Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

Кіберфізична система управління квадрокоптерами для розпізнавання об'єктів за допомогою машинного навчання

Назва теми

Рівень вищої освіти *другий (магістерський)*

Галузь знань *12 «Інформаційні технології»*
Шифр, назва

Спеціальність *123 «Комп'ютерна інженерія»*
Шифр, назва

Освітня програма *«Комп'ютерна інженерія та програмування»*
Назва

Шифр КвРКІ 240109.24.01.34.ПЗ

Виконав здобувач II курсу, група КІ2м-24-2 Підпис Ілля ГРІЦАЙ
Ініціали, прізвище

Керівник д.-техн. наук, професор Євген ФЕДОРОВ
Науковий ступінь, учене звання Підпис Ініціали, прізвище

Нормоконтролер д. техн. наук, професор Сергій ЛИСЕНКО
Науковий ступінь, учене звання Підпис Ініціали, прізвище

До захисту допускаю:
завідувач кафедри КПС Ольга ПАВЛОВА
«01» травня 2026 р. Підпис Ініціали, прізвище
дата

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Комп'ютерної інженерії та інформаційних систем

Рівень вищої освіти другий (магістерський)

Галузь знань 12 Інформаційні технології

Спеціальність 123 Комп'ютерна інженерія

Освітня програма «Комп'ютерна інженерія та програмування»

ЗАТВЕРДЖУЮ
Завідувач кафедри КІС

Ольга ПАВЛОВА

" 12 " 01 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Гріцай Ілля Сергійович
Прізвище, ім'я, по батькові студента

1.Тема проєкту (роботи): Кіберфізична система управління квадрокоптерами для розпізнавання об'єктів за допомогою машинного навчання

Керівник проєкту (роботи): Федоров Євген Євгенович, д.т.н., проф.
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 12.01.2026 р. № 6

2. Термін подання здобувачем роботи на кафедру 01.05.2026 р.

3. Вихідні дані до роботи Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Аналіз відомих моделей, методів та засобів управління квадрокоптерами для розпізнавання об'єктів за допомогою машинного навчання;

Розроблення моделей та методів для вирішення задач управління квадрокоптерами для розпізнавання об'єктів за допомогою машинного навчання;

Проєктування кіберфізичної системи управління квадрокоптерами для розпізнавання об'єктів за допомогою машинного навчання;

Прототипування, розробка, опис і тестування розроблених модулів кіберфізичної системи управління квадрокоптерами для розпізнавання об'єктів за допомогою машинного навчання.

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

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «12» 01 2026 р.

КАЛЕНДАРНИЙ ПЛАН

ч	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проекту (роботи)	Примітка
1	Проаналізувати літературу за темою роботи та аналіз існуючих систем	12.01.2026	виконано
2	Проаналізувати існуючі технології машинного навчання	12.01.2026	виконано
3	Проаналізувати відомих моделей, методів та засобів управління квадрокоптерами для розпізнавання об'єктів за допомогою машинного навчання	20.01.2026	виконано
4	Розробити алгоритми інтеграції системи з відкритими базами	01.02.2026	виконано
5	Робота над науковою статтею	01.03.2026	виконано
6	Провести навчання налаштування і навчання нейронної мережі	15.03.2026	виконано
7	Провести тестування розробленої системи	01.04.206	виконано
8	Оформлення пояснювальної записки згідно вимог	18.04.2026	виконано
9	Попередній захист ВКР	29.04.2026	виконано
10	Захист ВКР на засіданні ЕК	До 15.05.2026	виконано

Здобувач _____ Ілля ГРІЦАЙ
Підпис Ім'я, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи _____ Євген ФЕДОРОВ
Підпис Ім'я, ПРІЗВИЩЕ

РЕФЕРАТ

Тема кваліфікаційної роботи магістра: “Кіберфізична система управління квадрокоптерами для розпізнавання об'єктів за допомогою машинного навчання”.

Автор роботи: Гріцай Ілля Сергійович.

Керівник роботи: Федоров Євген Євгенович.

Пояснювальна записка: 83 стор., 29 рис., 1 табл., 3 дод., 85 джерел.

КІБЕРФІЗИЧНА СИСТЕМА, КВАДРОКОПТЕР, БПЛА, МАШИННЕ НАВЧАННЯ, КОМП’ЮТЕРНИЙ ЗІР, НЕЙРОННА МЕРЕЖА, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, ГЛИБОКЕ НАВЧАННЯ, АВТОНОМНЕ КЕРУВАННЯ, КОМП’ЮТЕРНЕ БАЧЕННЯ.

Об’єктом дослідження є процес автоматизованого управління квадрокоптерами на основі аналізу відеоданих та методів машинного навчання.

Предметом дослідження є методи та програмні засоби розпізнавання об’єктів у кіберфізичних системах управління безпілотними літальними апаратами.

Метою кваліфікаційної роботи магістра є розробка кіберфізичної системи управління квадрокоптерами для розпізнавання об’єктів за допомогою технологій машинного навчання та комп’ютерного зору.

Методи дослідження: для розв’язання поставлених задач у роботі використано методи системного аналізу, проєктування кіберфізичних систем, методи збору та обробки даних, а також елементи математичного моделювання. Для підвищення якості обробки інформації застосовано підходи машинного навчання, а експериментальна частина базувалася на методах прототипування та тестування апаратно-програмних комплексів.

Наукова новизна отриманих результатів полягає в удосконаленні підходу до інтеграції модулів комп’ютерного зору у структуру кіберфізичної системи управління квадрокоптерами шляхом використання адаптованої архітектури нейронної мережі для розпізнавання об’єктів на зображеннях аерозйомки в умовах обмежених обчислювальних ресурсів.

Практичне значення отриманих результатів полягає у створенні програмного комплексу, який може бути використаний для:

– автоматизованого моніторингу територій;

– задач аеророзвідки;

– аналізу відеопотоку з БПЛА;

– підтримки прийняття рішень;

– побудови інтелектуальних систем автономного керування квадрокоптерами.

У першому розділі виконано аналіз сучасних кіберфізичних систем управління квадрокоптерами та методів розпізнавання об’єктів за допомогою технологій машинного навчання. Проведено дослідження існуючих моделей комп’ютерного зору, архітектур нейронних мереж та програмно-апаратних платформ для реалізації систем автоматизованого аналізу відеоданих. Розглянуто особливості використання БПЛА для задач аеророзвідки та моніторингу, виконано аналіз методів формування датасетів і сучасних алгоритмів детекції об’єктів. На основі проведеного аналізу сформульовано постановку задачі дослідження та визначено основні вимоги до розроблюваної системи.

У другому розділі досліджено моделі та методи функціонування кіберфізичної системи управління квадрокоптерами. Розроблено математичну модель процесу розпізнавання об’єктів на кадрах аерозйомки та сформовано структурно-функціональну архітектуру системи. Проведено аналіз алгоритмів комп’ютерного зору та моделей глибокого навчання, обґрунтовано вибір архітектури нейронної мережі для задач класифікації об’єктів у режимі реального часу. Також розглянуто підходи до оптимізації моделі та інтеграції програмних модулів у структуру системи управління БПЛА.

У третьому розділі розроблено програмне та інформаційне забезпечення кіберфізичної системи. Побудовано функціональні моделі системи у нотації *IDEF*0, спроєктовано концептуальну, логічну та фізичну моделі бази даних, реалізовано структуру збереження результатів класифікації та параметрів об’єктів. Розроблено користувацький інтерфейс програмного модуля, алгоритми попередньої обробки

зображень, механізми формування датасетів та процедури навчання нейронної мережі. Виконано підбір гіперпараметрів моделі та оцінювання якості навчання із використанням методів перехресної перевірки.

У четвертому розділі представлено результати прототипування, програмної реалізації та тестування розроблених модулів кіберфізичної системи управління квадрокоптерами. Реалізовано програмний модуль розпізнавання об’єктів на основі згорткової нейронної мережі, проведено навчання моделі та оцінювання її ефективності за допомогою сучасних метрик машинного навчання. Представлено результати аналізу точності класифікації, матриці помилок, графіків навчання та тестування системи на нових даних. Отримані результати підтвердили працездатність і ефективність запропонованого підходу для задач автоматизованого аналізу відеоданих із БПЛА.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

CNN – Convolutional Neural Network

MCC – Matthews Correlation Coefficient

AUC – Area Under the Curve

PR – Precision-Recall

DS – dataset

ReLU – Rectified Linear Unit

ГНМ – глибинна нейронна мережа

ШНМ – штучна нейронна мережа

ВСТУП

Сучасний розвиток безпілотних літальних апаратів та систем штучного інтелекту призвів до активного впровадження кіберфізичних систем у сфери моніторингу, навігації, автоматизованого керування та аналізу навколишнього середовища. Особливого значення такі системи набувають у задачах аеророзвідки, спостереження за територіями, супроводу об’єктів та підтримки прийняття рішень у реальному часі. В умовах постійного зростання обсягів відеоданих, отриманих із бортових сенсорів БПЛА, виникає необхідність автоматизації процесів виявлення та розпізнавання об’єктів без постійного втручання оператора.

Одним із найбільш перспективних напрямів розвитку кіберфізичних систем є інтеграція методів комп’ютерного зору та машинного навчання у структуру автономного керування квадрокоптерами. Використання нейронних мереж дозволяє автоматично аналізувати відеопотік, виявляти об’єкти, визначати їх характеристики та формувати рішення для подальшого управління безпілотним літальним апаратом. При цьому особливого значення набуває забезпечення високої швидкості обробки даних та стабільності роботи моделей в умовах обмежених обчислювальних ресурсів бортових платформ.

Актуальність теми дослідження обумовлена необхідністю створення інтелектуальних кіберфізичних систем, здатних забезпечувати автоматизований аналіз зображень та відеоданих у режимі реального часу. Існуючі системи розпізнавання об’єктів часто потребують значних обчислювальних ресурсів, мають складність інтеграції з автономними системами керування або демонструють недостатню ефективність при роботі зі зображеннями аерозйомки, що містять шуми, спотворення та складні умови освітлення [1-5]. Тому розробка ефективної кіберфізичної системи управління квадрокоптерами із використанням сучасних методів машинного навчання є актуальною науково-прикладною задачею.

Об’єктом дослідження є процес автоматизованого управління квадрокоптерами на основі аналізу відеоданих та методів машинного навчання.

Предметом дослідження є методи та програмні засоби розпізнавання об’єктів у кіберфізичних системах управління безпілотними літальними апаратами.

Метою роботи є розробка кіберфізичної системи управління квадрокоптерами для розпізнавання об’єктів за допомогою технологій машинного навчання та комп’ютерного зору.

Для досягнення поставленої мети у роботі необхідно вирішити такі задачі:

– провести аналіз сучасних методів та систем розпізнавання об’єктів у БПЛА;

– дослідити підходи до побудови кіберфізичних систем управління квадрокоптерами;

– виконати аналіз моделей машинного навчання та алгоритмів комп’ютерного зору;

– розробити функціональну та інформаційну архітектуру системи;

– реалізувати програмний модуль розпізнавання об’єктів;

– сформувати та підготувати навчальний датасет;

– виконати навчання та налаштування нейронної мережі;

– провести тестування та оцінювання ефективності роботи системи.

У роботі використано такі методи дослідження:

– методи системного аналізу;

– методи машинного навчання;

– методи комп’ютерного зору;

– згорткові нейронні мережі;

– методи цифрової обробки зображень;

– методи математичного моделювання;

– методи тестування програмного забезпечення.

Наукова новизна отриманих результатів полягає в удосконаленні підходу до інтеграції модулів комп’ютерного зору у структуру кіберфізичної системи управління квадрокоптерами шляхом використання адаптованої архітектури нейронної мережі для розпізнавання об’єктів на зображеннях аерозйомки в умовах обмежених обчислювальних ресурсів.

Практичне значення отриманих результатів полягає у створенні програмного комплексу, який може бути використаний для:

– автоматизованого моніторингу територій;

– задач аеророзвідки;

– аналізу відеопотоку з БПЛА;

– підтримки прийняття рішень;

– побудови інтелектуальних систем автономного керування квадрокоптерами.

Результати роботи можуть бути використані у сфері безпілотних авіаційних систем, систем моніторингу, робототехніки, комп’ютерного зору та вбудованих систем штучного інтелекту.

1 АНАЛІЗ ВІДОМИХ МОДЕЛЕЙ, МЕТОДІВ ТА ЗАСОБІВ УПРАВЛІННЯ КВАДРОКОПТЕРАМИ ДЛЯ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ЗА ДОПОМОГОЮ МАШИННОГО НАВЧАННЯ

1.1 Аналіз предметної області і виявлення наявних проблем і завдань

Сучасний етап розвитку автономних систем характеризується переходом до концепції кіберфізичних систем, які інтегрують обчислювальні потужності, мережеву взаємодію та фізичні процеси [1, 6-9]. На відміну від традиційних інформаційних систем, основний виклик при проектуванні CPS полягає в їхній безпосередній взаємодії з фізичним світом, де програмне забезпечення має реагувати на зміни середовища в режимі реального часу.

Безпілотні літальні апарати (БПЛА) є типовим прикладом складних кіберфізичних систем, де вбудоване програмне забезпечення керує двигунами, навігацією та безпекою.

Більшість комп'ютерів, що керують фізичними процесами, є вбудованими системами з лімітованою обчислювальною потужністю. Для БПЛА це створює потребу у використанні архітектур нейронних мереж, що мінімізують навантаження на процесор [10-14]. Сучасні рішення, такі як YOLO26, вирішують цю проблему шляхом впровадження NMS-free інференсу (відмови від енергозатратного придушення не-максимумів), що критично важливо для стабільної роботи на крайових пристроях або при передачі даних для швидкої обробки.

У динамічному середовищі швидкість детекції об'єктів є критичною. Традиційні двостадійні детектори є занадто повільними, тому безальтернативним підходом є використання моделей класу YOLO, які розглядають детекцію як єдину задачу регресії [15-22]. Наприклад, уже версія YOLOv3 демонструвала швидкість 22 мс на зображення, що у 3,8 раза швидше за аналоги типу RetinaNet при подібній точності. Це дозволяє підтримувати високу частоту кадрів, необхідну для запобігання аваріям у реальному часі.

Необхідно забезпечити надійний зв'язок між модулем комп'ютерного зору та польотним контролером. Для цього використовуються спеціалізовані протоколи, такі як MAVLink. У сучасних системах на базі PX4 для передачі завдань автопілоту найчастіше використовується режим Offboard, який дозволяє зовнішній системі керувати траєкторією через повідомлення uORB або міст uXRCE-DDS.

Оскільки обрана архітектура передбачає передачу відеопотоку на сервер для обробки з наступним поверненням команд автопілоту, виникає проблема стабільності каналу зв'язку. Будь-яка втрата пакетів або затримка в мережі може критично вплинути на безпеку польоту, що вимагає розробки алгоритмів безпеки у стеку PX4.

Використання відкритого ПЗ, такого як Ultralytics YOLO та PX4 Autopilot, дозволяє гнучко адаптувати систему під різні типи завдань – від простої детекції до складного автономного стеження та сегментації.

В той же час еволюція методів детекції об'єктів демонструє постійне прагнення до балансу між точністю та швидкістю обробки – від традиційних методів комп'ютерного зору до сучасних моделей глибокого навчання. Традиційні двостадійні детектори забезпечували високу точність, проте через розділення процесу на пропонування регіонів та їх подальшу класифікацію вони залишаються надто повільними для систем, що працюють у реальному часі.

Проривом стала поява сімейства моделей YOLO, розроблених Джозефом Редмоном та Алі Фархаді. Унікальний підхід YOLO полягає в тому, що він розглядає детекцію як єдину задачу регресії, прогнозуючи обмежувальні рамки та ймовірності класів безпосередньо з повних зображень за один прохід нейронної мережі [21-23]. Це робить YOLO значно швидшим за попередні моделі при збереженні високої точності.

Вже у 2016 році YOLOv2 впровадив нормалізацію пакетів та анкерні бокси. Випущений у 2018 році YOLOv3 продемонстрував значну перевагу над конкурентами: він працював утричі швидше за SSD та у 3,8 раза швидше за RetinaNet при аналогічних показниках точності.

Кожна версія додавала інновації, такі як аугментація даних Mosaic v4, оптимізація гіперпараметрів v5 та спеціалізація для автономних роботів доставки v6. Важливим етапом стала YOLOv10, яка впровадила End-to-End голову, що усунула потребу в придушенні не-максимумів.

Версії YOLOv8–YOLO11 стали мультизадачними платформами, що підтримують не лише детекцію, а й сегментацію, класифікацію, оцінку пози та відстеження об'єктів. YOLO11, випущена у вересні 2024 року, забезпечує високу продуктивність у широкому спектрі AI-додатків, включаючи підтримку орієнтованих обмежувальних рамок та мультиоб'єктного трекінгу.

YOLO26 найсучасніша версія, спеціально оптимізована для розгортання на крайових пристроях Головною перевагою є NMS-free інференс, що усуває обчислювальне вузьке місце на стадії постобробки. Це дозволяє системі працювати стабільно навіть на слабких процесорах або в умовах хмарних API, де мінімізація затримок є критичною.

Використання YOLO26 у клієнт-серверній архітектурі БПЛА дозволяє отримати переваги швидкої серверної обробки без затримок, характерних для застарілих алгоритмів постобробки, забезпечуючи при цьому універсальність у виконанні різних візуальних завдань.

1.3 Аналіз апаратних платформ для вбудованого штучного інтелекту

Для реалізації інтелектуальної автономності БПЛА в межах кіберфізичних систем критично важливим є вибір обчислювального модуля, який забезпечить баланс між продуктивністю інференсу нейромережі та енергоспоживанням бортової мережі [27-30]. Основними критеріями вибору є швидкість обробки кадрів, підтримка апаратної квантизації та інтеграція з польотним стеком.

Огляд сучасних платформ БПЛА:

1. NVIDIA Jetson Nano –базується на архітектурі Maxwell з 128 ядрами CUDA, що робить її найбільш гнучким рішенням для екосистеми Ultralytics YOLO:

– завдяки підтримці обчислень з плаваючою комою, Jetson Nano здатна забезпечувати стабільний інференс для моделей класу YOLOv8n/YOLO11n на рівні 15–25 FPS;

– використання найновішої моделі YOLO26 з NMS-free інференсом дозволяє додатково знизити навантаження на CPU, усуваючи затримки постобробки;

– підтримує 8-бітну квантизацію (INT8) через бібліотеку TensorRT, що дозволяє подвоїти пропускну здатність при незначній втраті точності;

– енергоспоживання варіюється від 5W до 10W, що є прийнятним для середніх квадрокоптерів, але вимагає ефективного охолодження.

2. Raspberry Pi 4 та Raspberry Pi 5 – одноплатні комп'ютери широко використовуються в спільноті PX4 через низьку вартість та наявність готових образів:

– позбавлена виділеного графічного прискорювача для нейромереж, що на моделях YOLOv8n/YOLO11n RPi 4 демонструє низьку швидкість, тоді як RPi 5 за рахунок потужнішого CPU може досягати 5–8 FPS (недостатньо для динамічного маневрування, але придатне для статичного моніторингу);

– використовує переважно 8-бітну квантизацію через TensorFlow Lite або OpenVINO для прискорення обчислень на CPU, проте приріст FPS залишається обмеженим архітектурою;

– енергоспоживання на рівні 5–12W, тому платформа потребує використання супровідних плат інтеграції для надійного живлення.

3. Прискорювачі Google Coral – спеціалізовані тензорні процесори, що підключаються через USB або M.2 інтерфейси:

– демонструє найвищу ефективність саме для моделей, оптимізованих під Edge (при використанні квантованих моделей YOLO11n, Coral здатний забезпечувати понад 30 FPS, що робить його ідеальним для завдань відстеження об'єктів у реальному часі;

– вимагає обов'язкової 8-бітної квантизації (моделі, не конвертовані в цей формат, не зможуть використовувати переваги TPU);

– енергоспоживання дуже низьке, що дозволяє інтегрувати його навіть у малі БПЛА без суттєвого впливу на час польоту.

Результати порівняння платформ наведено в табл. 1.1.

Таблиця 1.1 – Порівняльні характеристики платформ

Параметр	NVIDIA Jetson Nano	Raspberry Pi 5	Google Coral
Архітектура AI	GPU	CPU	TPU
Оптимальна модель	YOLO11/26	YOLO11n	YOLO11n
FPS	15–25	5–8	30
Квантизація	INT8/FP16	INT8	INT8
Споживання	5–10W	5–12W	2–4W
Зв'язок з FC	UART/MAVLink	UART/uXRCE-DDS	Companion PC

Для задач, що потребують високої швидкості обробки в реальному часі при мінімальному споживанні, найбільш ефективним є використання прискорювачів Google Coral у зв'язці з Raspberry Pi. Однак для систем, що потребують універсальності та роботи з різними типами нейромереж без жорсткої прив'язки до INT8 квантизації, кращим вибором залишається серія NVIDIA Jetson.

1.4 Програмні середовища та протоколи даних у кіберфізичних систем

Проєктування сучасних кіберфізичних систем для управління БПЛА потребує створення стійкої програмної архітектури, здатної інтегрувати високопродуктивні обчислення, надійні канали зв'язку та механізми керування фізичними об'єктами в реальному часі [31-33]. Основним викликом при розробці таких систем є забезпечення детермінованості передачі даних та мінімізація

затримок у контурі прийняття рішень, особливо при використанні ресурсомістких алгоритмів штучного інтелекту для розпізнавання об'єктів.

Для дослідження ключовим інструментом є ROS 2, в основі якого лежить архітектура Data Distribution Service (DDS) – стандарт проміжного програмного забезпечення, що забезпечує децентралізовану передачу даних за принципом «публікація-підписка».

Використання DDS дозволяє системі динамічно виявляти вузли мережі без необхідності в центральному майстер-сервері, що підвищує відмовостійкість кіберфізичної системи [34-36]. Для БПЛА це означає можливість паралельної роботи модулів навігації, комп'ютерного зору на базі моделей YOLO та систем моніторингу телеметрії, де кожен модуль може мати власні параметри якості обслуговування (Quality of Service, QoS). Налаштування QoS у ROS 2 дозволяє пріоритезувати критично важливі повідомлення, такі як команди аварійної зупинки або траєкторії польоту, забезпечуючи їх гарантовану доставку навіть у зашумлених мережевих умовах.

