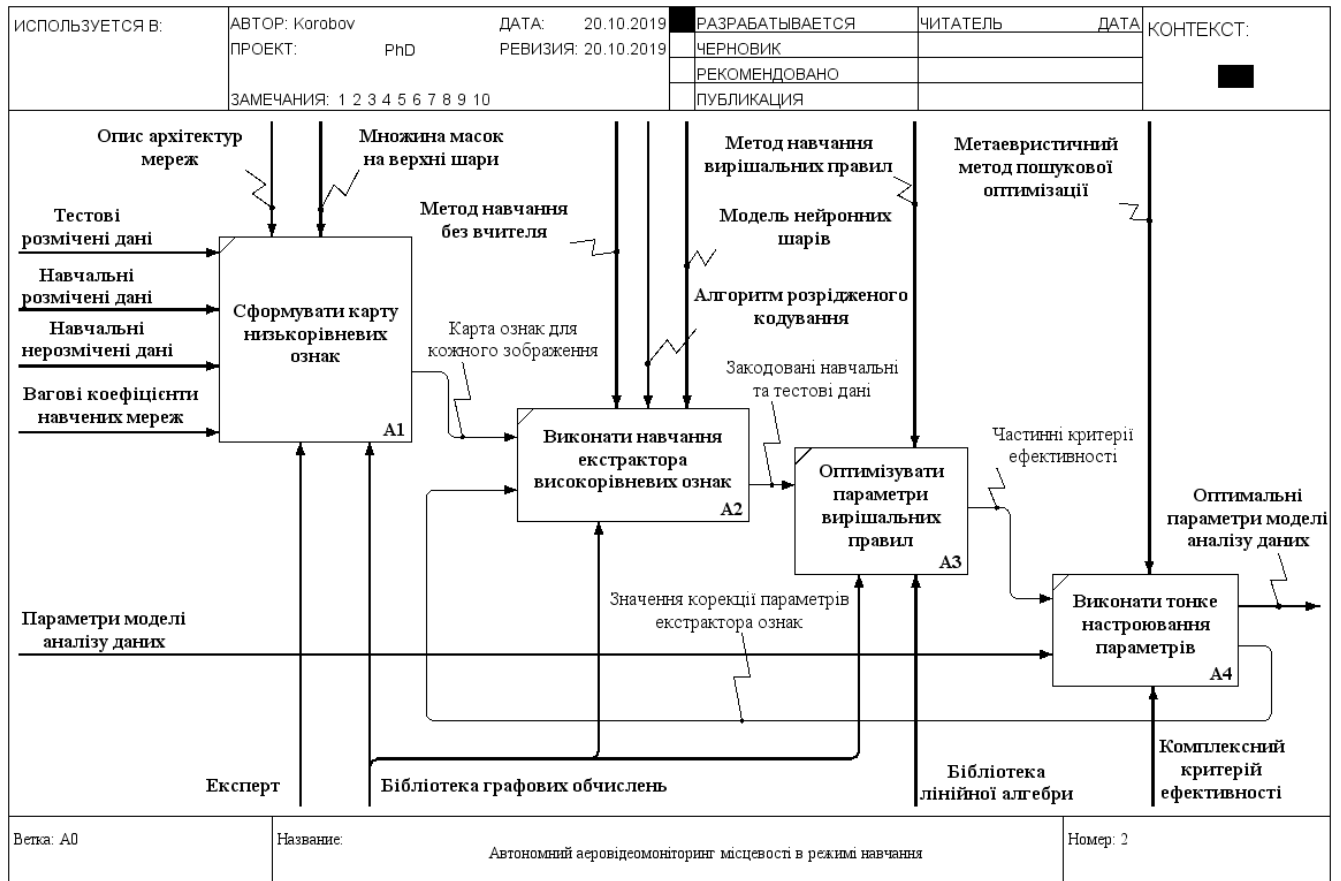


Рисунок 2.10 – Функціональна модель інформаційної технології автономного аеро- відео- моніторингу місцевості в режимі навчання

Як видно з рис. 2.10 процес оптимізації має ітераційну структуру, оскільки потребує наявності зворотного зв'язку. Тому використання обчислювально ефективних алгоритмів побудови вирішальних правил та оптимізованих бібліотек графових обчислень і лінійної алгебри визначають умови для навчання за прийнятний час без використання додаткового обладнання чи хмарних ресурсів.