У архітектурі керування квадрокоптерами фундаментальну роль відіграє протокол MAVLink. Це легковажний протокол обміну повідомленнями, спеціально розроблений для передачі даних між польотним контролером, бортовим комп'ютером та наземною станцією керування. MAVLink характеризується високою ефективністю за рахунок малого розміру заголовків пакетів, що мінімізує накладні витрати на смугу пропускання каналу зв'язку.

Основним завданням MAVLink у даному дослідженні є трансляція телеметрії та статусів транспортного засобу до зовнішнього сервера обробки. Протокол підтримує понад 250 типів повідомлень і широко використовується в екосистемі PX4 Autopilot для конфігурації параметрів польоту та отримання зворотного зв'язку від автопілота в реальному часі.

Одним із найважливіших технологічних рішень для сучасних кіберфізичних систем є впровадження моста Micro eXtremely Resource Constrained Environments DDS (uXRCE-DDS). Цей протокол дозволяє інтегрувати вбудований польотний стек PX4 безпосередньо з екосистемою ROS 2. Технічно uXRCE-DDS забезпечує

трансляцію внутрішніх повідомлень автопілота (uORB topics) у топіки ROS 2 і навпаки без значних обчислювальних витрат на боці мікроконтролера.

У контексті розпізнавання об'єктів за допомогою нейромереж YOLOv11 або YOLO26, використання uXRCE-DDS надає такі переваги як зменшення затримки, час синхронізації і режим offboard.

Пряме відображення повідомлень uORB у ROS 2 усуває необхідність у важких протоколах-посередниках, що дозволяє модулю ШІ швидше отримувати одометрію та передавати назад команди керування [37-40].

Завдяки режиму Offboard мосту зовнішній сервер обробки може публікувати повідомлення, які PX4 інтерпретує як цільові точки маршруту в режимі реального часу, забезпечуючи плавне автономне стеження за ціллю, а зв'язка ROS 2 та PX4 Autopilot є оптимальною архітектурою для створення стабільного циклу управління в кіберфізичній системі.

1.5 Специфіка фомування та обробки датасетів для БПЛА

Ефективність функціонування кіберфізичних систем на базі БПЛА безпосередньо залежить від якості та специфіки вхідних даних, що використовуються для навчання моделей глибокого навчання. На відміну від традиційних завдань комп'ютерного зору, розпізнавання об'єктів з борту квадрокоптера має низку унікальних особливостей, зумовлених фізичними процесами зйомки та динамікою польоту.

Однією з основних проблем при розробці систем детекції для БПЛА є невідповідність стандартних наборів даних, таких як MS COCO або Pascal VOC, реальним умовам експлуатації апаратів. У класичних датасетах об'єкти зазвичай зняті з горизонтального ракурсу, тоді як камера БПЛА частіше за все працює в режимі ракурсу зверху. Це призводить до суттєвої зміни візуальних ознак об'єктів: наприклад, автомобіль з висоти виглядає як прямокутник, що нівелює ефективність ознак, вивчених на фронтальних зображеннях.

Також виникає критична проблема масштабу. Об'єкти на знімках БПЛА часто мають надзвичайно малий розмір – від 10 до 50 пікселів. Для стандартних архітектур нейромереж такі об'єкти стають «невидимими» через втрату просторової інформації на глибоких шарах згортки. Це вимагає використання спеціалізованих архітектур, таких як YOLO11 або YOLO26, які мають оптимізовані механізми виділення ознак для дрібних об'єктів.

Для подолання обмежень стандартних вибірок у науковій спільноті використовують спеціалізовані датасети:

VisDrone, один із наймасштабніших наборів даних, зібраний командою AISKYEYE. Він містить тисячі зображень та відеокадрів, знятих у різних локаціях за різних умов освітлення. VisDrone включає складні сценарії з високою щільністю об'єктів, що є критичним для перевірки стійкості моделей SOTA.

Unmanned Aerial Vehicle Benchmark, фокусується на детекції та відстеженні транспортних засобів. Він пропонує анотації для різних сценаріїв польоту та кутів нахилу камери, що дозволяє моделювати роботу кіберфізичну систему у динамічних міських середовищах.

Для підвищення точності розпізнавання дрібних цілей у сучасних ітераціях YOLO застосовуються специфічні методи обробки даних. Одним із найефективніших є Mosaic Augmentation, впроваджений ще у YOLOv4 і вдосконалений у версіях YOLOv8/v11. Суть методу полягає у змішуванні чотирьох навчальних зображень в одне, що змушує модель розпізнавати об'єкти в меншому масштабі та в нетипових контекстах, значно покращуючи здатність детекції на периферії.

Для вирішення проблеми низької роздільної здатності об'єктів при інференсі використовується підхід Slicing Aided Hyper Inference або Tiler. Ця техніка передбачає розбиття вхідного кадру високої роздільної здатності на дрібні фрагменти з певним перекриттям. Кожен фрагмент обробляється нейромережею окремо, після чого результати об'єднуються. Це дозволяє виявляти мікро-об'єкти, які при звичайному стисненні кадру до розміру 640x640 пікселів просто зникли.

Формування реальних датасетів для БПЛА часто обмежене законодавчими нормами, погодними умовами або ризиком пошкодження апарату. У межах кіберфізичного підходу важливу роль відіграють симулятори, такі як Gazebo та AirSim. Вони дозволяють генерувати практично необмежену кількість синтетичних даних з ідеальною розміткою. Симуляція в Gazebo інтегрована безпосередньо в польотний стек PX4, що дає змогу тестувати не лише точність ШІ, а й реакцію автопілота на виявлені об'єкти в безпечному віртуальному середовищі перед льотними випробуваннями.

1.6 Класифікація БПЛА за конструктивними особливостями та сферами застосування

Класифікація безпілотних авіаційних систем (БАС) здійснюється за низкою ключових ознак, серед яких основними є аеродинамічна схема, масо-габаритні характеристики та функціональне призначення.

Літакового типу, конструкція базується на використанні нерухомого крила, що створює підйомну силу за рахунок швидкості набігаючого повітряного потоку. Такі апарати характеризуються високою енергоефективністю, тривалим часом польоту та здатністю охоплювати значні території, проте потребують спеціально обладнаних злітних смуг або систем запуску і не здатні до зависання.

Мультироторні системи, апарати з декількома несучими гвинтами: трикоптери, квадрокоптери, гексакоптери, октокоптери. Основним об'єктом дослідження є квадрокоптери, де керування польотом реалізується через зміну тяги та крутного моменту чотирьох роторів. Дана схема забезпечує вертикальний зліт і посадку, високу маневреність та стабільне зависання у визначеній точці простору.

Гібридні системи, поєднують у собі конструктивні особливості літакового і мультироторних типів [19]. Вони використовують ротори для вертикального зльоту та посадки, але виконують основну частину польоту в режимі круїзу за допомогою крила, що дозволяє поєднувати здатність до зависання з високою дальністю польоту.

Приклади мультироторних систем представлені на рисунку 1.1.

Рисунок 1.1 – Вигляд бікоптеру і трикоптеру

Згідно з міжнародними стандартами, БПЛА за злітною вагою поділяються на такі категорії:

1. Клас I: включає мікро-БПЛА, міні-БПЛА та малі апарати. Прикладом є БПЛА «Лелека-100» зі злітною масою близько 5 кг.

2. Клас II: системи масою від 150 до 600 кг. До цього класу можна віднести апарати типу PD-2 з максимальною злітною вагою 55 кг.

3. Клас III: апарати масою понад 600 кг, зокрема середньовисотні великої тривалості польоту та висотні великої тривалості польоту. До категорії належить апарат Zephyr вагою 75 кг при розмаху крил 25 м, здатний перебувати у повітрі понад 60 діб.

Для завдань автоматизованого розпізнавання об'єктів квадрокоптери мають перевагу перед літаковими БПЛА завдяки здатності до зависання. Це дозволяє оптико-електронним системам тривалий час фокусуватися на об’єкті, мінімізуючи розмиття зображення та забезпечуючи високу роздільну здатність для роботи нейронних мереж. Маневреність у обмеженому просторі робить квадрокоптери незамінними при інспекції об’єктів інфраструктури, де літакові БПЛА не можуть працювати через високу швидкість польоту та обмежені радіуси розвороту.

Функція розпізнавання об’єктів за допомогою машинного навчання є критично важливою у таких сферах:

– пошуково-рятувальні операції, використання згорткових нейронних мереж дозволяє в реальному часі виявляти людей, фрагменти одягу або транспортні засоби у важкодоступній місцевості [35, 37];

– моніторинг інфраструктури, БПЛА забезпечують візуальну детекцію тріщин, корозії або механічних пошкоджень трубопроводів та ліній електропередач. Алгоритми глибинного навчання автоматизують процес ідентифікації дефектів, виключаючи людський фактор [38, 39];

– точне землеробство, дрони використовуються для аналізу стану посівів через розрахунок вегетаційних індексів. Штучний інтелект критично важливий для диференційованого розпізнавання типів бур’янів, шкідників та хвороб рослин, що дозволяє точково вносити пестициди та добрива [40].

1.7 Постановка задачі

Об'єктом дослідження є система управління квадрокоптерами для розпізнавання об'єктів.

Предметом дослідження є методи машинного навчання, що використовуються в кіберфізичної системі управління квадрокоптерами для розпізнавання об'єктів

Метою даної роботи є підвищення ефективності кіберфізичної системи управління квадрокоптерами для розпізнавання об'єктів за допомогою методів машинного навчання.

Для досягнення поставленої мети необхідно вирішити такі задачі:

1. Формування та анотування спеціалізованого набору даних:

– підготовка та розмітка наборів зображень, що відповідають специфічним умовам експлуатації БПЛА;

– анотування у форматі, сумісному з екосистемою Ultralytics, для подальшого навчання моделей детекції, сегментації або відстеження об'єктів.

2. Навчання та оптимізація нейромережевої моделі:

– вибір та тренування нейронної моделі, яка дозволяє усунути затримки на етапі постобробки, що критично важливо для обробки відеопотоку в реальному часі на сервері або крайових пристроях;

– оптимізація моделі для досягнення мінімальної затримки при збереженні високої точності розпізнавання.

3. Комплексне тестування та верифікація, що передбачає перевірку системи в симуляційному середовищі, перевірку логіки безпеки, конфігурацію захисту від втрати зв'язку з сервером та відпрацювання алгоритмів автономного прийняття рішень у різних сценаріях.

1.8 Висновки до розділу 1

У першому розділі було проведено аналіз сучасних моделей, методів та засобів управління квадрокоптерами для розпізнавання об’єктів за допомогою технологій машинного навчання. Розглянуто особливості побудови кіберфізичних систем, що поєднують програмні компоненти, засоби обробки даних, сенсорні модулі та виконавчі механізми безпілотних літальних апаратів. Встановлено, що сучасні БПЛА активно використовуються для задач моніторингу, аеророзвідки, навігації та автоматизованого виявлення об’єктів у режимі реального часу.

У результаті аналізу предметної області визначено основні проблеми існуючих систем, серед яких можна виділити високі вимоги до обчислювальних ресурсів, залежність точності розпізнавання від якості датасетів, складність забезпечення стабільної роботи алгоритмів у реальних умовах польоту та необхідність швидкої обробки відеоданих у режимі реального часу. Також встановлено, що ефективність систем комп’ютерного зору значною мірою залежить від правильного вибору архітектури нейронної мережі, способу підготовки даних та апаратної платформи.

Проведено порівняльний аналіз існуючих підходів до розпізнавання об’єктів на зображеннях і визначено доцільність використання згорткових нейронних

мереж для задач детекції та класифікації об’єктів аеророзвідки. Встановлено, що сучасні моделі сімейства YOLO забезпечують високу швидкодію та достатню точність для інтеграції у кіберфізичні системи управління БПЛА.

Окрему увагу приділено аналізу апаратних платформ для реалізації вбудованого штучного інтелекту. Визначено, що використання спеціалізованих одноплатних комп’ютерів та енергоефективних обчислювальних модулів дозволяє реалізувати алгоритми комп’ютерного зору безпосередньо на борту квадрокоптера, що зменшує затримки передачі даних та підвищує автономність системи.

У межах розділу також було розглянуто програмні середовища, бібліотеки та протоколи передачі даних, які використовуються у кіберфізичних системах БПЛА. Встановлено доцільність використання ROS 2, MAVLink та сучасних бібліотек машинного навчання для реалізації програмного комплексу управління та розпізнавання об’єктів.

Проведений аналіз підходів до формування та обробки датасетів показав, що якість навчальної вибірки безпосередньо впливає на результати роботи моделей комп’ютерного зору. Визначено основні вимоги до структури датасету, способів аугментації даних та підготовки зображень для подальшого навчання нейронної мережі.

На основі проведеного аналізу сформульовано постановку задачі дослідження, яка полягає у розробці кіберфізичної системи управління квадрокоптером із використанням методів машинного навчання для автоматизованого розпізнавання об’єктів. Визначено основні функціональні вимоги до системи, критерії ефективності та напрями подальшого дослідження, що стали основою для розробки моделей, алгоритмів та програмного забезпечення у наступних розділах роботи.

2 РОЗРОБЛЕННЯ МОДЕЛЕЙ ТА МЕТОДІВ ДЛЯ ВИРІШЕННЯ ЗАДАЧ УПРАВЛІННЯ КВАДРОКОПТЕРАМИ ДЛЯ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ЗА ДОПОМОГОЮ МАШИННОГО НАВЧАННЯ

2.1 Математична модель процесу детекції об’єктів

Процес детекції об’єктів у кіберфізичній системі управління БПЛА базується на підході, що розглядає розпізнавання як єдину задачу регресії. Це дозволяє прогнозувати просторові координати та ймовірності класів за один прохід нейронної мережі, забезпечуючи високу швидкість обробки даних у реальному часі [41-47].

Вхідним сигналом для системи є дискретизований відеопотік, де кожен окремий кадр представляється як тривимірний тензор (матриця) I розмірністю:

$$I \in R^{W \times H \times C}, \tag{1.1}$$

де:

– W (Width) та H (Height) – просторові розміри зображення в пікселях (наприклад, 640×640);

– C (Channels) – кількість колірних каналів (зазвичай C=3 для формату RGB).

Процес екстракції ознак у Backbone-мережі реалізується через послідовність операцій згортки.

На виході Head-частини мережі формується тензор прогнозів, що містить інформацію про обмежувальні рамки та класифікаційні ймовірності для кожної клітинки сітки.

Для кожного об’єкта модель прогнозує вектор $B = \{t_x, t_y, t_w, t_h, p_o\}$, де координати центру та розміри обчислюються відносно зміщень клітинки сітки (c_x, c_x).

Навчання моделі полягає в мінімізації комбінованої функції втрат $\mathcal{L}_{total}$, яка у сучасних версіях YOLO (v8–v11) має наступний вигляд:

$$\mathcal{L}_{total} = \lambda_{box}\mathcal{L}_{box} + \lambda_{cls}\mathcal{L}_{cls} + \lambda_{dfl}\mathcal{L}_{dfl}, \tag{1.2}$$

де $\mathcal{L}_{\text{box}}$ (Localization Loss) використовує метрику CIoU (Complete IoU), яка враховує не лише накладання, а й відстань між центрами та співвідношення сторін рамок;

$\mathcal{L}_{\text{cls}}$ (Classification Loss) базується на бінарній крос-ентропії (BCE) для кожного класу;

$\mathcal{L}_{\text{dfl}}$ (Distribution Focal Loss) допомагає моделі точніше визначати межі рамок у випадках нечітких кордонів об’єктів.

Традиційні детектори використовують алгоритм Non-Maximum Suppression (NMS) для видалення дублікатів рамок на етапі постобробки, що створює обчислювальне «вузьке місце». Математична різниця сучасних моделей полягає у впровадженні End-to-End голови з методом One-to-one assignment [48-51].

Традиційний підхід, під час навчання одна реальна мітка зіставляється з декількома прогнозами мережі, що змушує використовувати NMS для вибору найкращого варіанту під час інференсу.

NMS-free підхід, завдяки використанню подвійних гілок навчання та специфічних функцій відповідності, мережа навчається видавати рівно один прогноз для кожного реального об’єкта. Математично це реалізується через мінімізацію вартості призначення за допомогою угорського алгоритму або його аналогів, що дозволяє виключити етап NMS і значно знизити затримку на крайових пристроях.

2.2 Структурно-функціональна архітектура кіберфізичної системи

Проєктування системи управління квадрокоптером з функціями розпізнавання об'єктів базується на парадигмі кіберфізичних систем, де обчислювальний та комунікаційний стрижень забезпечує моніторинг, координацію та контроль фізичних процесів [52-55]. У контексті даного дослідження архітектура CPS розглядається як інтимна взаємодія між дискретною логікою обчислень та безперервною динамікою фізичного світу.

Для забезпечення стабільного функціонування та гнучкості системи її архітектура розподіляється на три взаємозалежні рівні:

– фізичний рівень, що включає безпосередньо конструкцію БПЛА та його апаратне забезпечення, до якого належать сенсори для збору первинних даних про стан середовища та власну орієнтацію, а виконавчі механізми, такі як електронні регулятори швидкості та безколекторні мотори, забезпечують фізичну реалізацію отриманих команд через зміну обертів пропелерів;

– мережевий рівень, що виступає критичною ланкою, що забезпечує інтероперабельність між бортовими системами та зовнішніми ресурсами, де основним проміжним програмним забезпеченням для передачі телеметрії та низькорівневих статусів є протокол MAVLink, який характеризується високою ефективністю при обмеженій пропускній здатності каналів зв'язку, а для інтеграції з високорівневими алгоритмами використовується міст uXRCE-DDS, який транслює повідомлення внутрішньої шини uORB у середовище ROS 2, що дозволяє суттєво знизити затримку передачі даних;

– кібернетичний або обчислювальний рівень, що зосереджений на інтелектуальній обробці інформації, де розгорнуто модулі детекції на базі нейромереж YOLOv11 або YOLO26, які виконують аналіз відеопотоку в реальному часі, де логіка прийняття рішень та формування складних траєкторій реалізується вузлами ROS 2, що дозволяє будувати сервісно-орієнтовану архітектуру управління.

Ефективність управління БПЛА досягається через замкнений цикл зворотного зв'язку, який інтегрує всі рівні системи [56-60]. Процес починається з отримання даних сенсорами фізичного рівня, які передаються через мережеві інтерфейси Wi-Fi або радіоканали до обчислювального блоку.

На кібернетичному рівні відбувається обробка даних нейромережею YOLO, яка ідентифікує цільові об'єкти та визначає їхні координати. На основі отриманих результатів логічний модуль ROS 2 виконує корекцію польотного завдання, розраховуючи нові настановні точки . Фінальний етап циклу передбачає виконання маневру через режим Offboard, де автопілот PX4 транслює високорівневі команди

в сигнали для виконавчих механізмів, забезпечуючи детерміновану реакцію системи на зміни в середовищі.

Використання сервісно-орієнтованої архітектури на базі ROS 2 дозволяє розділити систему на модульні вузли, кожен з яких відповідає за конкретну функцію: детекція, одометрія, навігація. Це підвищує гнучкість системи та її здатність до самоорганізації, що є ключовим для автономних БПЛА.

2.3 Підходи до вирішення задачі за темою дослідження

Для розв'язання задачі інтеграції комп'ютерного зору в автономний БПЛА пропонується комплексний підхід, що базується на концепції кіберфізичних систем. Такий підхід фокусується на інтеграції обчислень, мережевої взаємодії та фізичних процесів, де головним викликом є взаємодія вбудованого ПЗ із фізичним світом.

Програмна платформа та моделі детекції. Використання екосистеми Ultralytics YOLO дозволяє реалізувати гнучкий процес навчання кастомних моделей на специфічних датасетах за допомогою CLI або Python. Основним вибором є моделі YOLO11 та YOLO26. Зокрема, YOLO26 оптимізована для крайового розгортання завдяки NMS-free інференсу, що усуває затримки на етапі постобробки. Для академічних цілей використовується відкрита ліцензія AGPL-3.0, яка сприяє спільній розробці та обміну знаннями.

Архітектура керування та польотний стек. В якості базової операційної системи для БПЛА обрано PX4 Autopilot – відкрите ПЗ під ліцензією BSD 3-clause. Ключовим для даного дослідження є режим Offboard, який дозволяє зовнішнім компонентам, наприклад: серверу або супровідному комп'ютеру; передавати команди керування польотному контролеру в реальному часі. Це забезпечує можливість автономної навігації на основі даних комп'ютерного зору.

Апаратна реалізація та вбудовані системи. Використовуються польотні контролери стандарту Pixhawk, наприклад: серія FMUv6X, як у Pixhawk 6X. Вони з’єднуються з супровідними комп'ютерами через високошвидкісні UART/Serial

порти. Така апаратна конфігурація дозволяє розділити критично важливі польотні завдання та ресурсомістку обробку даних із сервером.

Комунікаційний міст та мережева взаємодія. Для забезпечення зв'язку між нейромережею та автопілотом використовується проміжне ПЗ uXRCE-DDS. Він забезпечує трансляцію внутрішніх повідомлень uORB, таких як координати цілі або статус транспортного засобу, у середовище ROS 2, що працює на сервері або супровідному комп'ютері. Також для передачі команд через канали телеметрії використовується протокол MAVLink.

Алгоритмічний цикл «БПЛА – Сервер – Автопілот». Специфіка підходу полягає у створенні замкненого циклу керування. Відеопотік із бортової камери передається на сервер, де модель YOLO26 виконує детекцію. Отримані координати обробляються логічним модулем, який генерує повідомлення. Ці завдання передаються назад на БПЛА через Offboard інтерфейс, дозволяючи автопілоту PX4 виконувати маневри у фізичному просторі.

2.3.1 Методика оптимізації моделі методом квантизації ваг

При розгортанні моделей глибокого навчання на бортових обчислювальних системах БПЛА критичною проблемою постає «обчислювальне вузьке місце». Це зумовлено обмеженою пропускною здатністю пам’яті та високим енергоспоживанням операцій над числами з рухомою комою. Перехід до використання цілочисельних операцій дозволяє радикально підвищити продуктивність за рахунок ефективнішої утилізації ресурсів апаратного прискорення, оскільки більшість сучасних мобільних архітектур та GPU мають спеціалізовані блоки для виконання 8-бітних арифметичних операцій, що працюють значно швидше за аналоги з FP32.

Симетрична квантизація – діапазон значень центрований відносно нуля, що спрощує математичні розрахунки під час виконання операцій згортки, але може призводити до меншої точності при нерівномірному розподілі даних.