Таким чином, задача інформаційного синтезу здатної навчатися автономної бортової системи ідентифікації об'єктів інтересу на місцевості полягає в оптимізації за комплексним критерієм ефективності параметрів функціонування системи, які впливають на її ефективність.

2.3 Алгоритми функціонування бортової системи в режимі класифікаційного аналізу рухомих об'єктів

У загальному випадку система детектування рухомих об'єктів, окрім режиму онлайн навчання, може мати режим офлайн навчання, під час якого відбувається узагальнення накопиченого досвіду. При цьому режим онлайн навчання та режим розпізнавання можуть слугувати механізмами збору розмічених навчальних даних для офлайн навчання глибокої моделі детектування об'єктів (рис. 2.11).

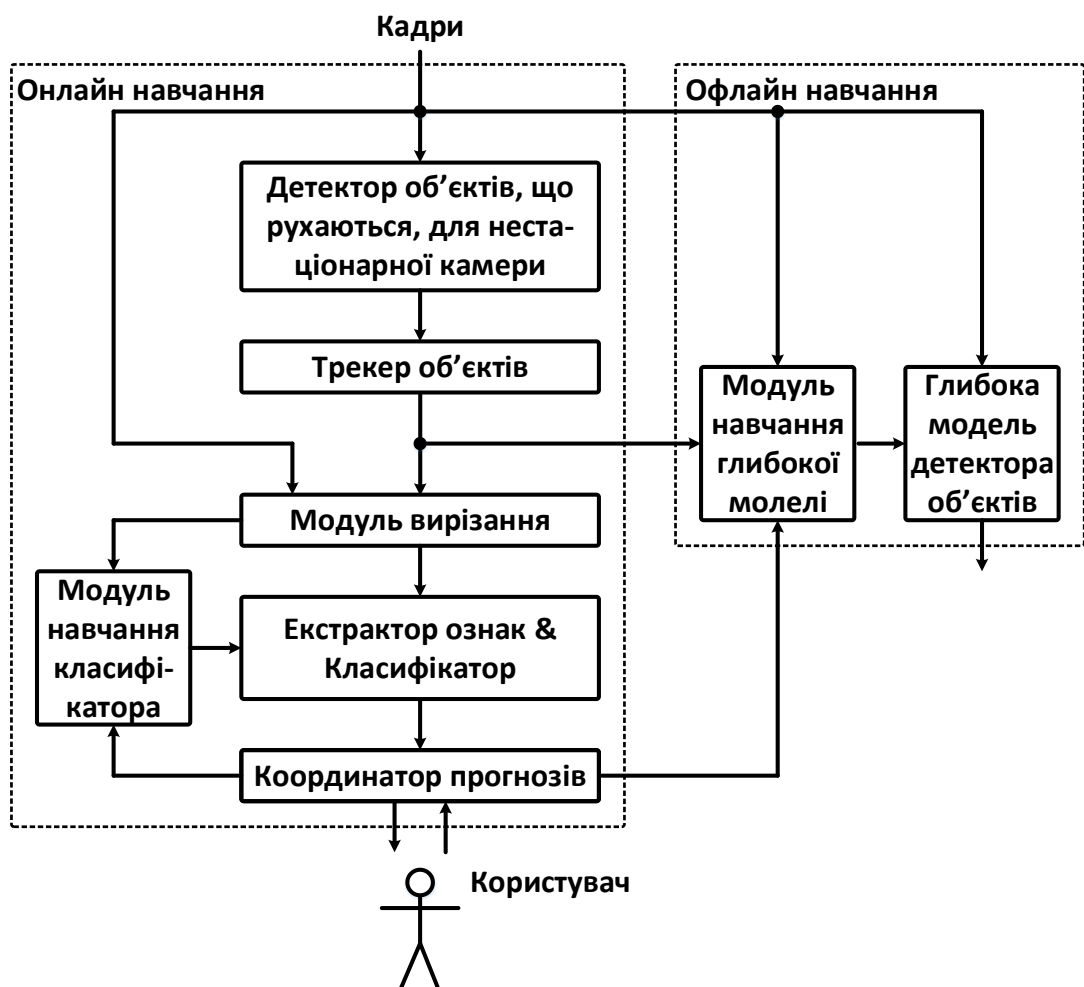


Рисунок 2.11 – Узагальнена структурна схема здатного навчатися детектора рухомих об'єктів

Як показано на рис. 2.11 важливим компонентом системи, що пропонується є детектор руху. Пропонується побудова детектора руху на основі визначення

різниці двох послідовних вирівняних кадрів. При цьому вирівнювання пропонується здійснювати на основі співставлення ключових точок, оцінки гомографії та проективної трансформації (рис. 2.12). Для підвищення оперативності обчислення дескрипторів ключових точок при збереженні їх інваріантності до повороту і масштабу можна використати алгоритм SURF замість SIFT [14].