Асиметрична квантизація – використовує повний доступний діапазон цілих чисел через параметр зміщення Z, що дозволяє точніше описувати активації після функції ReLU.

Методологія Post-Training Quantization. Дана методика передбачає оптимізацію вже навченої моделі без потреби у повторному навчанні. Основними етапами є:

– калібрування: використання невеликого репрезентативного датасету для збору статистики розподілу активацій та визначення їхнього оптимального динамічного діапазону;

– мінімізація втрат за Кулльбаком-Лейблером: для знаходження оптимального значення S використовується розбіжність Кулльбака-Лейблера. Цей метод порівнює розподіл значень до та після квантизації, мінімізуючи втрату ентропії та забезпечуючи максимальну схожість інформаційного сигналу при переході до 8 біт.

У моделі YOLOv8n квантизація найбільше впливає на шари згортки, де зосереджена основна маса параметрів та обчислень. Особливістю модифікації YOLO26 є впровадження архітектури NMS-free, що дозволяє усунути етап Non-Maximum Suppression на стадії інференсу. Це критично важливо, оскільки NMS зазвичай виконується на CPU, створюючи затримки [61-63]. Поєднання квантизації ваг та NMS-free архітектури дозволяє всьому конвеєру обробки даних працювати виключно в межах апаратного прискорення.

Застосування описаної методики дозволяє досягти наступних результатів:

– скорочення об’єму пам’яті, необхідного для зберігання моделі, у 4 рази;

– суттєве прискорення швидкості виконання інференсу на цільових платформах завдяки зменшенню навантаження на шину пам'яті та використанню швидких INT8-інструкцій;

– зниження енергоспоживання бортової системи, що є критичним фактором для тривалості автономного польоту БПЛА.

2.3.2 Алгоритмічне забезпечення автономного керування

Для забезпечення автономності БПЛА в задачах стеження та навігації критично важливим є перетворення візуальної інформації, отриманої від сенсорів, у керуючі сигнали для польотного контролера [64-66].

Для реалізації автономного руху використовується підхід Image-Based Visual Servoing (IBVS). Основною особливістю IBVS є те, що зворотний зв'язок замикається безпосередньо в площині зображення.

Цей підхід дозволяє уникнути складних обчислень повної 3D-реконструкції сцени, що критично для систем з обмеженими обчислювальними ресурсами. Керування здійснюється шляхом мінімізації цієї помилки через зміну швидкостей руху дрона.

Для формування команд польотному контролеру необхідно перерахувати координати об'єкта з піксельної системи камери в систему координат тіла дрона та глобальну систему.

Використовуючи матрицю внутрішніх параметрів камери K проводиться перехід від пікселів (u, v) до нормованих координат (x_c, y_c).

Отримані координати переносяться в систему тіла дрона з урахуванням матриці повороту та вектора зсуву камери відносно центру мас апарата.

Для навігації в режимі Offboard координати трансформуються в локальну систему за допомогою поточної орієнтації дрона, що отримується від польотного контролера.

Керування здійснюється в режимі Offboard через передачу повідомлень до польотного стека PX4 [67-69]. Логіка формування команд базується на відхиленні центру детектованого об'єкта $(\Delta u, \Delta v)$ від центру кадру.

Формується пропорційно відхиленню по осях, що передбачає рух вперед/назад регулювати розміром об'єкта в кадрі, а рух вліво/вправо – відхиленням Δu.

Для утримання об'єкта в центрі кадру по горизонталі генерується команда на обертання. Оскільки детектори сімейства YOLO можуть видавати зашумлені дані

або тимчасово втрачати об'єкт, застосовується фільтр Калмана, що дозволяє системі прогнозувати положення цілі під час короткочасних зникнень детекції, запобігаючи різким стрибкам дрона та забезпечуючи плавність польоту.

Дискретизація команд керування в Offboard режимі повинна відбуватися з частотою не менше 2 Гц для PX4, проте для якісного візуального стеження вона має бути синхронізована з FPS нейромережі [70-73]. Якщо затримка обробки кадру нейромережею YOLO перевищує час між оновленнями команд, виникає розсинхронізація, що призводить до нестійкості контуру керування. Тому частота відправки адаптується до темпу виходу даних з конвеєра комп'ютерного зору.

2.4 Висновки до розділу 2

У другому розділі було розроблено моделі та методи для вирішення задач управління квадрокоптерами з використанням технологій машинного навчання та комп’ютерного зору. Основну увагу приділено формуванню математичного та алгоритмічного підходу до детекції об’єктів у складі кіберфізичної системи управління БПЛА. Визначено, що застосування сучасних моделей глибокого навчання дозволяє забезпечити автоматизоване розпізнавання об’єктів у режимі реального часу з достатнім рівнем точності та швидкодії.

У межах розділу сформовано математичну модель процесу детекції об’єктів, яка базується на використанні нейронних мереж для одночасного визначення координат об’єктів та їх класів. Встановлено, що такий підхід дозволяє скоротити час обробки вхідних даних та забезпечити швидке прийняття рішень у процесі автономного керування квадрокоптером.

Розроблено структурно-функціональну архітектуру кіберфізичної системи, яка об’єднує модулі отримання відеоданих, попередньої обробки інформації, нейромережевого аналізу, передачі команд та системи керування польотом. Запропонована архітектура забезпечує взаємодію між програмними та апаратними компонентами системи, а також підтримує можливість масштабування та модернізації окремих модулів.

3 ПРОЄКТУВАННЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ УПРАВЛІННЯ КВАДРОКОПТЕРАМИ ДЛЯ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ЗА ДОПОМОГОЮ МАШИННОГО НАВЧАННЯ

3.1 Аналіз функціональних компонентів кіберфізичної системи

Основним призначенням розробленої кіберфізичної системи є автоматизоване виявлення та розпізнавання об’єктів на основі відеопотоку, отриманого з бортових сенсорів безпілотного літального апарата. Система повинна забезпечувати обробку вхідних даних у режимі реального часу, підтримувати адаптацію моделей машинного навчання до нових типів об’єктів та забезпечувати передачу результатів розпізнавання до підсистеми керування БПЛА.

Однією з ключових вимог до системи є забезпечення високої швидкості аналізу відеоданих при збереженні достатньої точності детекції об’єктів. Крім цього, система повинна підтримувати можливість розширення датасету, перенавчання моделей та інтеграцію нових класів цілей без необхідності зміни базової архітектури програмного забезпечення.

3.1.1. Функціональний аналіз системи

Для формування архітектури кіберфізичної системи управління квадрокоптером проведено функціональний аналіз основних процесів обробки та аналізу даних. На основі цього аналізу сформовано дерево функцій системи, яке відображає взаємодію користувацьких, аналітичних та адміністративних компонентів програмного забезпечення.

У структурі системи можна виділити три основні групи функцій:

– функції оператора системи;

– функції підсистеми машинного навчання;

– функції адміністративного керування.

До функцій оператора системи належать:

– завантаження фото- та відеоданих із БПЛА;

– попередня обробка отриманих кадрів;

– запуск алгоритмів виявлення та класифікації об’єктів;

– відображення результатів аналізу відеопотоку;

– перегляд параметрів виявлених об’єктів;

– передача результатів до підсистеми керування польотом.

Графічне представлення функціональної структури кіберфізичної системи наведено на рис. 3.1.

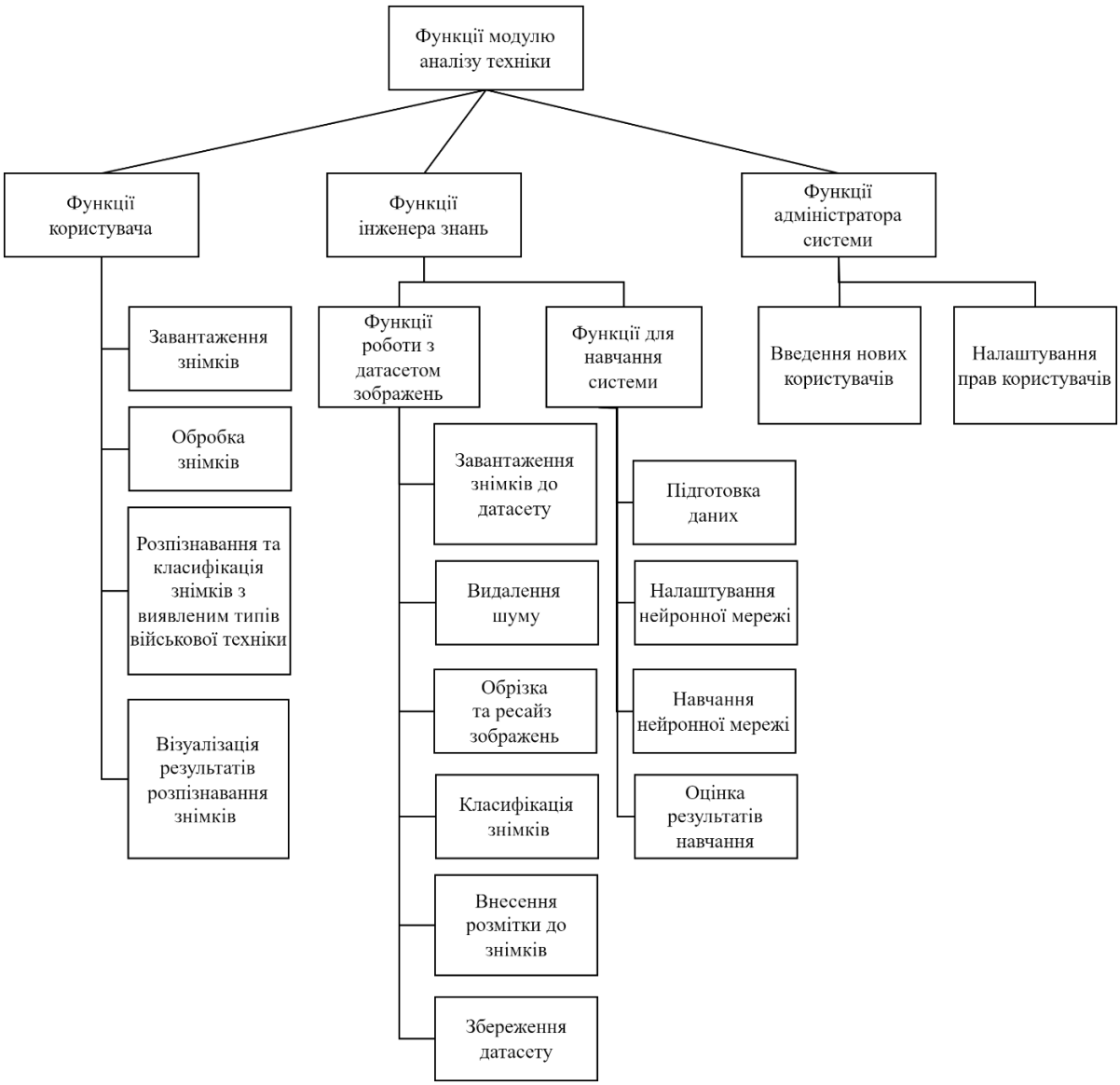


Рисунок 3.1 – Дерево функцій кіберфізичної системи управління БПЛА з модулем комп’ютерного зору

Функції підсистеми машинного навчання забезпечують формування та підтримку наборів даних, а також навчання моделей комп’ютерного зору. До цієї групи належать:

1. Функції роботи з датасетами:

– завантаження кадрів аерозйомки до системи;

– фільтрація та очищення даних;

– нормалізація та масштабування зображень;

– аугментація навчальних даних;

– розмітка об’єктів на зображеннях;

– формування тренувальних і тестових вибірок;

– збереження датасетів у форматах, сумісних із моделями YOLO.

2. Функції навчання моделей:

– підготовка даних до навчання;

– налаштування архітектури нейронної мережі;

– запуск процесу навчання моделі;

– оцінювання точності моделі;

– контроль перенавчання;

– збереження ваг нейронної мережі;

– тестування моделі на валідаційних наборах даних;

– моніторинг процесу навчання та аналіз метрик.

До функцій адміністративної підсистеми належать:

– керування обліковими записами користувачів;

– налаштування рівнів доступу;

– конфігурація параметрів системи;

– контроль доступу до моделей та датасетів.

Після функціонального аналізу системи наступним етапом є формування моделі її функціонування та опис взаємодії основних компонентів кіберфізичної системи управління БПЛА.

3.1.2 Моделювання функціонування модуля розпізнавання

Контекстна діаграма функціонування кіберфізичної системи управління квадрокоптером із модулем комп’ютерного зору в нотації IDEF0 наведена на рис. 3.2.

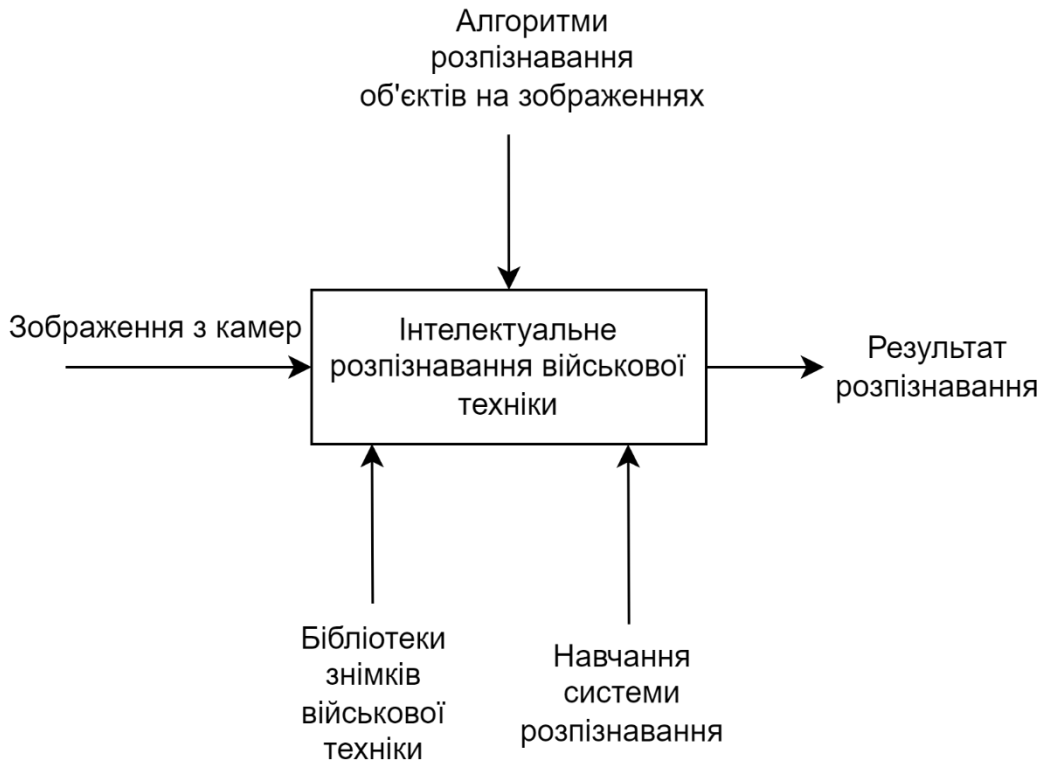


Рисунок 3.2 – Контекстна діаграма функціонування модуля розпізнавання кіберфізичної системи управління БПЛА

На контекстній діаграмі показано загальний процес обробки вхідного відеопотоку, отриманого з бортових камер безпілотного літального апарата. На першому етапі виконується попередня обробка кадрів, яка включає нормалізацію, фільтрацію шуму, зміну масштабу зображень та підготовку даних до подальшого аналізу. Після цього дані передаються до модуля комп’ютерного зору, який використовує алгоритми глибокого навчання для виявлення, класифікації та локалізації об’єктів.

Результатом роботи системи є координати виявлених цілей, класи об’єктів, параметри впевненості детекції та службова інформація, яка може бути використана підсистемою навігації та автономного керування БПЛА.

Основними компонентами кіберфізичної системи є:

– відеопотік із бортових сенсорів БПЛА;

– оператор системи;

– підсистема машинного навчання;

– модуль комп’ютерного зору;

– результати розпізнавання;

– навчена нейронна мережа;

– підсистема керування польотом.

На вході система отримує потік зображень або окремі кадри аерозйомки. Залежно від обраного режиму роботи система може виконувати або детекцію об’єктів у режимі реального часу.

Для деталізації контекстної моделі виконано декомпозицію системи на два основних функціональних процеси:

– процес розпізнавання об’єктів;

– процес навчання нейронної мережі.

Відповідна діаграма декомпозиції першого рівня наведена на рис. 3.3.

Процес розпізнавання об’єктів включає:

– отримання та завантаження кадрів із БПЛА;

– попередню обробку відеоданих;

– виділення об’єктів;

– запуск моделі детекції;

– класифікацію цілей;

– формування результатів аналізу;

– передачу координат і параметрів об’єктів до підсистеми керування.

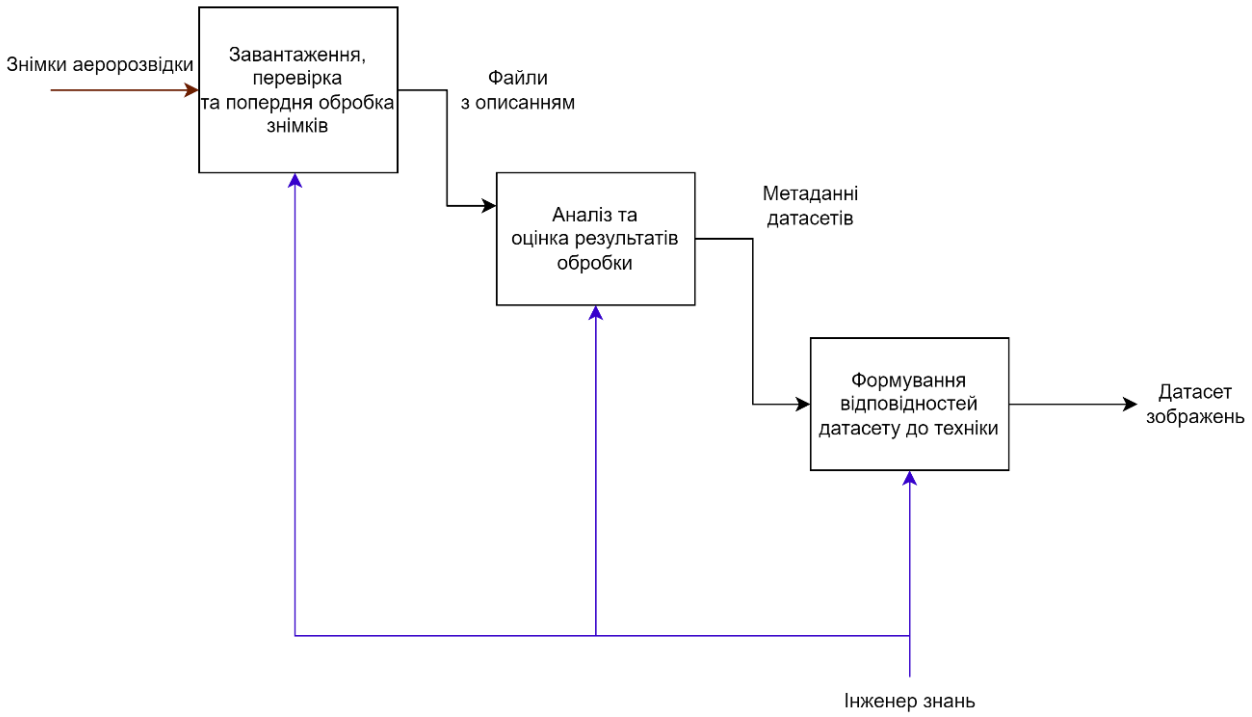


Рисунок 3.3 – Діаграма декомпозиції IDEF0 першого рівня

Результатом виконання цього процесу є сформований набір параметрів розпізнаних об’єктів, який використовується для підтримки прийняття рішень та автономного керування квадрокоптером.

Процес навчання нейронної мережі включає:

– формування датасету;

– аналіз і перевірку навчальних даних;

– аугментацію зображень;

– налаштування параметрів моделі;

– навчання нейронної мережі;

– оцінювання метрик точності;

– збереження ваг навченої моделі.

Результатом даного процесу є підготовлена модель комп’ютерного зору, адаптована до роботи з даними аерозйомки.

На рис. 3.4 наведено діаграму процесу завантаження та попередньої обробки кадрів відеопотоку.

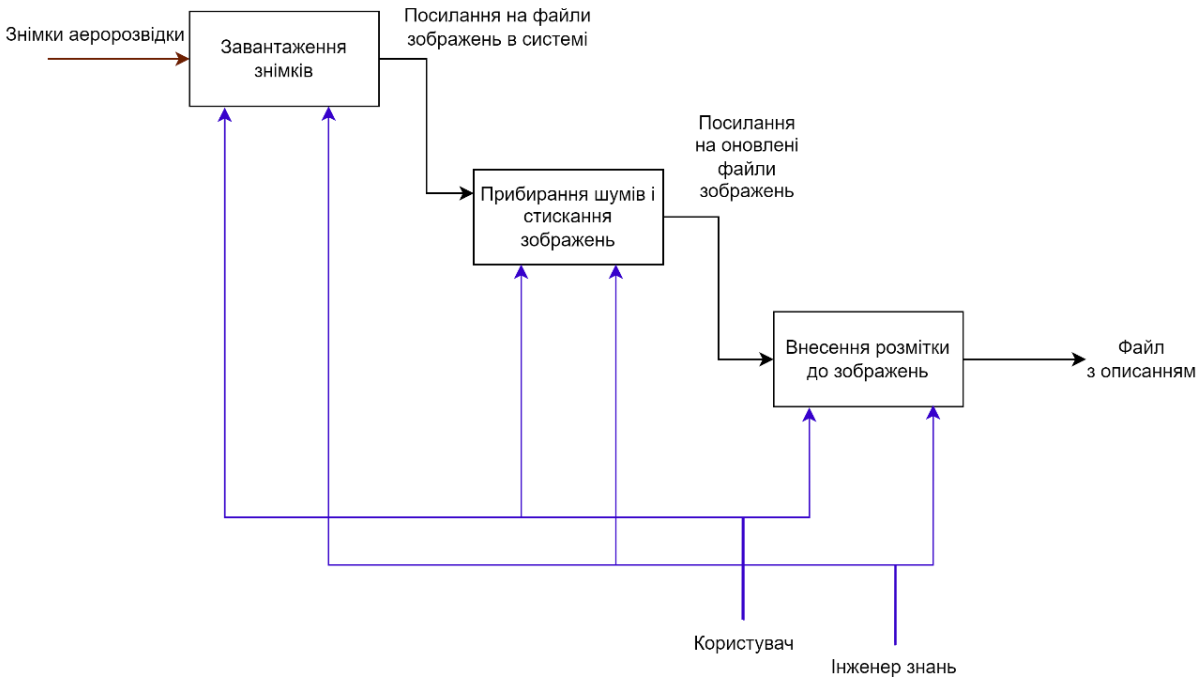


Рисунок 3.4 – Діаграма процесу попередньої обробки відеоданих

На рис. 3.5 представлено декомпозицію процесу виявлення та класифікації об’єктів.

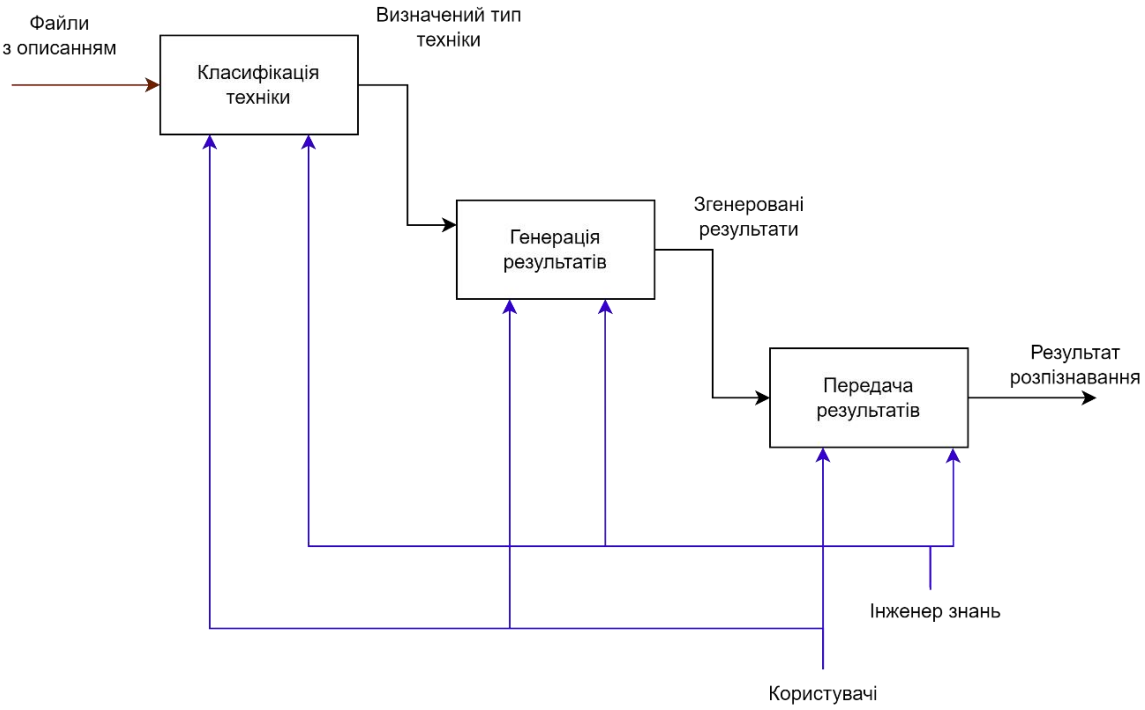


Рисунок 3.5 – Діаграма процесу детекції та класифікації об’єктів

На рис. 3.6 наведено діаграму формування та аналізу датасету для навчання нейронної мережі.

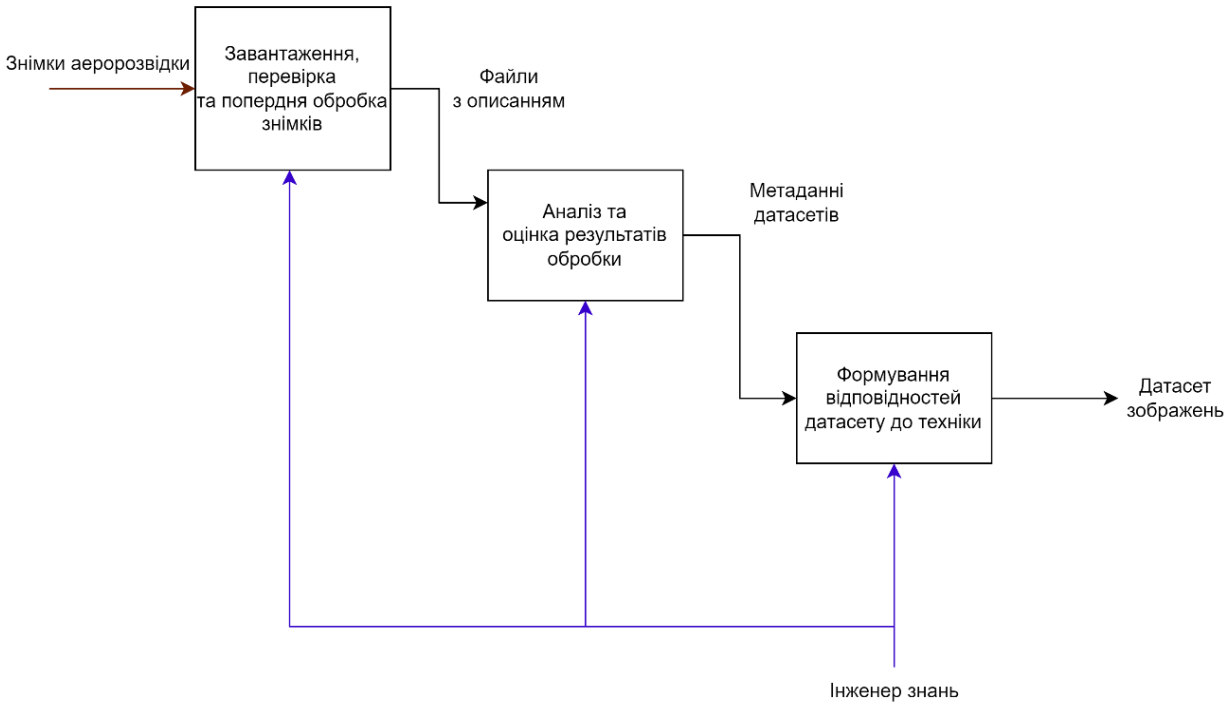


Рисунок 3.6 – Діаграма процесу формування датасету

На рис. 3.7 наведено діаграму процесу навчання моделі машинного навчання.

На основі декомпозиції функціональних процесів та визначених вимог до системи сформовано UML-діаграму варіантів використання, яка наведена на рис. 3.8.

На діаграмі відображено взаємодію оператора, адміністратора та підсистеми машинного навчання із компонентами кіберфізичної системи. При цьому адміністративні функції керування користувачами та конфігурацією системи доцільно реалізувати у вигляді окремої сервісної підсистеми, що дозволить зменшити навантаження на основний контур обробки відеоданих та автономного керування БПЛА.

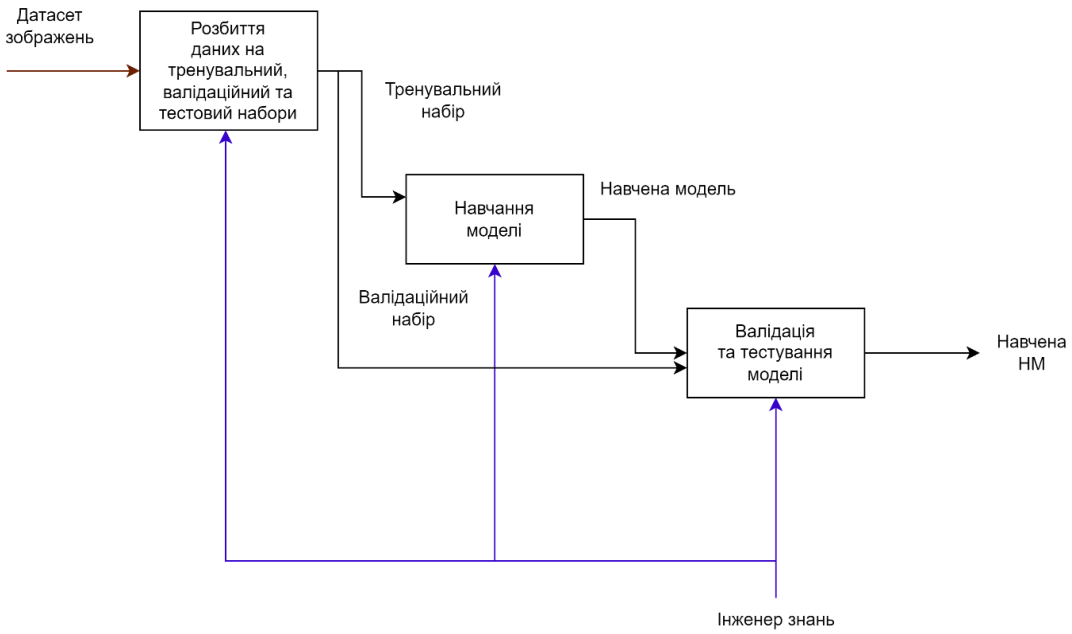


Рисунок 3.7 – Діаграма процесу навчання нейронної мережі

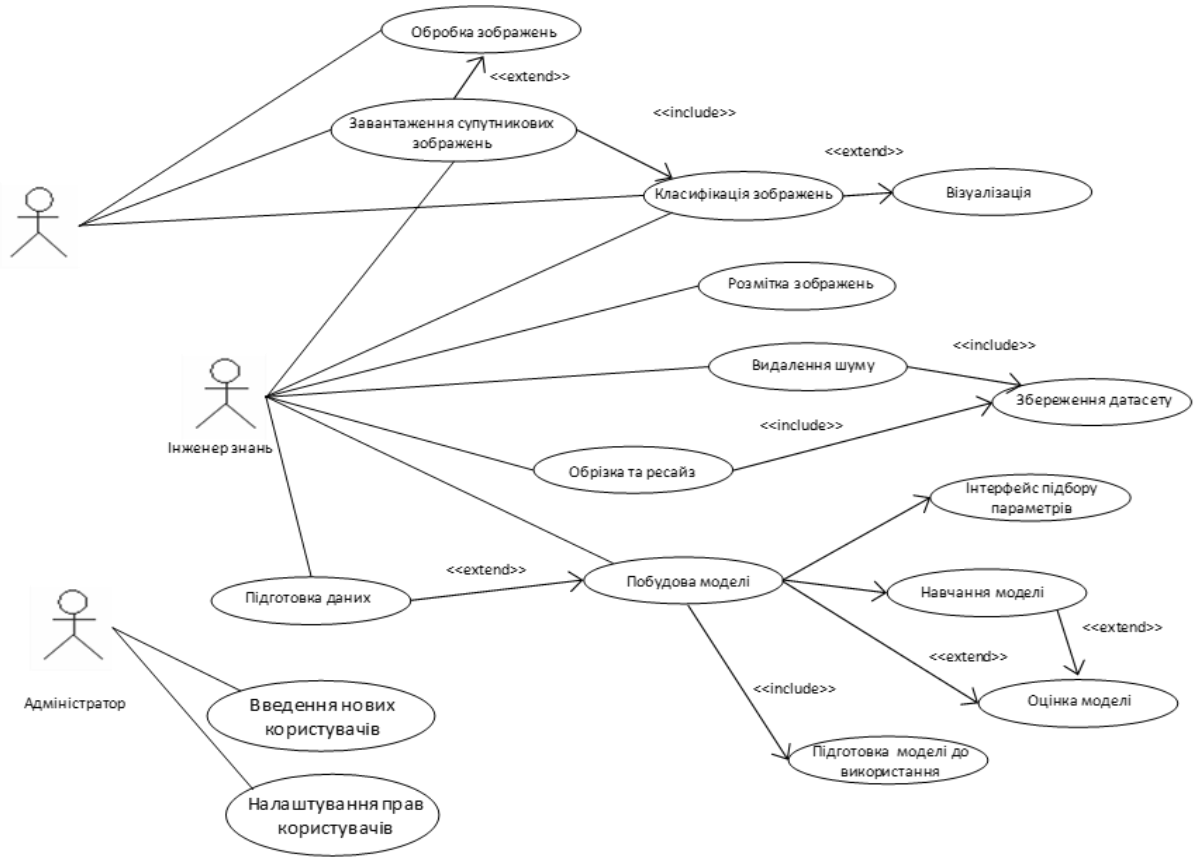


Рисунок 3.8 – UML-діаграма варіантів використання модуля комп’ютерного зору

3.2 Проектування архітектури програмного комплексу кіберфізичної системи

Архітектура програмного комплексу управління квадрокоптером базується на парадигмі кіберфізичних систем, що передбачає тісну інтеграцію обчислювальних алгоритмів, мережевої взаємодії та фізичних процесів [74-76]. Основним викликом при проектуванні такої системи є забезпечення детермінованості передачі даних та мінімізація латентності в контурі прийняття рішень.

У межах даного дослідження розроблено розподілену архітектуру, що дозволяє оптимально розподілити обчислювальні ресурси між бортовим обладнанням та зовнішніми потужностями. Логіка розподілу обчислень реалізована наступним чином:

1. Бортовий польотний контролер Pixhawk (рис. 3.9) відповідає за фізичний рівень Physical Layer, виконуючи критично важливі завдання стабілізації, обробки даних IMU та низькорівневого керування моторами під управлінням операційної системи реального часу.

2. Супровідний комп’ютер або сервер виконує роль обчислювального рівня, де реалізуються ресурсомісткі алгоритми комп'ютерного зору та планування складних траєкторій.

Рисунок 3.9 – Вигляд бортового польотного контролера Pixhawk

Програмний комплекс побудований на основі мікросервісного підходу в середовищі ROS 2, де функціональність системи розподілена між незалежними вузлами . Взаємодія між вузлами здійснюється за принципом публікація-підписка через спеціалізовані топіки. video_streamer_node забезпечує захоплення та трансляцію відеопотоку з камери БПЛА на аналітичний рівень. yolo_inference_node виконує інференс оптимізованої моделі YOLOv8n або YOLO26. Для підвищення продуктивності використовується 8-бітна квантизація, що дозволяє скоротити об'єм пам'яті у 4 рази та прискорити обробку даних на апаратних прискорювачах. offboard_control_node: логічний центр системи, що інтерпретує результати розпізнавання та формує команди керування [77-79].

Комунікаційний рівень системи базується на стандарті DDS та реалізований через міст uXRCE-DDS. Цей протокол забезпечує пряму трансляцію внутрішніх повідомлень автопілота у топіки ROS 2, що дозволяє створити стійкий замкнений контур управління.

Логіка обробки даних у системі реалізує повний цикл від сенсора до фізичної дії. Процес починається з ідентифікації цільового об'єкта вузлом yolo_inference_node. Отримані координати обробляються у offboard_control_node, де на їх основі розраховується відхилення від цільової точки. Результат обробки перетворюється на навігаційну команду – цільову точку траєкторії. Через повідомлення ці дані передаються до польотного стека PX4 Autopilot у режимі Offboard, що забезпечує автономне маневрування та стеження за об'єктом у реальному часі. Така архітектура дозволяє використовувати потужні серверні ресурси для складних AI-завдань, зберігаючи при цьому стабільність польоту, що гарантується вбудованими алгоритмами автопілота.

3.3 Проєктування інформаційного забезпечення кіберфізичної системи

Для забезпечення роботи кіберфізичної системи управління квадрокоптером із модулем комп’ютерного зору необхідно реалізувати інформаційне забезпечення, яке підтримує зберігання навчальних даних, параметрів моделей машинного

навчання, результатів детекції об’єктів та службової інформації, що використовується під час аналізу відеопотоку.

База даних системи повинна забезпечувати:

– збереження кадрів аерозйомки та супровідних метаданих;

– збереження інформації про класи об’єктів;

– збереження параметрів моделей машинного навчання;

– накопичення результатів класифікації та детекції;

– підтримку процесів навчання і перенавчання моделей;

– швидкий доступ до результатів аналізу відеоданих.

У структурі бази даних передбачено збереження зображень різних типів об’єктів, які використовуються під час формування навчального датасету. До таких об’єктів можуть належати бронетехніка, автомобільна техніка, артилерійські системи, безпілотні комплекси та інші типи цілей, що розпізнаються модулем комп’ютерного зору.

Крім самих зображень, база даних містить інформацію про:

– тип об’єкта;

– модель техніки;

– параметри об’єкта;

– результати класифікації;

– рівень впевненості нейронної мережі;

– характеристики моделей машинного навчання;

– параметри попередньої обробки зображень.

Також у системі зберігаються результати попереднього аналізу кадрів, які включають:

– координати виявлених об’єктів;

– розміри bounding box;

– параметри орієнтації;

– службові характеристики відеопотоку;

– параметри освітлення;

– часові мітки кадрів.

Наявність такої інформації дозволяє виконувати подальший аналіз результатів роботи моделі та використовувати накопичені дані для перенавчання нейронної мережі.

Для формалізації структури інформаційного забезпечення побудовано концептуальну модель бази даних, яка наведена на рис. 3.10.

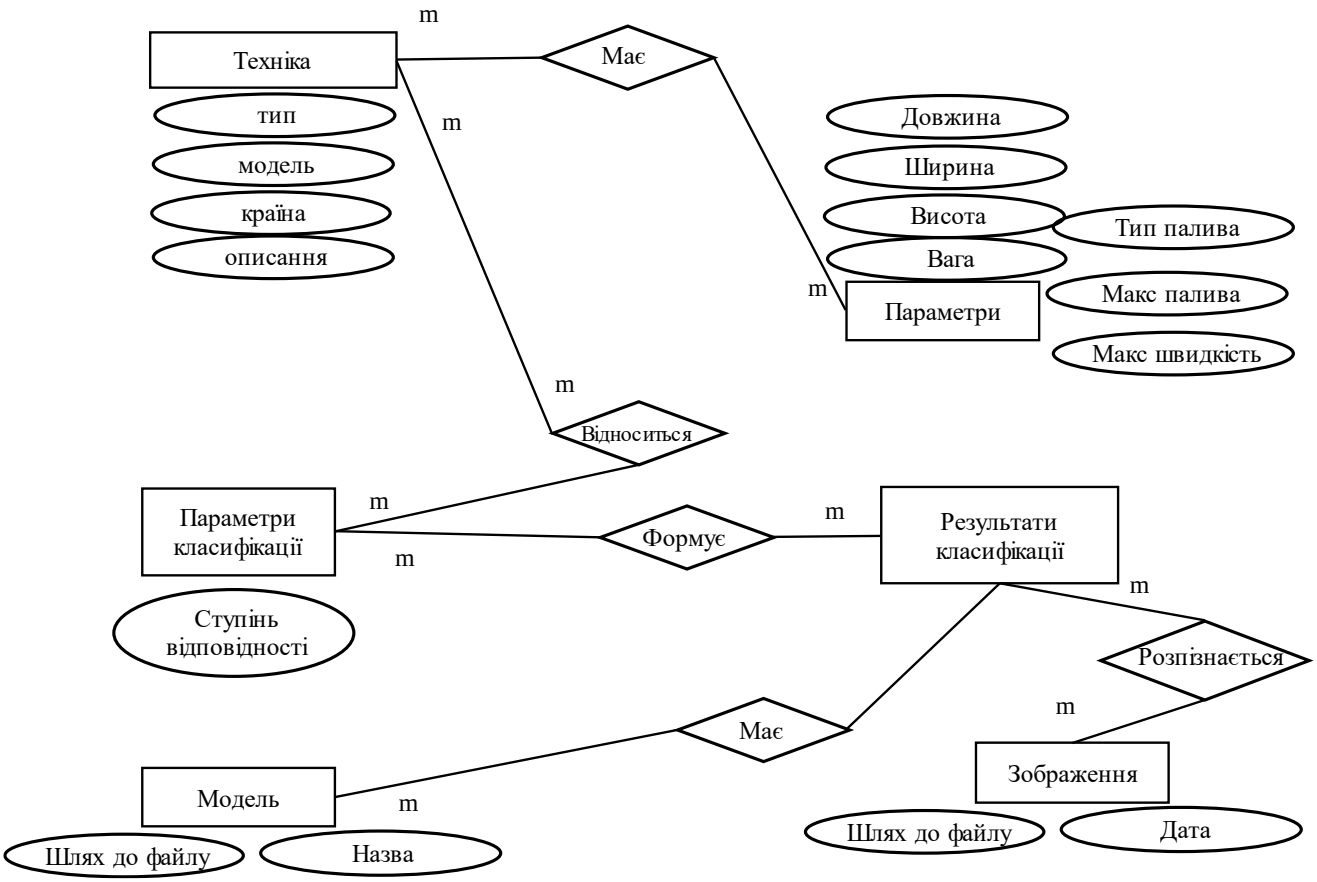


Рисунок 3.10 – Концептуальна модель бази даних кіберфізичної системи

Концептуальна модель відображає основні сутності системи та зв’язки між ними.

На основі концептуальної моделі сформовано логічну модель бази даних, яка наведена на рис. 3.11.

Після формування логічної моделі виконано проєктування фізичної структури бази даних. Для зберігання інформації про класи об’єктів використовується таблиця Vehicles, яка містить такі поля:

– vehicle_id – унікальний ідентифікатор об’єкта;

– vehicle_type – тип об’єкта;

– vehicle_model – модель техніки;

– vehicle_country – країна виробник;

– description – опис об’єкта.