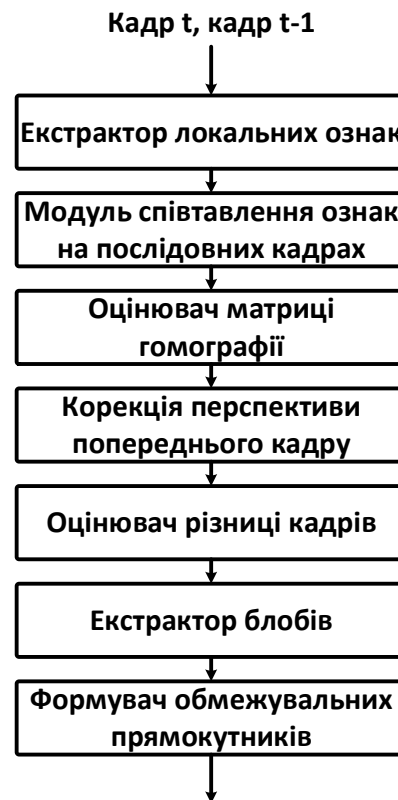


Рисунок 2.12 – Функціональна схема детектора руху

Різницеве зображення, утворене після віднімання вирівняних кадрів, містить шум. Фільтрацію найпростіше виконати на основі швидких морфологічних операцій, таких як ерозія (Erode) та дилатація (Dilate) [15]. Ядра морфологічних фільтрів обираються з огляду на мінімальний розмір об'єктів інтересу на зображенні.

Трекер детектованого об'єкта дозволяє продовжити збір навчальних даних навіть при зупинці руху об'єкта інтересу відносно камери. Як показано в праці [16] найбільш ефективними з точки зору стабільності і обчислювальної складності є алгоритм Predator, ще відомий як TLD (Tracking-Learning-Detection). Вихід

детектора може бути поданий для просторового пулінгу в глибоку мережу детектування. Тоді детектор руху і трекер виконуватимуть функцію мережі RPN (Region proposal network). Однак для невеликого алфавіту класів і невеликої кількості об'єктів інтересу в області спостереження цілком достатньо виконати вирізання і уніфікацію розміру зображень області інтересу для їх подальшого класифікаційного аналізу. В режимі онлайн навчання можна усунути з розгляду кадри, що містять забагато об'єктів інтересу для підвищення оперативності функціонування.

Важливим компонентом запропонованого детектора є екстрактор ознак, який пропонується реалізувати у вигляді тришарової згорткової нейронної мережі з субдискретизуючим шаром типу max-pooling між ними. Вхідна роздільна здатність згорткової мережі обирається рівною 64x64. Ядра фільтрів встановлюються рівними 3x3. Перший шар містить 64 фільтри, а другий і третій – по 32 фільтри.

Зони інтересу кодуються згортковою нейронною мережею і надходять на вхід класифікатора. Інформаційно-екстремальний класифікатор здійснює двійкове кодування вхідних спостережень, використовуючи дерева рішень. Прийняття рішень відбувається на основі радіально-базисних вирішальних правил, що розбивають двійковий простір на області класів (рис. 1.13).

Навчання моделі детектора пропонується здійснювати в декілька етапів.

- попереднє навчання без вчителя згорткового екстрактора ознак на великому наборі даних ImageNet64;
- початкове навчання класифікатора за малою вибіркою об'єктів інтересу;
- донавчання класифікатора з використанням техніки самонавчання та активного навчання;
- навчання глибокої моделі детектора на зібраних розмічених навчальних зразках.

Для реалізації першого етапу пропонується використати модифікацію нейронного газу, описану в працях [9]. Цей алгоритм збігається близько до глобального оптимуму завдяки м'якій конкуруючій схемі навчання. При цьому

завдяки використанню правила Ойа глобальний оптимум відповідає такому словнику фільтрів, що дозволяє найкраще реконструювати вхідні дані [12].

Під час роботи системи накопичуються нові розмічені навчальні зразки. Вони з'являються в результаті стратегії активного навчання : початковий зразок з двійковим поданням b надходить на розмітку користувачу, якщо виконується умова $\max_z \mu_z(b) > \mu_{th1}$, де μ_{th1} – порогове значення функції належності, значення нижче якого інтерпретуються як невпевнене розпізнавання; z – номер класу об'єкта інтересу. При цьому навчальний зразок з двійковим поданням b з міткою $z^* = \arg \max_z \mu_z(b)$ додається в навчальну вибірку з цією міткою, якщо $\mu_z > \mu_{th2}$, де μ_{th2} – порогове значення функції належності, перевищення якого є ознакою впевненості розпізнавання. В обох випадках після отримання ручної чи автоматичної мітки, ця мітка зберігатиметься для всіх зразків, що формуватимуться під час трекінгу розпізнаних об'єктів. Зразки з максимальним значенням функції належності з діапазону $[\mu_{th1}, \mu_{th2}]$ ігноруються.

Класифікатор після першого навчання на малій вибірці залишає 80% дерев фіксованими. Решта дерев будуються заново при поповненні навчальної вибірки.

2.4 Результати фізичного моделювання

Для початкового навчання класифікатора сформовано алфавіт з 7-ми класів : X_1^o – велосипед (bicycle); X_2^o – автобус (bus); X_3^o – легковий автомобіль (car), X_4^o – мотоцикл (motorcycle), X_5^o – пікап (pickup truck), X_6^o – вантажівка (articulated truck), X_7^o – фон. Для цього було використано відкриті набори даних MIO-TCD dataset [20].



Рисунок 2.13 – Приклади зразків з набору даних MIO-TCD

У тестовій і навчальній вибірці мітиться 100 зразків на клас. Навчальні зразки приводяться до уніфікованої роздільної здатності 64×64 пікселя. Залежність нормованого значення інформаційного критерію (1.6) за навчальною вибіркою та точність за тестовою вибіркою від кількості дерев для двійкового кодування показано на рис. 2.14. При цьому глибина дерев рішень була рівна 5, а глибина стекінгу рівна 3.

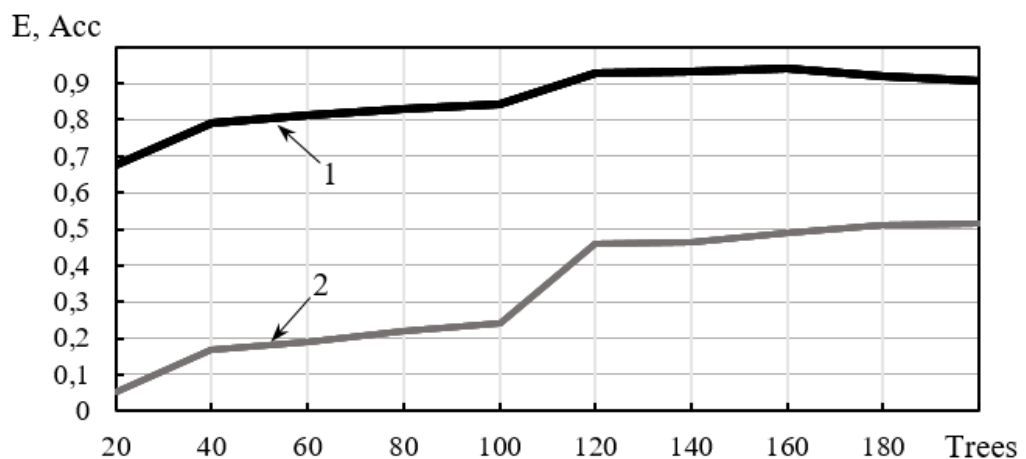


Рисунок 2.14 – Графік залежності нормованого інформаційного критерію (1.6) за навчальною вибіркою та точність за тестовою вибіркою від кількості дерев рішень для двійкового кодування

Аналіз рис. 5 показує, що після побудови 160 дерев точність класифікатора починає зменшуватися внаслідок перенавчання. Тому обирається оптимальна кількість дерев, що забезпечує точність 0,94%. Подальше збільшення точності може здійснюватися за рахунок збільшення кількості інформативних навчальних даних. Збір даних здійснюватиметься в інтерактивному режимі з відеофайлів набору даних VIRAT Video Dataset [20].