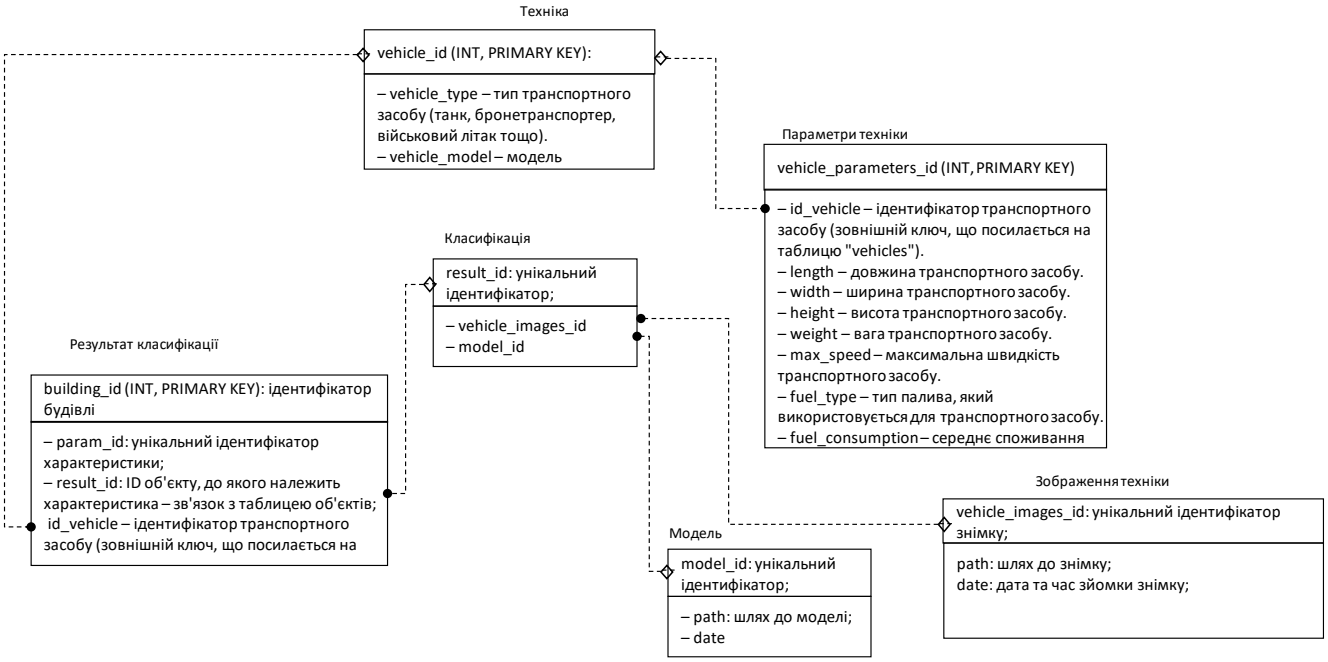


Рисунок 3.11 – Логічна модель бази даних

Для збереження параметрів техніки використовується таблиця Vehicle_Parameters, до складу якої входять:

– vehicle_params_id – ідентифікатор параметрів;

– length – довжина;

– width – ширина;

– height – висота;

– weight – маса;

– max_speed – максимальна швидкість;

– fuel_type – тип палива;

– fuel_max – об’єм паливної системи.

Така структура дозволяє зберігати технічні характеристики об’єктів, які можуть бути використані під час додаткового аналізу та ідентифікації.

Для роботи із результатами детекції передбачено таблицю Classification_Results, яка містить інформацію про результати класифікації окремих кадрів.

У таблиці Classification_Parameters зберігаються:

– параметри класифікації;

– ідентифікатори результатів;

– зв’язки з типами об’єктів;

– рівень впевненості моделі (vehicle_probability).

Для збереження інформації про використані моделі машинного навчання використовується таблиця Models, яка містить:

– model_id – ідентифікатор моделі;

– model_name – назва моделі;

– model_path – шлях до файлу ваг нейронної мережі.

Зображення та кадри відеопотоку зберігаються у таблиці Vehicle_Images, яка містить:

– image_id – ідентифікатор зображення;

– image_path – шлях до файлу;

– image_date – дата формування кадру.

Фізична модель бази даних наведена на рис. 3.12.

Запропонована структура інформаційного забезпечення дозволяє:

– забезпечити централізоване збереження результатів роботи системи;

– організувати швидкий доступ до навчальних даних;

– підтримувати процес перенавчання моделей;

– накопичувати результати роботи алгоритмів детекції;

– реалізувати масштабування системи при збільшенні обсягу відеоданих.

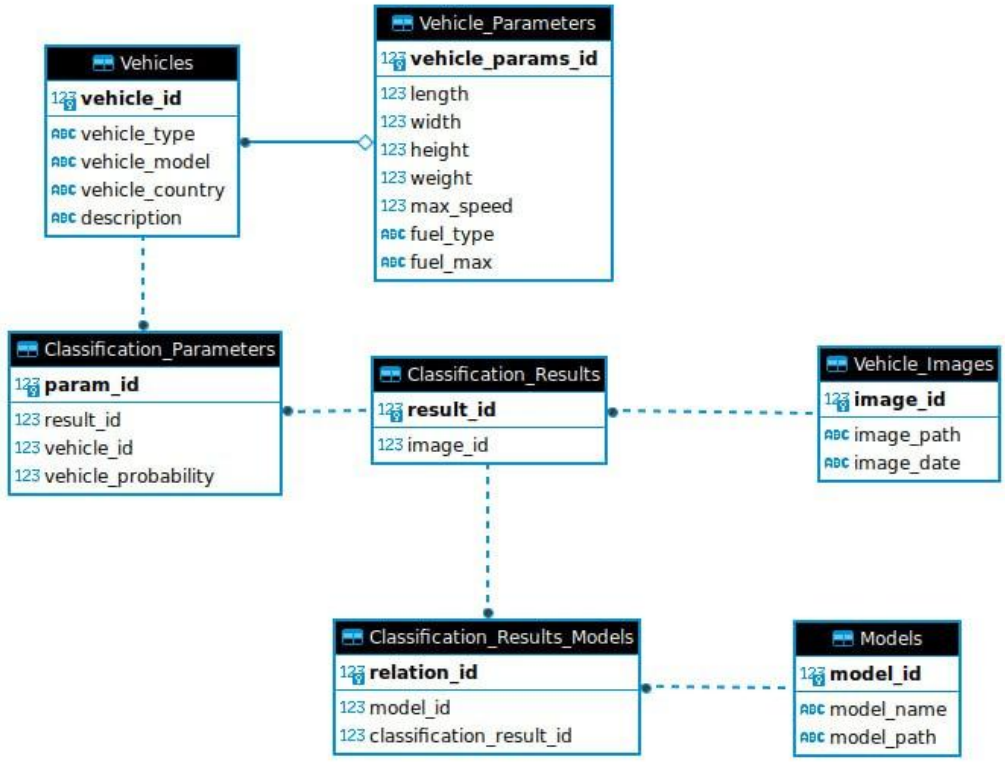


Рисунок 3.12 – Фізична модель бази даних

3.4 Розроблені алгоритми кіберфізичної системи

3.4.1 Попередня обробка відеоданих

Для забезпечення високої продуктивності системи детектування об'єктів у режимі реального часу на бортовому комп'ютері квадрокоптера реалізовано оптимізований конвеєр обробки даних. Цей алгоритм охоплює шлях від отримання «сирого» кадру з сенсора до інтерпретації результатів розпізнавання нейромережею YOLOv8n. Основна мета даного етапу – мінімізація затримки та максимізація частоти кадрів шляхом ефективного використання апаратних ресурсів.

Етапи обробки відеоданих:

1. Захват та декодування відеопотоку, де проводиться отримання цифрового сигналу з камери і залежно від типу підключення, система використовує бібліотеку OpenCV у поєднанні з фреймворком GStreamer для апаратного декодування відеопотоку [80-84]. Для CSI-камер використовується специфічний рядок конвеєра

GStreamer, що дозволяє задіяти ISP-процесор для первинної корекції зображення. Функція cv::VideoCapture ініціалізує потік, забезпечуючи передачу кожного кадру в пам'ять бортового комп'ютера у форматі BGR.

2. Попередня обробка, що передбачає зміну розміру методом Letterbox вхідного зображення з камери, яке масштабується до цільового розміру 640x640 пікселів [1, 81, 83-86]. Конвертація кольорового простору, оскільки OpenCV стандартно працює у просторі BGR, виконується перетворення у формат RGB, на якому базувалося навчання моделі YOLOv8. Нормалізація – значення інтенсивності кожного пікселя в діапазоні перераховуються у плаваючу точку [0.0, 1.0] шляхом ділення на 255. Це необхідно для стабілізації роботи активаційних функцій нейромережі. Транспонування осей, виконується перехід від формату зберігання даних HWC до формату CHW, що є стандартом для більшості тензорних обчислювачів.

3. Етап інференсу та INT8-оптимізація, де підготовлений тензор передається в тензорний прискорювач, роль якого виконує обчислювальне ядро TensorRT [2, 5, 84-86]. Ключовою особливістю цього етапу є використання квантованих ваг моделі у форматі INT8. TensorRT автоматично оптимізує граф нейромережі, зливаючи шари та обираючи найбільш ефективні алгоритми згортки для конкретної архітектури бортового обчислювача.

4. Етап формування вихідних даних та їх інтерпретація, де вихідний тензор містить прогнозовані параметри об'єктів (Bounding Boxes – координати центрів та розміри знайдених об'єктів). Для їх коректного відображення на вихідному відеопотоці застосовується денормалізація та зворотне перетворення координат з урахуванням зміщення, а показник Confidence Scores визначає ступінь впевненості моделі у присутності об'єкта в даній зоні. Система інтерпретує ці дані через вузол ROS 2, який фільтрує результати за порогом впевненості та передає координати об'єктів до алгоритмів автоматичного стеження або навігації.

3.4.2 Перетворення екранних координат об'єкта в команди навігації

Для реалізації автономного слідування за динамічним об'єктом використовується концепція візуального сервокерування, де керуючий сигнал формується безпосередньо на основі аналізу зображення з камери. Ключовою особливістю даної реалізації є робота польотного контролера в режимі Offboard, що дозволяє зовнішній системі перехоплювати керування та передавати команди навігації в обхід стандартних алгоритмів автопілота PX4.

Процес керування починається з визначення вектора стану об'єкта в системі координат зображення. Нехай (x_c, y_c) – координати центру детекції об'єкта в пікселях, а W та H – ширина та висота кадру відповідно. Для забезпечення універсальності алгоритму незалежно від роздільної здатності камери, виконується перехід до нормалізованих координат у діапазоні від 1 до 1. У цій моделі оптичний центр кадру має координати (0, 0), що спрощує розрахунок похибки для зворотного зв'язку.

Для зміни курсу дрона необхідно трансформувати 2D-похибку зображення у вектор зміщення або кутові швидкості в системі координат апарата. Згідно зі стандартами PX4, використовується орієнтація FRD (Forward, Right, Down):

– похибка по осі X зображення (e_x) корелює з поворотом навколо осі Z дрона – кутом нахилу;

– похибка по осі Y зображення (e_y) використовується для корекції кута тангажу або лінійної швидкості по осі X для підтримки дистанції до об'єкта.

Для мінімізації відхилення центру об'єкта від центру кадру застосовується ПІД-регулятор.

Враховуючи цифрову природу системи, проводиться дискретизація алгоритму, де інтегральна складова накопичується на кожному кроці ітерації, а диференціальна – оцінює швидкість зміни похибки між кадрами:

– канал Yaw формує кутову швидкість ω_z для розвороту носа дрона на об'єкт;

– канал Pitch формує лінійну швидкість v_x або нахил для наближення/віддалення.

Динаміка системи описується через передавальну функцію, де коефіцієнти K_p, K_i, K_d підбираються таким чином, щоб забезпечити критичне демпфування без надлишкових осциляцій.

Розраховані значення швидкостей та напрямку упаковуються в повідомлення типу px4_msgs::msg::TrajectorySetpoint. Ці дані передаються з бортового комп'ютера на польотний контролер через мікросервісний міст uXRCE-DDS.

Важливо, щоб частота публікації setpoint була стабільною, не менше 20 Гц, оскільки в режимі Offboard PX4 автоматично переходить у режим Safe Stop, якщо зв'язок із зовнішнім алгоритмом переривається на час більше 0.5 секунди.

Для стабілізації польоту та захисту механічних вузлів в алгоритм закладено наступні обмеження:

Якщо $|e_x| < \in$ та $|e_y| < \in$, команда на зміну курсу не подається. Це запобігає "смиканню" дрона через мікроколивання об'єкта або шум сенсора.

Максимальні значення кутової та лінійної швидкостей жорстко обмежені параметрами MPC_XY_VEL_MAX та MPC_YAW_RA_MAX в налаштуваннях PX4, щоб уникнути неконтрольованого прискорення при різкому зникненні об'єкта з кадру.

3.4.3 Формування та навчання моделі комп’ютерного зору

Одним із ключових етапів розробки кіберфізичної системи управління квадрокоптерами є формування та навчання моделі комп’ютерного зору для автоматизованого виявлення й класифікації об’єктів у відеопотоці. Для реалізації програмної частини системи використано бібліотеки TensorFlow, OpenCV та Keras, які забезпечують підтримку алгоритмів глибокого навчання, обробки зображень та оптимізації нейронних мереж.

Під час побудови системи було реалізовано комплекс алгоритмів попередньої обробки та аналізу відеоданих, що забезпечують стабільну роботу моделі в умовах аерозйомки та динамічного руху БПЛА.

До основних етапів обробки даних належать:

1) попередня обробка зображень, де перед передачею кадрів до нейронної мережі виконується попередня підготовка відеоданих, що включає зміну розміру зображень, нормалізацію піксельних значень, фільтрацію шумів, перетворення зображень у формат, придатний для обробки моделлю, підвищення контрастності та стабілізацію кадрів;

2) алгоритми локалізації та сегментації об’єктів, де для виділення об’єктів на кадрах застосовуються алгоритми локалізації та сегментації, які забезпечують визначення положення цілей у межах зображення. У процесі аналізу використовуються підходи semantic segmentation та instance segmentation, що дозволяють: виділяти окремі об’єкти, визначати межі цілей, формувати координати bounding box, передавати результати до підсистеми навігації квадрокоптера;

3) витягнення ознак, де для підвищення точності розпізнавання реалізовано механізми автоматичного витягнення ознак із вхідних кадрів. У процесі роботи моделі аналізуються: геометричні характеристики, текстурні особливості, форма об’єктів, спектральні характеристики, просторові співвідношення елементів зображення;

4) класифікація об’єктів, де застосовується нейронна мережа сімейства YOLO, адаптована до роботи з даними аерозйомки. Модель забезпечує: детекцію об’єктів у режимі реального часу, визначення класу цілі, оцінювання рівня впевненості, формування координат об’єктів;

5) оптимізація продуктивності системи, де використовуються апаратне прискорення GPU, пакетна обробка даних, оптимізація структури нейронної мережі, квантизація моделей, механізми кешування проміжних результатів;

6) формування ансамблів моделей, де для підвищення стійкості системи до помилок класифікації можуть використовуватись ансамблі моделей машинного навчання. У такому випадку результати декількох моделей об’єднуються для формування остаточного рішення щодо належності об’єкта до певного класу.

Схему алгоритму попередньої обробки та навчання моделі наведено на рис. 3.13, де на першому етапі алгоритму виконується завантаження набору зображень та відповідних міток класів. Після цього здійснюється попередня

обробка даних, яка включає нормалізацію, масштабування та перетворення зображень у формат, придатний для роботи нейронної мережі.

Рисунок 3.13 – Схема алгоритму попередньої обробки та навчання моделі комп’ютерного зору

Далі виконується аугментація даних та формування навчальних і тестових вибірок. Після завершення підготовки даних система ініціалізує модель нейронної мережі та запускає процес навчання. Результатом виконання алгоритму є сформована модель комп’ютерного зору, придатна до використання у складі кіберфізичної системи управління квадрокоптерами.

Для роботи із вхідними даними також реалізовано алгоритм попереднього аналізу зображень, який дозволяє автоматично перевіряти коректність завантажених кадрів та виконувати їхню фільтрацію перед запуском процесу детекції.

Схему алгоритму роботи користувацького інтерфейсу системи наведено на рис. 3.14.

Алгоритм роботи користувацького інтерфейсу модуля комп’ютерного зору починається із введення оператором параметрів аналізу та завантаження вхідних даних. На даному етапі користувач може обрати окремі зображення, серію кадрів або відеопотік, отриманий із бортових камер безпілотного літального апарата.

Після завантаження даних система виконує перевірку наявності службових характеристик зображення. До таких характеристик належать:

– дата та час формування кадру;

– координати GPS;

– параметри висоти польоту;

– просторове положення БПЛА;

– роздільна здатність зображення;

– параметри експозиції;

– орієнтація камери;

– службові EXIF-дані.

Наявність метаданих дозволяє системі підвищити якість подальшого аналізу та забезпечує можливість просторової прив’язки результатів розпізнавання.

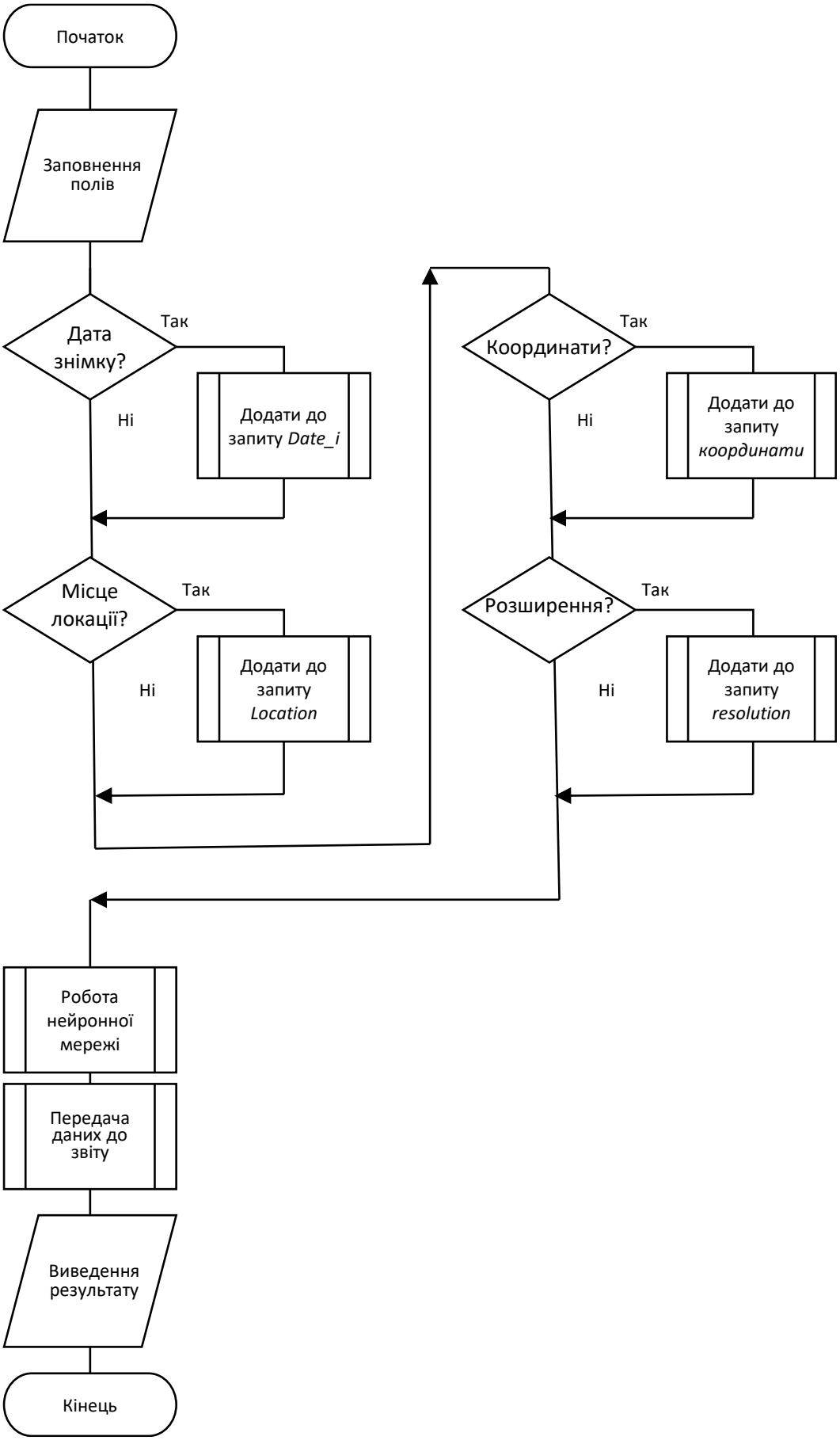


Рисунок 3.14 – Схема алгоритму роботи інтерфейсу модуля комп’ютерного зору

Після перевірки службових параметрів виконується попередня обробка зображення. На цьому етапі система:

– виконує масштабування кадру до розміру, необхідного для роботи нейронної мережі;

– нормалізує значення пікселів;

– виконує усунення шумів;

– підвищує контрастність зображення;

– формує тензори для подальшого інференсу.

Якщо зображення не відповідає мінімальним вимогам якості, система може:

– відхилити кадр;

– попередити оператора про недостатню якість;

– автоматично виконати спробу корекції зображення.

Після завершення попередньої обробки дані передаються до модуля нейронної мережі. У процесі інференсу модель комп’ютерного зору виконує:

– аналіз вхідного кадру;

– локалізацію об’єктів;

– побудову bounding box;

– класифікацію цілей;

– оцінювання рівня впевненості розпізнавання.

У разі виявлення об’єктів система формує структурований набір результатів, який містить:

– клас об’єкта;

– координати об’єкта на зображенні;

– рівень впевненості;

– розміри області детекції;

– часову мітку;

– координати БПЛА на момент отримання кадру.

Отримані результати передаються до підсистеми формування звітів та збереження даних. На цьому етапі система виконує:

– запис результатів до бази даних;

– прив’язку результатів до конкретного кадру;

– збереження інформації про використану модель;

– формування журналу подій;

– накопичення статистики роботи системи.

Після завершення аналізу результати відображаються у користувацькому інтерфейсі оператора. У вікні результатів можуть відображатися:

– вихідне зображення;

– межі виявлених об’єктів;

– типи розпізнаних цілей;

– параметри детекції;

– рівень впевненості моделі;

– службова інформація про кадр.

3.5 Проектування програмних вузлів взаємодії

3.5.1 Проектування програмних вузлів взаємодії

Модель XRCE-DDS як проміжне ПЗ, основою взаємодії між польотним стеком PX4 та екосистемою ROS 2 є проміжне ПЗ стандарту Micro XRCE-DDS. Архітектура побудована на моделі «клієнт-агент»:

– XRCE-DDS Client: Функціонує безпосередньо на польотному контролері. Його завданням є серіалізація даних внутрішніх повідомлень дрона та їх передача через фізичний транспортний рівень;

– XRCE-DDS Agent: Працює на супровідному комп’ютері. Він виступає як брокер повідомлень, що отримує дані від клієнта та транслює їх у повноцінні об’єкти DDS.