На рис. 2.15 показано залежність нормованого значення інформаційного критерію (1.6) за навчальною вибіркою та точність за тестовою вибіркою від кількості зразків, які були розмічені користувачем в режимі активного навчання зі значення параметра $\mu_{ml} = 0.1$. Самонавчання при цьому не застосовувалося.

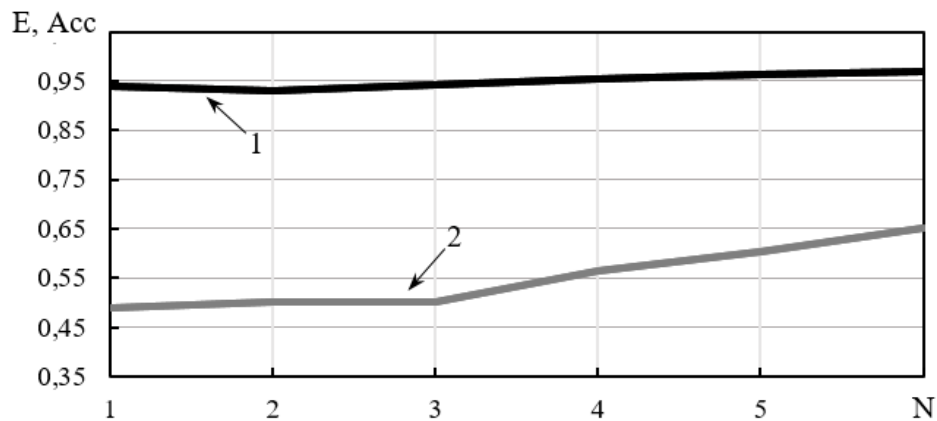


Рисунок 2.15 – Графік залежності нормованого інформаційного критерію (1.6) за навчальною вибіркою та точність за тестовою вибіркою від кількості зразків, які були розмічені користувачем в режимі активного навчання

Аналіз рис. 2.15 показує, що розмітка і додавання зразків, які розпізнаються з низькою впевненістю, до навчальної вибірки дозволяє підвищити точність класифікатора на тестовому наборі даних. При цьому було додано всього 6 зразків, проте вдалося підвищити точність до 97%. Тобто додатково 15 зразків тестової вибірки стали розпізнаватися правильно.

Для підвищення репрезентативності навчальних даних додаватимемо зразки, що виникають під час трекінгу об'єктів, розмічених користувачем та тих, що розпізнаються з високим рівнем впевненості. На рис. 2.16 показано залежності

нормованого інформаційного критерію (1.6) за навчальною вибіркою та точність за тестовою вибіркою від кількості зразків, що додаються за результатами трекінгу. При цьому встановлено таке порогове значення $\mu_{th2} = 0,7$.

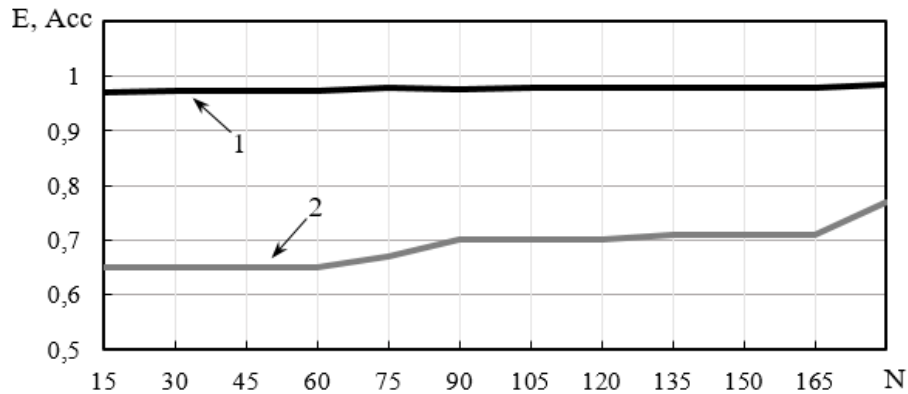


Рисунок 2.16 – Графік залежності нормованого інформаційного критерію (1.6) за навчальною вибіркою та точність за тестовою вибіркою від кількості зразків, які були розмічені користувачем в режимі активного навчання

Аналіз рис. 2.16 показує, що збільшення варіації навчальних зразків, отриманих внаслідок трекінгу дозволив додатково підвищити точність вирішальних правил навіть без втручання користувача. Було додано 178 навчальних зразків і отримано точність за тестовою вибіркою до 98.5%.