Така структура дозволяє вузлам ROS 2 безпосередньо бачити стан сенсорів, польотні дані та статус систем через стандартну модель публікація-підписка, забезпечуючи безшовну інтеграцію низькорівневого обладнання з алгоритмами високого рівня.

Проектування вузла розпізнавання, спроектований як основний компонент обробки візуальної інформації. Він отримує відеопотік з камери, проводить інференс нейромережі та виконує роль публікатора. Після обробки кожного кадру вузол формує повідомлення, що містить координати виявлених об’єктів у тривимірному просторі або відносно кадру, і публікує їх у спеціалізований топік. Даний підхід забезпечує модульність: інші вузли можуть використовувати ці дані, не знаючи про архітектуру нейромережі чи модель камери.

Проектування вузла управління, відповідає за прийняття рішень та формування траєкторії руху. Він реалізує логіку на основі наступних кроків:

– вузол підписується на результати розпізнавання від вузла розпізнавання;

– на основі отриманих координат та поточного стану дрона розраховується необхідний вектор руху;

– вузол перетворює логічні рішення у команди TrajectorySetpoint, для позиціонування, та VehicleCommand, для зміни польотних режимів або керування корисним навантаженням, які відправляються назад у PX4 через міст uXRCE-DDS.

Роль протоколу MAVLink у системній архітектурі. Попри використання сучасного DDS-моста, протокол MAVLink залишається критично важливим компонентом системи, створюючи паралельний канал зв’язку. Основні функції MAVLink у даному проекті:

– передача даних про стан системи на наземну станцію керування через радіоканал;

– забезпечення можливості миттєвого перехоплення управління оператором у разі збою алгоритмів нейромережі або програмної помилки у вузлах ROS 2. Це гарантує детерміновану взаємодію з апаратною частиною навіть при відмові супровідного комп’ютера.

Налаштування якості обслуговування, для забезпечення стабільності системи та ефективного використання пропускної здатності каналів зв’язку, налаштовано різні профілі якості обслуговування:

– Best Effort, який застосовується для передачі відеопотоку та високочастотної телеметрії. У цих випадках актуальність даних важливіша за

гарантовану доставку – втрата одного кадру чи повідомлення про стан не є критичною, оскільки наступне надійде через мілісекунди;

– Reliable, який використовується для критичних команд управління, повідомлень про зміну режимів польоту та системних конфігурацій. Це гарантує, що кожна команда буде отримана та підтверджена, що є обов’язковим для безпечного функціонування безпілотної системи.

3.5.2 Розробка структури програмних вузлів для управління БПЛА

Програмна архітектура системи управління БПЛА базується на децентралізованому підході ROS 2, де функціональні модулі представлені як слабопов’язані вузли, що формують єдиний граф системи. Взаємодія між вузлами реалізується через рівень абстракції ROS Middleware, який використовує протокол DDS для забезпечення детермінованої передачі даних у реальному часі.

Основним завданням вузла захоплення відео є ініціалізація інтерфейсу камери та захоплення кадрів з заданою частотою. Вузол виконує конвертацію сирих байтових масивів у стандартизовані повідомлення формату sensor_msgs/msg/Image. Конфігурація QoS, для мінімізації затримок та запобігання накопиченню застарілих даних у черзі публікації використовується QoS-профіль. Це гарантує, що система обробляє лише найактуальніший кадр, ігноруючи втрачені пакети, що є критичним для контурів управління з високою частотою оновлення.

Вузол інтелектуального аналізу виступає основним елементом системи сприйняття. Вузол підписується на топік з відеопотоком. При отриманні повідомлення активується зворотний зв’язок, яка передає зображення на вхід нейромережевої моделі для виконання інференсу.

Результати детекції, координати обмежувальних рамок, класи об’єктів та рівень впевненості, серіалізуються у кастомний топік detected_objects. Для підвищення продуктивності на бортових ЕОМ інференс виконується з використанням апаратного прискорення через середовище виконання TensorRT.

Вузол реалізує високорівневу логіку польоту в режимі Offboard, транслюючи візуальні дані у команди руху.

На основі координат об’єкта, отриманих з топіка detected_objects, обчислюється траєкторія зближення або утримання дистанції.

Сформовані координати та вектори швидкості публікуються як повідомлення TrajectorySetpoint. Ці дані передаються до польотного контролера через міст uXRCE-DDS, який забезпечує ефективний маппінг між внутрішніми повідомленнями автопілота та графом ROS 2.

Для забезпечення паралельної роботи вузлів на багатоядерних процесорах бортових комп’ютерів використовується MultiThreadedExecutor. Розподіл завдань здійснюється через групи зворотних викликів. Зокрема, інтенсивні обчислення та критичні для реального часу завдання виділяються у Mutually Exclusive Callback Groups, що дозволяє виконавцю призначати їх на окремі потоки процесора. Такий підхід у поєднанні з налаштуваннями пріоритетів потоків у ядрі PREEMPT-RT гарантує передбачуваність виконання циклів управління та стабільність частоти публікації команд навіть при високому навантаженні на систему.

3.6 Проектування алгоритму фільтрації та стабілізації координат об’єктів

Використання нейромережевих детекторів сімейства YOLO для визначення координат об’єктів у реальному часі характеризується дискретністю виводу та чутливістю до варіацій освітлення. Крім того, високочастотні вібрації рами БПЛА та динаміка його руху вносять значний рівень шуму у виміряні координати центру об’єкта. Без попередньої обробки ці шуми призводять до виникнення паразитних осциляцій у контурі ПІД-регулятора, що негативно впливає на стабільність утримання цілі та загальну динаміку польоту. Для забезпечення плавності керування та мінімізації похибки вимірювання впроваджено алгоритм на основі лінійного фільтра Калмана.

Стан динамічного об’єкта в площині кадру описується вектором стану x, що включає просторові координати та швидкісні характеристики:

$$x = [u, v, \dot{u}, \dot{v}, w, h]^T, \quad (3.1)$$

де (u, v) – координати геометричного центру об'єкта, $(\dot{u}, \dot{v})$ – швидкості зміни відповідних координат, (w, h) – ширина та висота обмежувальної рамки.

Використання швидкостей у векторі стану дозволяє враховувати інерційність руху та здійснювати точне прогнозування.

Алгоритм Фільтра Калмана.

Реалізація алгоритму базується на рекурсивному оцінюванні стану системи через два послідовні етапи:

– етап екстраполяції, де на основі попереднього оціненого стану $x_k - 1$ розраховується апріорна оцінка стану на поточному кроці k;

– етап корекції, де після отримання нових даних від детектора YOLO формується вектор вимірювань z_k.

У випадках короткочасного перекриття об'єкта або збоїв у роботі детектора, алгоритм переходить у режим чистої екстраполяції. Система продовжує оновлювати вектор стану , базуючись лише на рівняннях прогнозу. Це дозволяє уникнути різких стрибків керуючого сигналу та забезпечує безперервність траєкторії супроводження цілі БПЛА навіть за відсутності прямого візуального контакту на коротких інтервалах.

Алгоритм інтегрований у програмну архітектуру як окремий вузол системи ROS 2. Для обчислень використовуються спеціалізовані бібліотеки, що забезпечують необхідну обчислювальну потужність для роботи в реальному часі. Вузол приймає повідомлення з координатами від детектора YOLO та публікує стабілізований вектор стану для модуля ПІД-регулятора.

Для забезпечення високої надійності функціонування БПЛА в режимі Offboard логіка управління реалізується на основі детермінованого скінченного автомата. Такий архітектурний підхід дозволяє чітко відокремити високорівневу логіку місії від низькорівневих алгоритмів навігації, що запобігає виникненню конфліктів у потоках команд керування. Кожен стан системи є взаємовиключним, а переходи між ними регламентуються суворими логічними умовами, що

забезпечує передбачувану поведінку автономної системи в динамічному середовищі,.

Логіка системи розподілена між наступними функціональними станами:

– STATE_IDLE – початковий стан очікування, де обробник подій блокує виконання польотних команд до моменту ініціалізації прапора активації режиму Offboard та підтвердження готовності сенсорної підсистеми;

– STATE_SEARCH – стан автономного пошуку, де система виконує політ за заданою траєкторією або утримує позицію, одночасно з цим модуль візуального сприйняття на базі нейромережі YOLO проводить безперервне сканування відеопотоку;

– STATE_TRACKING – стан активного супроводження цілі, де формуються вектори руху на основі розрахунків ПІД-регулятора для центрування об’єкта в кадрі та підтримки заданої дистанції;

– STATE_LOST_TARGET – спеціалізований стан обробки втрати візуального контакту, де при зникненні детекції обробник подій ініціалізує алгоритм утримання позиції або повернення до точки останньої успішної фіксації об'єкта для відновлення візуальних даних.

Програмна реалізація детермінованого скінченого автомата виконана у формі окремого вузла екосистеми ROS 2. Логіка базується на використанні класу StateMachine або спеціалізованих інструментів типу SMACH, що дозволяє асинхронно обробляти вхідні топіки від нейромережі та публікувати цільові значення для польотного контролера. Дана структура забезпечує модульність: зміна логіки одного стану не впливає на стабільність роботи інших компонентів системи.

Для запобігання аварійним ситуаціям передбачено стан STATE_EMERGENCY, що виконує роль механізму Fail-safe. Цей стан активується автоматично в разі втрати зв'язку з сервером обробки ШІ або припинення надходження даних від основних сенсорів. При переході в цей стан обробник подій ініціалізує зміну прапора стану на аварійний, що призводить до негайної зупинки

місії та переходу системи в режим безпечної посадки або повернення на точку старту.

У межах розробки архітектури було спроектовано систему взаємодії між польотним контролером Pixhawk та бортовим комп’ютером на базі проміжного програмного забезпечення uXRCE-DDS. Архітектурне рішення базується на топології Client-Agent. Програмний вузол Client інтегровано безпосередньо у прошивку PX4, що функціонує під управлінням RTOS NuttX, тоді як Agent розгорнуто в середовищі Linux на бортовому комп'ютері.

Для фізичного сполучення вузлів було обрано інтерфейс UART, оскільки він забезпечує стабільне з'єднання "точка-точка" з мінімальними апаратними затримками без необхідності керування мережевим стеком. Використання uXRCE-DDS замість застарілого MAVROS дозволило мінімізувати обчислювальні накладні витрати на серіалізацію та забезпечити нативну інтеграцію внутрішніх повідомлень uORB у мережу ROS 2.

Для забезпечення коректного обміну даними було сформовано схему відображення внутрішніх структур повідомлень PX4 у відповідні топіки ROS 2. Процес мапінгу вимагає ідентичності визначень повідомлень у обох системах.

Незважаючи на використання uXRCE-DDS для високорівневих обчислень, у системі було збережено та інтегровано протокол MAVLink. Це рішення обґрунтоване необхідністю забезпечення сумісності з наземними станціями управління, такими як QGroundControl, та трансляції телеметрії через радіоканал.

Потік телеметрії через MAVLink спроектовано як паралельний канал, що функціонує через окремий серійний інтерфейс.

3.7 Висновки до розділу 3

У третьому розділі виконано проєктування програмного та інформаційного забезпечення кіберфізичної системи управління квадрокоптерами для розпізнавання об’єктів із використанням технологій машинного навчання. У межах розділу сформовано функціональну структуру системи, визначено основні сценарії

взаємодії користувача з модулем комп’ютерного зору та розроблено архітектуру програмних компонентів.

Під час проєктування системи було проведено функціональний аналіз предметного середовища та визначено основні підсистеми, необхідні для забезпечення процесів обробки відеоданих, навчання нейронної мережі, класифікації об’єктів та збереження результатів аналізу. Побудовано дерево функцій та розроблено IDEF0-моделі, які дозволили формалізувати логіку роботи програмного комплексу та описати взаємодію між його основними компонентами.

У процесі дослідження сформовано концептуальну, логічну та фізичну моделі бази даних системи. Визначено структуру таблиць для зберігання параметрів об’єктів, характеристик техніки, результатів класифікації, моделей нейронних мереж, зображень та метаданих.

Використання бази даних SQLite дозволило реалізувати компактне та ефективне сховище даних, яке підтримує швидкий доступ до результатів роботи системи та забезпечує інтеграцію між окремими програмними модулями.

У межах розділу також було спроєктовано користувацький інтерфейс системи. Розроблено прототипи основних вікон програмного забезпечення, які забезпечують: завантаження зображень, запуск процесу розпізнавання, перегляд результатів класифікації, адміністрування класів об’єктів, роботу з навчальним датасетом.

Окрему увагу приділено проєктуванню алгоритмів попередньої обробки зображень та організації процесу навчання нейронної мережі.

У процесі дослідження сформовано архітектуру згорткової нейронної мережі та виконано підбір оптимальних параметрів моделі із використанням GridSearch та Keras Tuner.

4 ПРОТОТИПУВАННЯ, РОЗРОБКА, ОПИС І ТЕСТУВАННЯ РОЗРОБЛЕНИХ МОДУЛІВ КІБЕРФІЗИЧНОЇ СИСТЕМИ УПРАВЛІННЯ КВАДРОКОПТЕРАМИ ДЛЯ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ЗА ДОПОМОГОЮ МАШИННОГО НАВЧАННЯ

4.1. Проєктування та прототипування інтерфейсу кіберфізичної системи

Одним із важливих етапів розробки кіберфізичної системи управління квадрокоптерами є створення інтерфейсу взаємодії оператора з модулем комп’ютерного зору. Інтерфейс системи повинен забезпечувати швидкий доступ до функцій завантаження відеоданих, запуску процесів розпізнавання об’єктів, перегляду результатів аналізу та адміністрування навчальних даних.

Під час проєктування інтерфейсу основна увага приділялася:

– мінімізації часу взаємодії оператора із системою;

– забезпеченню зручності роботи в умовах реального часу;

– підтримці швидкого запуску процесів аналізу відеопотоку;

– можливості інтеграції з підсистемами машинного навчання;

– підтримці масштабування функціональних можливостей системи.

Для перевірки ергономіки інтерфейсу та логіки взаємодії між компонентами було розроблено набір прототипів основних вікон системи:

– вікно завантаження та обробки відеоданих;

– вікно відображення результатів розпізнавання;

– вікно керування параметрами класів об’єктів.

На рис. 4.1 наведено прототип головного вікна завантаження даних до системи.

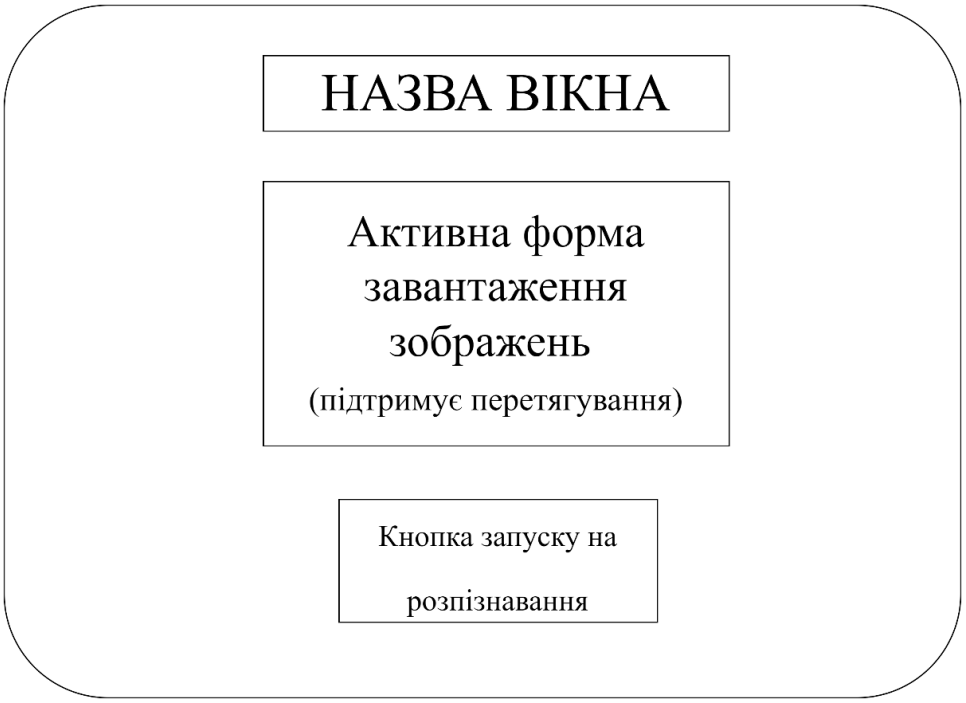


Рисунок 4.1 – Прототип вікна завантаження відеоданих

Даний інтерфейс призначений для завантаження кадрів аерозйомки або окремих зображень до модуля комп'ютерного зору. У вікні передбачено область активного завантаження файлів із підтримкою механізму Drag-and-Drop, що дозволяє оператору швидко передавати дані для подальшого аналізу.

Після завантаження даних оператор може ініціювати запуск алгоритмів розпізнавання за допомогою окремої кнопки керування [85]. Такий підхід спрощує роботу із системою під час виконання задач моніторингу та аналізу обстановки.

На рис. 4.2 представлено прототип вікна результатів роботи модуля комп'ютерного зору.

У даному вікні відображаються:

– завантажене зображення або кадр відеопотоку;

– результати детекції об'єктів;

– параметри розпізнавання;

– інформація про класи виявлених цілей.

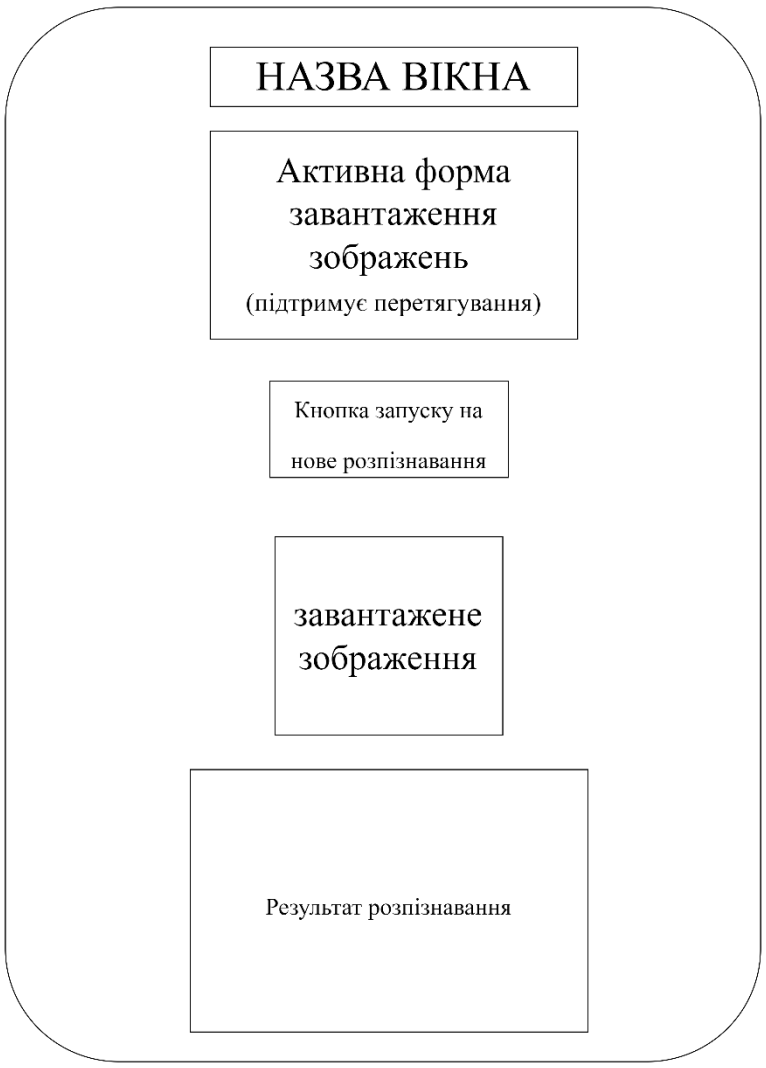


Рисунок 4.2 – Прототип вікна результатів розпізнавання об’єктів

Інтерфейс передбачає можливість повторного запуску процесу аналізу, що дозволяє оператору виконувати серійне тестування роботи моделі або аналізувати декілька наборів даних.

На рис. 4.3 наведено прототип вікна адміністрування класів об’єктів та параметрів навчального датасету. Дане вікно використовується для формування та редагування інформації про класи об’єктів, що застосовуються під час навчання моделей машинного навчання.

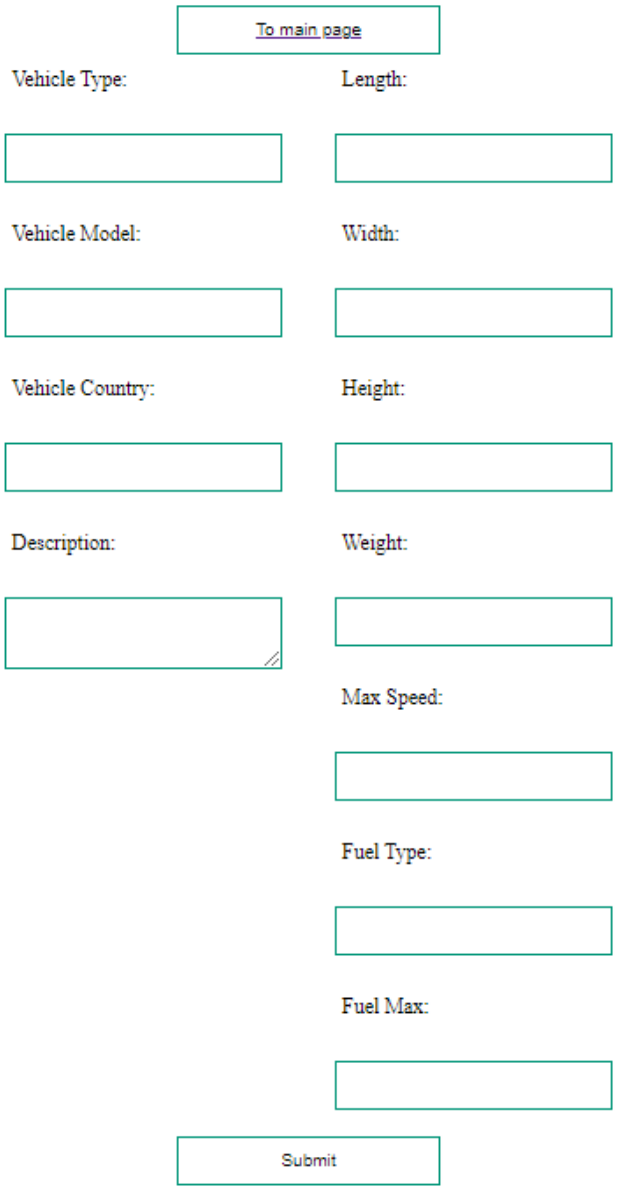


Рисунок 4.3 – Прототип вікна керування параметрами класів об’єктів

У структурі інтерфейсу передбачено поля для введення:

– типу об’єкта;

– моделі техніки;

– країни виробника;

– геометричних параметрів;

– масогабаритних характеристик;

– параметрів швидкості;

– додаткових характеристик об’єкта.

Використання такого інтерфейсу дозволяє централізовано керувати параметрами навчальних даних та забезпечує спрощення процесу формування датасетів для перенавчання нейронної мережі.

Розроблені прототипи стали основою для подальшої реалізації користувацького інтерфейсу кіберфізичної системи та дозволили оцінити зручність взаємодії оператора з модулем комп'ютерного зору ще до етапу програмної реалізації.

4.2. Навчання та налаштування нейронної мережі

Одним із ключових етапів реалізації кіберфізичної системи управління квадрокоптерами є навчання моделі комп'ютерного зору для автоматизованого виявлення та класифікації об'єктів у відеопотоці. Для побудови та тренування нейронної мережі використано бібліотеки TensorFlow, Keras та OpenCV, які забезпечують підтримку алгоритмів глибокого навчання та засобів обробки зображень.

На першому етапі виконувалося формування навчального датасету. До набору даних входили кадри аерозйомки з різними типами об'єктів, отримані з відкритих джерел та підготовлені для використання в задачах машинного навчання. Перед початком тренування всі зображення проходили процедуру попередньої перевірки та структуризації за класами.

Система автоматично формувала тренувальні, валідаційні та тестові вибірки. Приклад процесу завантаження датасету наведено на рис. 4.4.

```
Found 6672 files belonging to 8 classes.
Using 5338 files for training.
Found 6672 files belonging to 8 classes.
Using 1334 files for validation.
Found 747 files belonging to 8 classes.
```

Рисунок 4.4 – Формування тренувальної та валідаційної вибірки

У процесі формування датасету було використано 7419 зображень, що містили різні типи військової техніки та об’єктів, отриманих із матеріалів аерозйомки. Для забезпечення коректності навчання нейронної мережі набір даних було розділено на тренувальну, валідаційну та тестову вибірки.

Як тестову вибірку виділено 10% від загального набору даних, що становить 747 зображень. Дані зображення були випадковим чином вибрані із загального масиву та не використовувалися під час навчання моделі. Тестова вибірка застосовувалась для фінальної оцінки здатності нейронної мережі узагальнювати результати на нових даних.

Решта 90% датасету, що складає 6672 зображення, була розділена на:

– тренувальну вибірку – 5338 зображень (72%);

– валідаційну вибірку – 1334 зображення (18%).

Тренувальна вибірка використовувалась для безпосереднього налаштування ваг нейронної мережі, тоді як валідаційна вибірка застосовувалась для контролю процесу навчання, оцінювання точності моделі після кожної епохи та запобігання перенавчанню.

Після завершення формування вибірок система виконувала класифікацію об’єктів за визначеними класами техніки. До складу датасету входили зображення різних типів військових об’єктів, серед яких:

– танк Т-62;

– артилерійська установка 2С1;

– спеціалізована техніка Д7;

– вантажний автомобіль ЗІЛ-131;

– бронетранспортер БТР-60.

Після завершення підготовки даних виконувалось налаштування архітектури нейронної мережі. У роботі використовувалась згорткова нейронна мережа (CNN), побудована засобами Keras. Архітектура моделі включала: вхідний шар, шари масштабування, згорткові шари Conv2D, шари субдискретизації MaxPooling2D, повнозв’язні шари Dense, вихідний шар класифікації Softmax.

Архітектуру нейронної мережі наведено на рисунку 4.5.

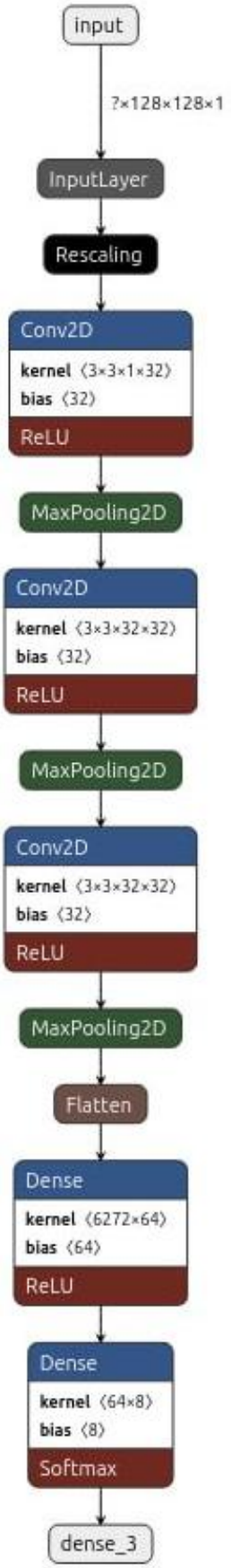


Рисунок 4.5 – Архітектура згорткової нейронної мережі

Вхідний шар (InputLayer) забезпечує прийом зображень фіксованого розміру та формує структуру вхідних даних для подальшої обробки.

Шар Rescaling виконує нормалізацію значень пікселів, приводячи їх до діапазону, оптимального для роботи моделі. Це дозволяє стабілізувати процес навчання та зменшити вплив різниці в освітленні кадрів.

Згорткові шари Conv2D виконують автоматичне виділення ознак із вхідних зображень. На початкових рівнях мережі виявляються:

– контури;

– текстури;

– переходи яскравості;

– локальні особливості.

На глибших рівнях мережі формуються складні ознаки:

– геометрія об’єктів;

– характерні елементи техніки;

– структурні особливості цілей.

Для активації нейронів використовувалась функція ReLU, яка дозволяє пришвидшити процес збіжності моделі та зменшити проблему зникнення градієнтів.

Шари MaxPooling2D використовувались для:

– зменшення розмірності даних;

– скорочення обчислювальної складності;

– підвищення інваріантності до незначних зміщень та масштабування об’єктів.

Після завершення згорткової обробки шар Flatten виконував перетворення багатовимірного представлення ознак у одновимірний вектор, який надалі передавався до повнозв’язних шарів.

Повнозв’язний шар Dense із функцією активації ReLU використовувався для формування високорівневих залежностей між ознаками, отриманими під час згорткової обробки.

Вихідний шар Dense із функцією Softmax забезпечував остаточну класифікацію об’єктів за визначеними класами. Кількість нейронів вихідного шару відповідала кількості класів у навчальному датасеті.

Для навчання мережі використовувались:

– оптимізатор Adam;

– функція втрат categorical crossentropy;

– механізм ранньої зупинки;

– контроль перенавчання;

– автоматичне збереження ваг моделі.

Під час навчання система виконувала постійний моніторинг:

– точності класифікації;

– значення функції втрат;

– швидкості збіжності;

– стабільності процесу тренування.

Після завершення навчання сформована модель зберігалась у файловій системі та інтегрувалась до модуля комп’ютерного зору кіберфізичної системи.

Отримана архітектура забезпечила достатню швидкість обробки кадрів та дозволила реалізувати розпізнавання об’єктів у режимі реального часу, що є критично важливим для задач автономного управління квадрокоптерами.

4.2.1 Оцінювання ефективності навченої моделі

Для оцінювання ефективності навченої нейронної мережі та перевірки її здатності до узагальнення було використано бібліотеку Scikit-learn (sklearn), яка містить набір інструментів для аналізу якості моделей машинного навчання. Однією з переваг даної бібліотеки є підтримка механізмів перехресної перевірки (cross-validation), що дозволяє отримати більш об’єктивну оцінку точності моделі.

У роботі застосовано метод K-Fold Cross Validation із кратністю 5. Даний підхід передбачає багаторазове розбиття датасету на тренувальні та валідаційні підвибірки. На кожній ітерації одна частина вибірки використовується для

перевірки якості моделі, тоді як решта використовується для навчання. Після завершення всіх ітерацій результати усереднюються.

Використання K-Fold Cross Validation дозволило:

– зменшити ризик випадкового переоцінювання точності;

– оцінити стабільність моделі;

– перевірити здатність нейронної мережі працювати з новими даними;

– знизити ризик перенавчання.

У результаті проведених експериментів було отримано такі показники:

– точність валідації для валідаційної вибірки із 1334 зображень – 0.987;

– середня точність при п’ятикратній перехресній перевірці – 0.986.

Отримані результати свідчать про високу ефективність сформованої архітектури нейронної мережі та стабільність її роботи на різних підвибірках даних.

Приклади зображень із навчального датасету наведено на рис. 4.6.

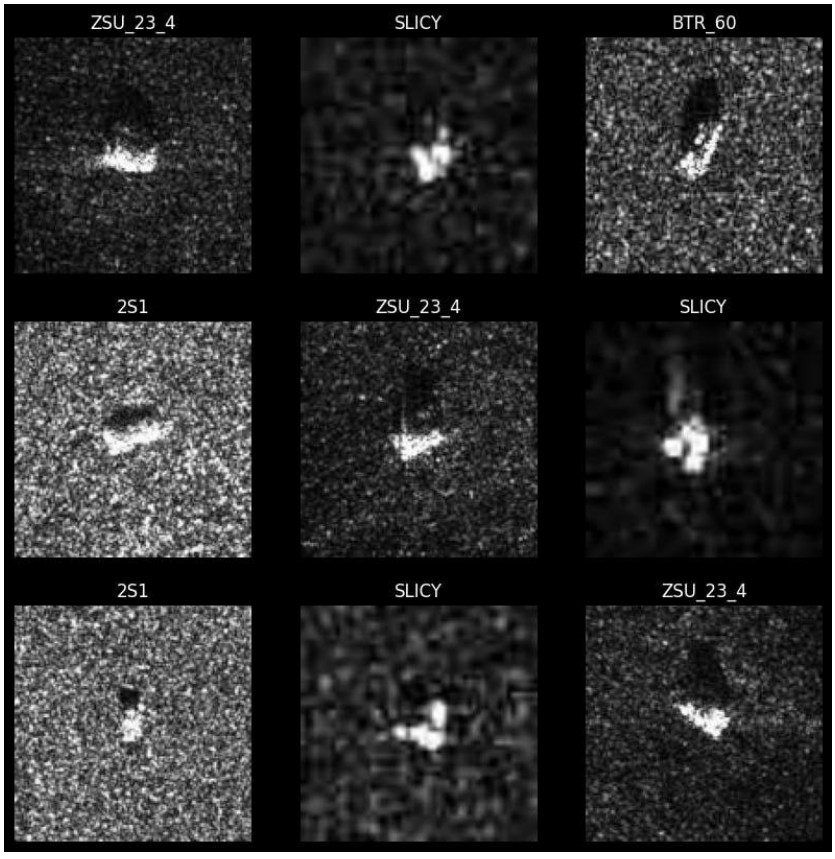


Рисунок 4.6 – Приклади зображень військової техніки з навчального датасету

Представлені зображення містять приклади кадрів аерозйомки різних типів військової техніки, які використовувалися під час навчання моделі. На зображеннях можна побачити об’єкти різних класів, зокрема:

– бронетранспортери;

– артилерійські установки;

– спеціалізовану техніку;

– транспортні засоби військового призначення.

Частина зображень містить значний рівень шуму та низьку контрастність, що характерно для матеріалів аерозйомки, отриманих у складних умовах спостереження. Це дозволило підвищити стійкість моделі до реальних умов експлуатації та забезпечити кращу адаптацію до аналізу відеопотоку з бортових камер БПЛА.

На рисунку 4.7 наведено структуру навченої моделі нейронної мережі.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 rescaling_1 (Rescaling)     (None, 128, 128, 1)       0

 conv2d_3 (Conv2D)           (None, 126, 126, 32)      320

 max_pooling2d_3 (MaxPooling  (None, 63, 63, 32)       0
 2D)

 conv2d_4 (Conv2D)           (None, 61, 61, 32)        9248

 max_pooling2d_4 (MaxPooling  (None, 30, 30, 32)       0
 2D)

 conv2d_5 (Conv2D)           (None, 28, 28, 32)        9248

 max_pooling2d_5 (MaxPooling  (None, 14, 14, 32)       0
 2D)

 flatten_1 (Flatten)         (None, 6272)              0

 dense_2 (Dense)             (None, 64)                401472

 dense_3 (Dense)             (None, 8)                 520

=================================================================
Total params: 420,808
Trainable params: 420,808
Non-trainable params: 0
_________________________________________________________________
```

Рисунок 4.7 – Структура та параметри нейронної мережі

Представлена архітектура містить три згорткові блоки Conv2D, між якими розташовані шари субдискретизації MaxPooling2D. Така структура дозволяє поступово зменшувати просторову розмірність ознак і водночас формувати високорівневе представлення об’єктів.

У процесі проходження даних через мережу:

– шари Conv2D виконують виділення ознак;

– шари MaxPooling2D зменшують розмірність даних;

– шар Flatten формує одновимірне представлення ознак;

– повнозв’язні шари Dense виконують остаточну класифікацію.

Загальна кількість параметрів моделі становить понад 420 тисяч параметрів, що забезпечує достатню складність архітектури для задач класифікації об’єктів на зображеннях аерозйомки.

Результатом навчання стало формування набору оптимізованих вагових коефіцієнтів нейронної мережі, які дозволяють моделі виконувати класифікацію військової техніки з високою точністю та стабільністю. Отримена модель інтегрована до складу кіберфізичної системи управління квадрокоптерами та використовується для автоматизованого аналізу відеопотоку в режимі реального часу.

4.2.2 Оцінювання результатів навчання нейронної мережі

Після завершення процесу навчання виконувалось комплексне оцінювання ефективності сформованої моделі комп’ютерного зору. Аналіз результатів проводився із використанням набору метрик, які дозволяють оцінити не лише загальну точність класифікації, але й стабільність роботи нейронної мережі, її здатність до узагальнення та стійкість до помилок.

Для оцінювання якості моделі використовувались такі метрики:

– Loss;

– Categorical Accuracy;

– MCC;

– F2-score;

– AUC (Area Under Curve);

– PRC (Precision-Recall Curve).

Інформацію про результати навчання нейронної мережі наведено на рис. 4.8.

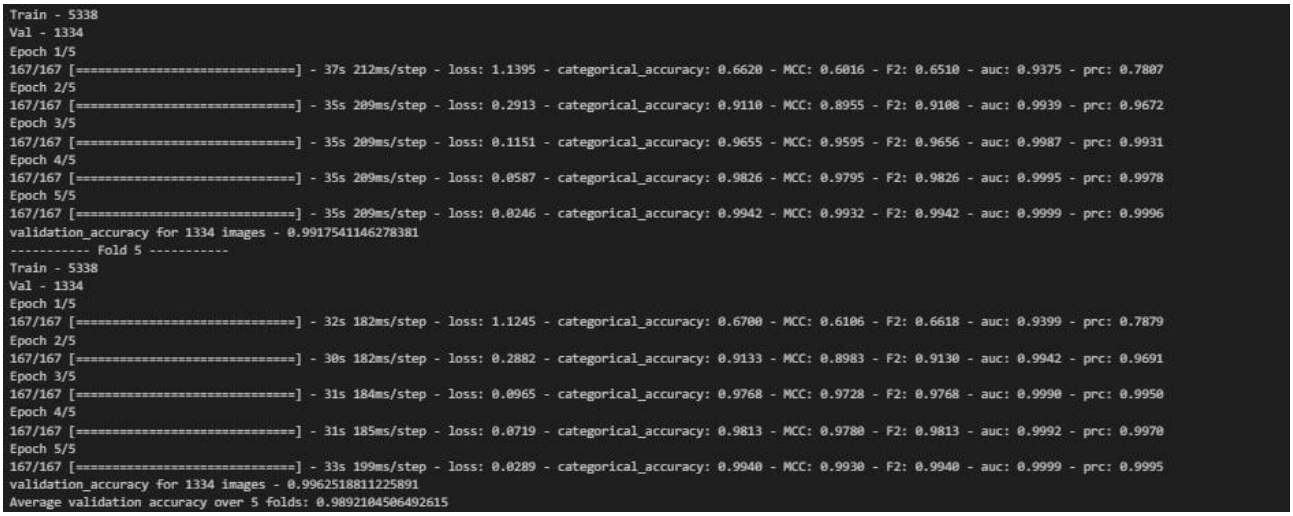


Рисунок 4.8 – Результати навчання нейронної мережі

У процесі навчання система виконувала оцінювання метрик як на тренувальній, так і на валідаційній вибірках. Це дозволило контролювати стабільність процесу навчання та своєчасно виявляти ознаки перенавчання.

Для оцінювання правильності класифікації окремих класів було сформовано матрицю помилок (Confusion Matrix), яка наведена на рис. 4.9.

Матриця помилок дозволяє оцінити:

– кількість правильно класифікованих об’єктів;

– кількість помилкових класифікацій;

– найбільш проблемні класи;

– частоту плутання між окремими типами техніки.

Основна кількість значень зосереджена на головній діагоналі матриці, що свідчить про високий рівень правильної класифікації об’єктів.

Наявність незначної кількості помилкових спрацьовувань пояснюється:

– низькою якістю окремих кадрів;

– високим рівнем шуму;

– схожістю контурів окремих типів техніки;

– складними умовами аерозйомки.

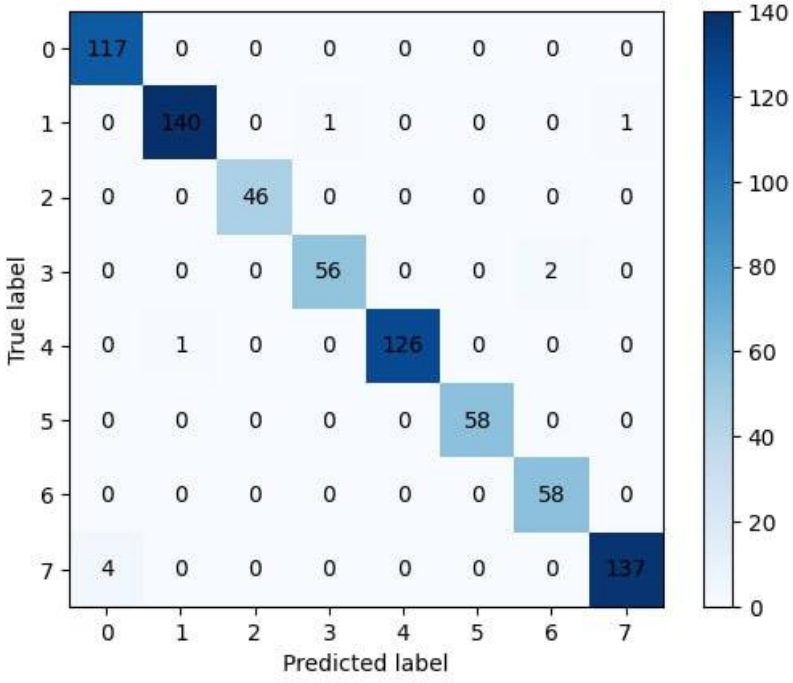


Рисунок 4.9 – Матриця помилок класифікації

Після завершення навчання виконувалось тестування моделі на даних, які не використовувались у процесі тренування. Результати оцінювання наведено на рис. 4.10.