Таким чином, запропоновані алгоритми дозволяють зменшити трудомісткість розмітки навчальних даних та підвищити репрезентативність вибірок і точність вирішальних правил. При цьому сформовані під час функціонування кадри зображень з розміченими обмежувальними прямокутниками можуть бути використані для навчання глибокого детектора об'єктів в режимі офлайн.

ВИСНОВКИ

У науково-дослідній роботі розв'язано важливу науково-технічну прикладну задачу аналізу і синтезу здатної навчатися автономної бортової системи БЛА для ідентифікації об'єктів на місцевості. Головні наукові та практичні результати роботи полягають у такому:

– вперше розроблено процесну діаграму та діаграму відображення множин для опису процесу синтезу вирішальних правил, що полегшує аналіз та реінжиніринг алгоритмів функціонування бортової системи в режимі навчання;

– вперше розроблено процесну діаграму та діаграму відображення множин для опису процесу прийняття рішень бортовою системою, що полегшує аналіз та реінжиніринг алгоритмів функціонування бортової системи в режимі екзамену;

– вперше розроблено метод навчання бортової системи безпілотного апарату на основі моделей з інкрементальним ускладненням, що дозволяє формувати оптимальні в інформаційному та вартісному сенсі регресійні та класифікаційні вирішальні правила;

– удосконалено алгоритм локалізації і трекінгу рухомих об'єктів шляхом використання оцінки гомографії за сусідніми кадрами, що дозволяє зберігати працездатність за умов рухомої камери;

– удосконалено алгоритм формування навчальних даних для глибокої моделі детектора об'єктів шляхом поєднання технік активного навчання, самонавчання та методів детектування і трекінгу рухомих об'єктів, що дозволяє зменшити складність і вартість розмітки навчальних даних;

– отримали подальшого розвитку методи екстракції візуальних ознак, що дозволяють формувати інформативний ознаковий опис за умов обмеженого обсягу розмічених навчальних даних за рахунок поєднання техніки розділення пояснючих факторів за нерозміченими даними та техніки переносу знань;

Практична значимість отриманих результатів полягає в зниженні вимог до обсягу обчислювальних ресурсів і навчальних даних для задач детектування об'єктів на місцевості. Крім того запропоновані алгоритми дозволяють знизити

трудомісткість формування розміченого навчального набору даних для донавчання глибоких моделей детектування об'єктів на зображенні.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Москаленко В.В. “Моделі і методи інтелектуального аналізу багатовимірних даних да умов апріорної невизначеності”: монографія / В. В. Москаленко. – Суми: Видавництво СумДУ, 2019. – 185 с.
2. Свідоцтво про реєстрацію авторського права на твір комп’ютерна програма «Інформаційно-екстремальний класифікатор багатовимірних спостережень з метаевристичною оптимізацією параметрів функціонування» №89337 / В.В. Москаленко, А.С. Москаленко, М.О. Зарецький, А.Г. Коробов, В.В. Нагорний // дата реєстр. 05.06.2019.
3. Свідоцтво про реєстрацію авторського права на твір комп’ютерна програма «Стекований інформаційно-екстремальний класифікатор» № 93727 / Москаленко В.В., Москаленко А.С., Зарецький М.О // дата реєстр. 06.11.2019.
4. Moskalenko V. V. A Model And Training Method Of Small-Sized Object Detection System For A Compact Aerial Drone // V. V. Moskalenko, A. S. Moskalenko, A. G. Korobov, M. O. Zaretsky / Radio Electronics, Computer Science, Control. – 2019. – № 1. (WoS)
5. V. Moskalenko, A. Moskalenko, A. Korobov, V. Semashko. The Model and Training Algorithm of Compact Drone Autonomous Visual Navigation System. – Data. – 2019. – Vol. 4(1). (Scopus, WoS)
6. Moskalenko V.V. Model and training method of moving object classification system for a compact drone / V.V. Moskalenko, M.O. Zaretskyi, A.S. Moskalenko // Radioelectronic and Computer Systems. – № 1. – 2019. – P. 108–117.
7. Moskalenko V. V. Model and training method of moving object classification system for a compact drone / V. V. Moskalenko, A. S. Moskalenko, M. O. Zaretsky // 2019 IEEE 14th International Conference on Computer Sciences and Information Technologies (CSIT) . – 2019. (Scopus & WoS).
8. Moskalenko A. Operative Recognition of Standard Signals in the Presence of Interference with Unknown Characteristics A. Moskalenko, V. Avramenko // The

Second International Workshop on Computer Modeling and Intelligent Systems. – Zaporizhzhia, Ukraine, April 15-19. – 2019. – p. 56–70 (Scopus).

9. В. В. Москаленко. Модель і метод навчання детектора малорозмірних об'єктів за умов ресурсних та інформаційних обмежень / В. В. Москаленко, М. О. Зарецький, Д. О. Лєвокутник // XIV International scientific conference «Intellectual systems of decision-making and problems of computational intelligence». – с. Залізний Порт, 21-25 трав. 2019 р. – Херсон: Видавництво ФОП Вишемирський В. С., 2019. – С. 116-118.

10. Москаленко В.В. Модель та алгоритм навчання системи автономної навігації для малогабаритного безпілотного апарату // В.В. Москаленко, А.С. Москаленко, А.Г. Коробов, М.О. Зарецький. – Біоніка інтелекту. – №2 (91). – 2018. – С. 16–24.

11. Forbus K. D. Representation and Computation in Cognitive Models / K. D. Forbus, C. Liang, I. Rabkina // Topic Continuation: Visions of Cognitive Science: Edited by Wayne D. Gray – Game XP: Action Games as Experimental Paradigms for Cognitive Science. – 2017. –Vol. 9. – I. 2. – P. 255–536.

12. Chen X. Privileged Semi-Supervised Learning / X. Chen, C. Gong, C. Ma, X. Huang et al // 25th IEEE International Conference on Image Processing (ICIP). – Athens, Greece, 2018. – P. 2999–3003.

13. Patel H. Dictionary Properties for Sparse Representation: Implementation and Analysis / H. Patel, H. Mewada // Journal of Artificial Intelligence. – 2018. – Vol. 11. – I. 1. – P. 1–8.

14. Hadsell R. Dimensionality Reduction by Learning an Invariant Mapping / R. Hadsell, S. Chopra, Y. LeCun // Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). – New York, NY, USA, 2006. – pp. 1735–1742.

15. Chang H. Stacked Predictive Sparse Decomposition for Classification of Histology Sections / H. Chang, Y. Zhou, A. Borowsky, K. Barner, et al // International Journal of Computer Vision. – 2015. –Vol. 113. –P. 3–18.

16. Labusch K. Sparse coding neural gas: learning of overcomplete data representations / K. Labusch, E. Barth, T. Martinetz // *Neurocomputing*. – 2009. – Vol. 72. – I. 7–9. – P. 1547–1555.

17. Palomo J. The Growing Hierarchical Neural Gas Self-Organizing Neural Network / J. Palomo, E. López-Rubio // *IEEE Transactions on Neural Networks and Learning Systems*. – 2017. – Vol. 28, No. 9. – P. 2000–2009.

18. Moskalenko V. Improving the effectiveness of training the on-board object detection system for a compact unmanned aerial vehicle / V. Moskalenko, S. Dovbysh, I. Naumenko, A. Moskalenko et al // *Eastern-European Journal of Enterprise Technologies*. – 2018. – No 4/9(94). – P. 19–26.

19. Tim S. C. W. / Multi-layer Dictionary Learning for Image Classification / S. C. W. Tim, M. Rombaut, D. Pellerin // *Proceedings of the 17th International Conference on Advanced Concepts for Intelligent Vision Systems*. – Lecce, Italy, Oct. 24–27. – 2016. – P. 522–533.

20. Vens C. Random Forest Based Feature Induction. / C. Vens, F. Costa // *Proceedings of the IEEE 11th International Conference on Data Mining*. – Vancouver Canada, 11–14 Dec, 2011. Piscataway, NJ, 2011. – P. 744–753.