```
24/24 [==============================] - 1s 47ms/step - loss: 0.0541 - categorical_accuracy: 0.9880 - MCC: 0.9859 - F2: 0.9879 - auc: 0.9984 - prc: 0.9970

{'loss': 0.054056040942668915,
 'categorical_accuracy': 0.9879518151283264,
 'MCC': 0.9859123229980469,
 'F2': 0.9879298806190491,
 'auc': 0.9984369277954102,
 'prc': 0.9969944357872009}
```

Рисунок 4.10 – Результати тестування моделі на нових даних

За результатами тестування було отримано:

– точність класифікації близько 98%;

– низький рівень функції втрат;

– високі значення MCC, AUC та F2-score.

Отримані результати підтверджують ефективність сформованої архітектури нейронної мережі та можливість її використання у складі кіберфізичної системи управління квадрокоптерами.

Під час експериментів додатково контролювались ознаки перенавчання моделі. Аналізувались:

– різниця між тренувальною та валідаційною точністю;

– поведінка функції втрат;

– стабільність метрик на нових даних;

– складність архітектури моделі;

– кількість параметрів нейронної мережі.

У процесі дослідження не було виявлено критичних ознак перенавчання. Це свідчить про коректність вибору:

– архітектури моделі;

– гіперпараметрів;

– методів регуляризації;

– структури навчального датасету.

Отримена модель забезпечує достатню точність та стабільність роботи для використання в задачах автоматизованого аналізу відеопотоку з бортових систем БПЛА.

4.3 Опис користувацького інтерфейсу та результатів роботи програмного забезпечення

Для забезпечення взаємодії оператора з кіберфізичною системою управління квадрокоптерами реалізовано користувацький інтерфейс модуля комп’ютерного зору. Основною задачею інтерфейсу є забезпечення швидкого завантаження зображень або кадрів відеопотоку, запуску алгоритмів розпізнавання та відображення результатів аналізу у зручному для оператора вигляді.

Під час проєктування інтерфейсу враховувались такі вимоги:

– мінімізація кількості дій оператора;

– швидкий запуск процесу аналізу;

– підтримка інтерактивного завантаження даних;

– візуалізація результатів класифікації;

– можливість розширення бази класів об’єктів;

– інтеграція з інформаційною базою системи.

На рис. 4.11 наведено головне вікно завантаження зображень до системи.

Рисунок 4.11 – Інтерфейс завантаження зображень до модуля комп’ютерного зору

У центральній частині інтерфейсу розташована область активного завантаження файлів із підтримкою технології Drag-and-Drop. Оператор може:

– перетягнути файл до області завантаження;

– вибрати файл вручну;

– завантажити окремий кадр;

– передати зображення для аналізу.

Після завантаження кадру користувач може ініціювати запуск алгоритмів класифікації за допомогою кнопки Classify.

Після завершення роботи нейронної мережі система формує результати класифікації та відображає їх у користувацькому інтерфейсі. Приклад інтерфейсу відображення результатів наведено на рис. 4.12.

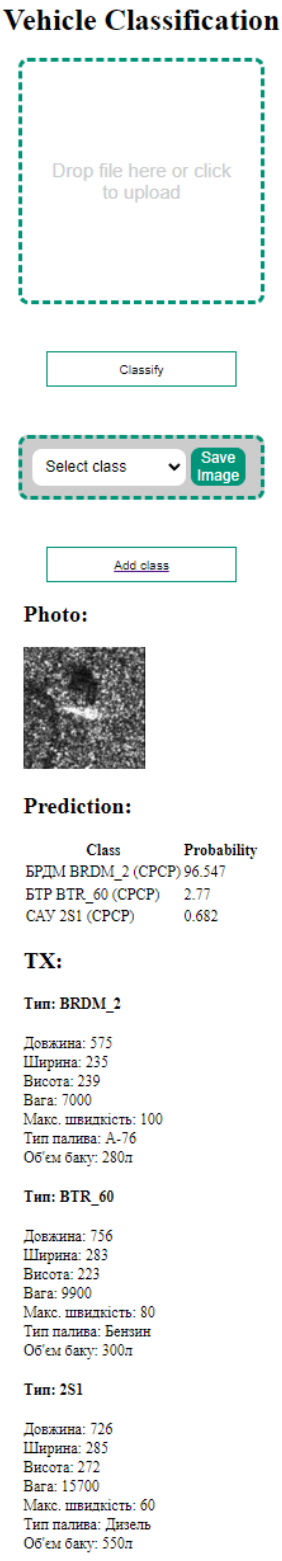


Рисунок 4.12 – Інтерфейс відображення результатів класифікації

У процесі роботи система:

– аналізує завантажене зображення;

– виконує класифікацію об’єкта;

– визначає найбільш імовірний клас;

– обчислює рівень впевненості;

– отримує характеристики об’єкта з бази даних.

У вікні результатів відображаються:

– завантажене зображення;

– список найбільш імовірних класів;

– ймовірність належності до кожного класу;

– технічні характеристики об’єкта;

– параметри класифікації.

Наприклад, після аналізу зображення система може визначити, що найбільш імовірним типом техніки є БРДМ-2 із відповідним рівнем впевненості моделі понад 96%.

Крім основного класу, система також відображає альтернативні результати класифікації із меншими значеннями ймовірності. Це дозволяє оператору оцінити ступінь впевненості моделі та проаналізувати можливі варіанти класифікації.

Після визначення класу система автоматично отримує з бази даних інформацію про характеристики об’єкта, серед яких:

– габаритні параметри;

– маса;

– максимальна швидкість;

– тип палива;

– об’єм паливної системи;

– додаткові характеристики.

Такий підхід дозволяє використовувати систему не лише для класифікації, а й для підтримки прийняття рішень під час аналізу обстановки.

Для забезпечення масштабованості системи реалізовано механізм додавання нових класів техніки та редагування параметрів існуючих об’єктів. Інтерфейс додавання нового класу наведено на рис. 4.13.

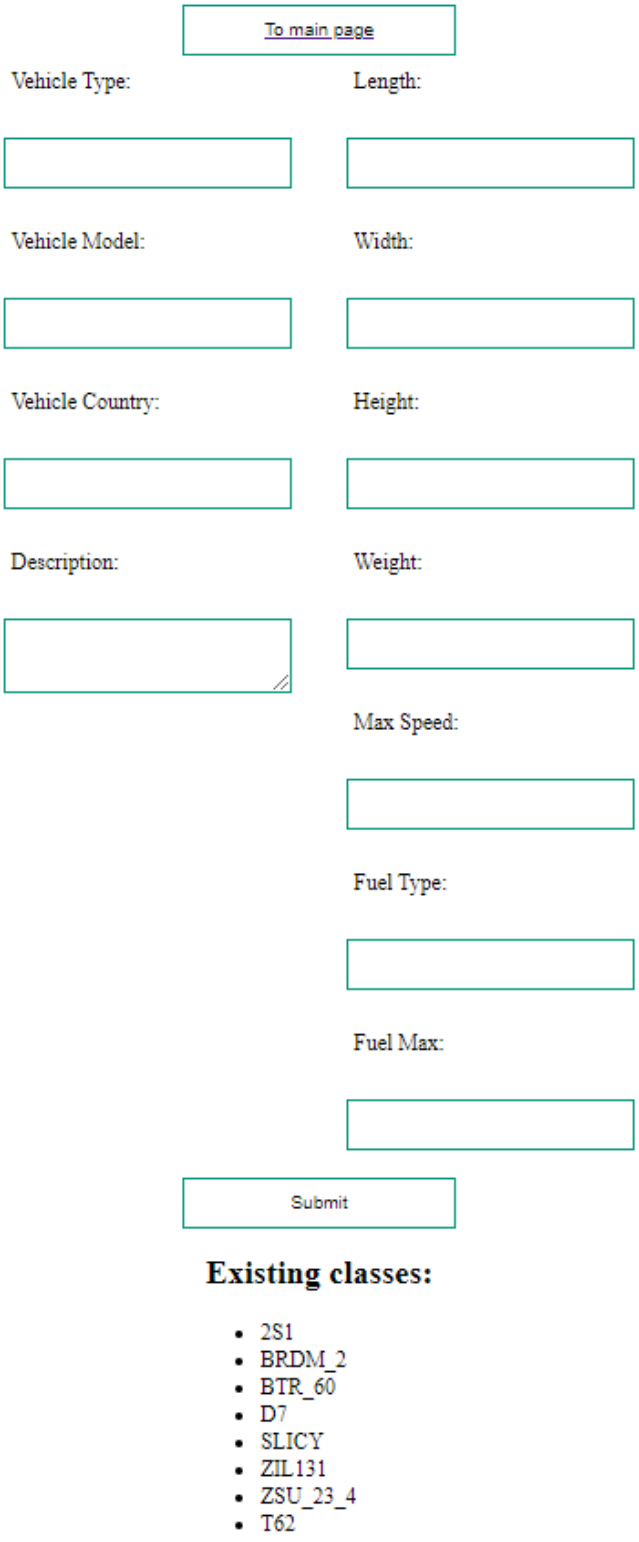


Рисунок 4.13 – Інтерфейс додавання нового класу об’єктів

У даному інтерфейсі оператор або адміністратор може:

– додавати нові класи техніки;

– задавати параметри об’єктів;

– формувати опис техніки;

– редагувати характеристики;

– оновлювати навчальний датасет.

Система підтримує зберігання:

– типу техніки;

– моделі;

– країни виробника;

– габаритних характеристик;

– маси;

– параметрів руху;

– описових характеристик.

Для збереження інформації використовується база даних SQLite, яка забезпечує:

– швидке збереження результатів;

– компактність;

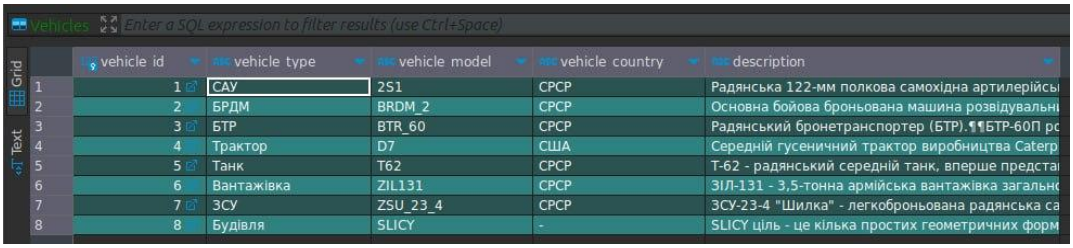
– простоту інтеграції;

– підтримку локальної роботи системи;

– можливість швидкого доступу до даних.

Використання SQLite дозволило реалізувати компактне локальне сховище для підтримки роботи модуля комп’ютерного зору та забезпечити інтеграцію між компонентами кіберфізичної системи.

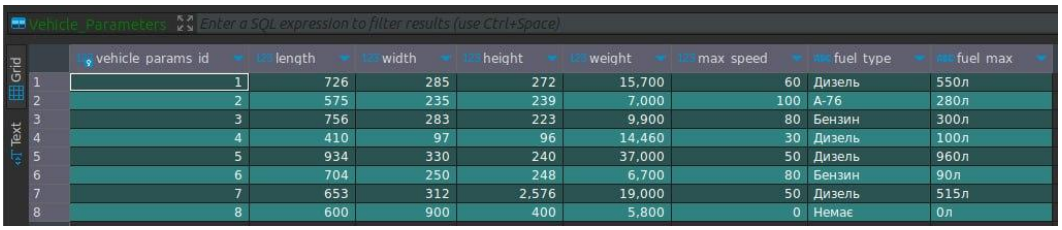
На рис. 4.14 наведено приклади збереження даних у таблицях бази даних, а розроблений користувацький інтерфейс забезпечує зручну взаємодію оператора із системою, підтримує швидкий запуск алгоритмів розпізнавання та дозволяє ефективно працювати з результатами класифікації об’єктів у режимі реального часу.



Vehicles — Enter a SQL expression to filter results (use Ctrl+Space)

	vehicle id	vehicle type	vehicle model	vehicle country	description
1	1	САУ	2S1	СРСР	Радянська 122-мм полкова самохідна артилерійсь
2	2	БРДМ	BRDM_2	СРСР	Основна бойова броньована машина розвідувальн
3	3	БТР	BTR_60	СРСР	Радянський бронетранспортер (БТР).¶¶БТР-60П ро
4	4	Трактор	D7	США	Середній гусеничний трактор виробництва Caterp
5	5	Танк	T62	СРСР	Т-62 - радянський середній танк, вперше предста
6	6	Вантажівка	ZIL131	СРСР	ЗІЛ-131 - 3,5-тонна армійська вантажівка загальн
7	7	ЗСУ	ZSU_23_4	СРСР	ЗСУ-23-4 "Шилка" - легкоброньована радянська са
8	8	Будівля	SLICY	-	SLICY ціль - це кілька простих геометричних форм

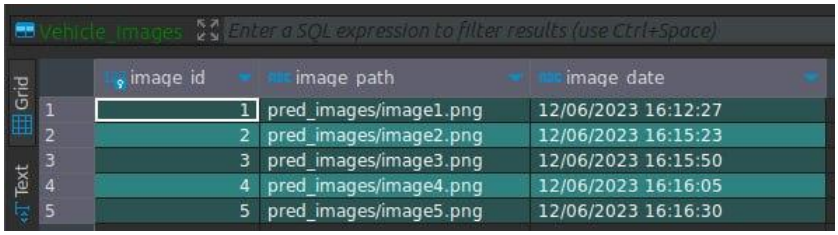
а)



Vehicle_Parameters — Enter a SQL expression to filter results (use Ctrl+Space)

	vehicle params id	length	width	height	weight	max speed	fuel type	fuel max
1	1	726	285	272	15,700	60	Дизель	550л
2	2	575	235	239	7,000	100	А-76	280л
3	3	756	283	223	9,900	80	Бензин	300л
4	4	410	97	96	14,460	30	Дизель	100л
5	5	934	330	240	37,000	50	Дизель	960л
6	6	704	250	248	6,700	80	Бензин	90л
7	7	653	312	2,576	19,000	50	Дизель	515л
8	8	600	900	400	5,800	0	Немає	0л

б)



Vehicle_Images — Enter a SQL expression to filter results (use Ctrl+Space)

	image id	image path	image date
1	1	pred_images/image1.png	12/06/2023 16:12:27
2	2	pred_images/image2.png	12/06/2023 16:15:23
3	3	pred_images/image3.png	12/06/2023 16:15:50
4	4	pred_images/image4.png	12/06/2023 16:16:05
5	5	pred_images/image5.png	12/06/2023 16:16:30

в)

Рисунок 4.14 – Приклади збереження даних у базі SQLite

4.4 Висновки до розділу 4

У четвертому розділі було виконано прототипування, програмну реалізацію, налаштування та тестування розроблених модулів кіберфізичної системи управління квадрокоптерами для розпізнавання об’єктів із використанням технологій машинного навчання. Основну увагу приділено практичній реалізації алгоритмів комп’ютерного зору, інтеграції нейронної мережі у програмний комплекс та оцінюванню ефективності роботи системи.

У межах розділу було розроблено користувацький інтерфейс програмного модуля, який забезпечує:

– завантаження зображень;

– запуск процесу класифікації;

– перегляд результатів розпізнавання;

– роботу з базою класів об’єктів;

– адміністрування параметрів техніки.

Реалізований інтерфейс дозволяє оператору виконувати взаємодію із системою в інтерактивному режимі та забезпечує швидкий доступ до результатів аналізу відеоданих.

У процесі реалізації системи було сформовано навчальний датасет, виконано його попередню обробку та організовано розподіл даних на тренувальні, валідаційні та тестові вибірки. Для підвищення якості роботи моделі використано методи аугментації даних та механізми автоматизованого підбору гіперпараметрів нейронної мережі.

У ході експериментів реалізовано та налаштовано згорткову нейронну мережу для задач класифікації об’єктів на зображеннях аерозйомки. Проведений підбір гіперпараметрів дозволив визначити оптимальну архітектуру моделі, що забезпечує баланс між точністю розпізнавання та швидкістю обробки даних.

Для оцінювання якості моделі використано комплекс метрик, серед яких:

– categorical accuracy;

– MCC;

– F2-score;

– AUC;

– PRC.

Результати тестування продемонстрували високі показники точності класифікації та стабільність роботи нейронної мережі на нових даних. Проведений аналіз графіків навчання та матриці помилок підтвердив відсутність критичних ознак перенавчання та достатню здатність моделі до узагальнення.

У процесі тестування було підтверджено працездатність:

– механізмів завантаження та обробки зображень;

– підсистеми класифікації;

– модуля взаємодії з базою даних;

– алгоритмів формування результатів;

– засобів відображення інформації оператору.

Також реалізовано механізми збереження результатів класифікації та параметрів об’єктів у базі даних SQLite, що забезпечило інтеграцію між програмними компонентами системи та підтримку накопичення результатів роботи моделі.

Отримані результати підтвердили можливість використання розробленого програмного комплексу у складі кіберфізичної системи управління квадрокоптерами для автоматизованого аналізу відеоданих і розпізнавання об’єктів у режимі реального часу. Практична реалізація модулів засвідчила ефективність використання методів глибокого навчання та комп’ютерного зору для задач автономного моніторингу й аеророзвідки.

ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальну науково-прикладну задачу розробки кіберфізичної системи управління квадрокоптерами для автоматизованого розпізнавання об’єктів із використанням технологій машинного навчання та комп’ютерного зору. Актуальність дослідження обумовлена необхідністю створення інтелектуальних систем аналізу відеоданих, здатних працювати в режимі реального часу в умовах обмежених обчислювальних ресурсів бортових платформ безпілотних літальних апаратів.

У результаті виконання роботи проведено комплексний аналіз сучасних підходів до побудови кіберфізичних систем БПЛА, методів комп’ютерного зору та моделей глибокого навчання для задач розпізнавання об’єктів. Встановлено, що найбільш перспективними для інтеграції у системи автономного керування є згорткові нейронні мережі та архітектури сімейства YOLO, які забезпечують поєднання високої швидкості обробки відеоданих та достатньої точності детекції.

У межах дослідження розроблено функціональну та структурну архітектуру кіберфізичної системи управління квадрокоптерами, яка об’єднує:

– підсистему отримання відеоданих;

– модуль попередньої обробки зображень;

– модуль комп’ютерного зору;

– систему збереження та аналізу результатів;

– підсистему підтримки прийняття рішень;

– компоненти взаємодії з оператором.

Побудовано математичну та алгоритмічну модель процесу розпізнавання об’єктів на кадрах аерозйомки. Реалізовано механізми попередньої обробки даних, сегментації, класифікації та формування результатів детекції, що дозволило забезпечити стабільну роботу системи в умовах змін освітлення, шумів та різної якості відеопотоку.

У роботі розроблено інформаційне забезпечення системи, яке включає концептуальну, логічну та фізичну моделі бази даних. Запропонована структура забезпечує ефективне збереження:

– параметрів об’єктів;

– характеристик техніки;

– результатів класифікації;

– метаданих зображень;

– параметрів моделей машинного навчання.

У процесі дослідження сформовано навчальний датасет та реалізовано процедури підготовки даних, включаючи аугментацію, нормалізацію та формування тренувальних, валідаційних і тестових вибірок. Для автоматизованого підбору конфігурації нейронної мережі використано методи GridSearch та Keras Tuner, що дозволило визначити оптимальні значення гіперпараметрів моделі.

У результаті навчання нейронної мережі було отримано високі показники якості класифікації:

– точність валідації – 0.987;

– середня точність при п’ятикратній перехресній перевірці – 0.986.

Додатково проведено оцінювання моделі із використанням метрик MCC, F2-score, AUC та PRC, результати яких підтвердили стабільність роботи системи та її здатність до узагальнення на нових даних. Аналіз графіків навчання та матриці помилок показав відсутність критичних ознак перенавчання та ефективність використаних методів регуляризації.

У межах практичної частини реалізовано користувацький інтерфейс програмного комплексу, який забезпечує:

– завантаження зображень;

– запуск процесу розпізнавання;

– відображення результатів класифікації;

– адміністрування класів об’єктів;

– роботу з базою даних.

Розроблене програмне забезпечення інтегровано з базою даних SQLite, що забезпечує накопичення результатів аналізу та підтримку роботи системи у локальному режимі.

Наукова новизна отриманих результатів полягає в:

– удосконаленні підходу до інтеграції модулів комп’ютерного зору у структуру кіберфізичної системи управління БПЛА;

– адаптації архітектури нейронної мережі до задач розпізнавання об’єктів аерозйомки в умовах обмежених обчислювальних ресурсів;

– удосконаленні процесу підготовки та аналізу даних для задач автоматизованої класифікації об’єктів у відеопотоці.

Практичне значення роботи полягає у можливості використання розробленої системи для:

– автоматизованого моніторингу територій;

– задач аеророзвідки;

– підтримки прийняття рішень;

– аналізу відеоданих із БПЛА;

– побудови інтелектуальних систем автономного керування квадрокоптерами.

Отримані результати можуть бути використані у подальших дослідженнях, пов’язаних із:

– автономною навігацією БПЛА;

– системами колективного управління дронами;

– багатокласовою детекцією об’єктів;

– оптимізацією моделей глибокого навчання для вбудованих систем;

– інтеграцією алгоритмів штучного інтелекту в кіберфізичні системи реального часу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Guilherme V. R., Manuel G. O., Francisco R. R. Robust Nonlinear Control for Path Tracking of a Quad-Rotor Helicopter. Asian Journal of Control. 2015. Vol. 17. pp. 142-156. DOI: 10.1002/asjc.823

2. Lee E. A., Seshia S. A. Introduction to Embedded Systems: A Cyber-Physical Systems Approach. Second Edition. MIT Press, 2017. 306 p. ISBN 978-0-262-53381-2.

3. Redmon J., Farhadi A. YOLOv3: An Incremental Improvement. *Computer Vision and Pattern Recognition*. 2018. Vol. 1. P. 1–6. DOI: 10.48550/arXiv.1804.02767.

4. Bochkovskiy A., Wang C.Y., Liao H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *Computer Vision and Pattern Recognition*. 2020. Vol. 1. P. 1–17. DOI: 10.48550/arXiv.2004.10934.

5. Jocher G., Chaurasia A., Qiu J. ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation. *Zenodo*. 2022. DOI: 10.5281/zenodo.3908559.

6. He K., Gkioxari G., Dollár P., Girshick R. Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2020. Vol. 42(2). P. 386–397. DOI: 10.1109/TPAMI.2018.2844175.

7. Ren S., He K., Girshick R., Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2017. Vol. 39(6). P. 1137–1149. DOI: 10.1109/TPAMI.2016.2577031.

8. Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.Y., Berg A.C. SSD: Single Shot MultiBox Detector. *Lecture Notes in Computer Science*. 2016. Vol. 9905. P. 21–37. DOI: 10.1007/978-3-319-46448-0_2.

9. Tan M., Le Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *Proceedings of the 36th International Conference on Machine Learning*. 2019. Vol. 97. P. 6105–6114. DOI: 10.48550/arXiv.1905.11946.

10. Howard A.G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H. MobileNets: Efficient Convolutional Neural Networks for

Mobile Vision Applications. *Computer Vision and Pattern Recognition*. 2017. Vol. 1. P. 1–9. DOI: 10.48550/arXiv.1704.04861.

11. Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *IEEE Conference on Computer Vision and Pattern Recognition*. 2018. P. 4510–4520. DOI: 10.1109/CVPR.2018.00474.

12. Lin T.Y., Dollár P., Girshick R., He K., Hariharan B., Belongie S. Feature Pyramid Networks for Object Detection. *IEEE Conference on Computer Vision and Pattern Recognition*. 2017. P. 936–944. DOI: 10.1109/CVPR.2017.106.

13. Kingma D.P., Ba J. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*. 2015. Vol. 1. P. 1–15. DOI: 10.48550/arXiv.1412.6980.

14. Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A. Going Deeper with Convolutions. *IEEE Conference on Computer Vision and Pattern Recognition*. 2015. P. 1–9. DOI: 10.1109/CVPR.2015.7298594.

15. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations*. 2015. Vol. 1. P. 1–14. DOI: 10.48550/arXiv.1409.1556.

16. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition*. 2016. P. 770–778. DOI: 10.1109/CVPR.2016.90.

17. Dosovitskiy A., Beyer L., Kolesnikov A., Weissenborn D., Zhai X., Unterthiner T. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *International Conference on Learning Representations*. 2021. Vol. 1. P. 1–22. DOI: 10.48550/arXiv.2010.11929.

18. Wu X., Li W., Hong D., Tao R., Du Q. Deep Learning for UAV-based Object Detection and Tracking: A Survey. *IEEE Geoscience and Remote Sensing Magazine*. 2022. Vol. 10(3). P. 91–124. DOI: 10.48550/arXiv.2110.12638.

19. Tang G., Zhang Z., Wang P., Liu H. A Survey of Object Detection for UAVs Based on Deep Learning. *Remote Sensing*. 2023. Vol. 16(1). P. 149. DOI: 10.3390/rs16010149.

20. Du D., Qi Y., Yu H., Yang Y., Duan K., Li G., Zhang W., Huang Q., Tian Q. The Unmanned Aerial Vehicle Benchmark: Object Detection and Tracking. *European Conference on Computer Vision*. 2018. Vol. 11206. P. 375–391. DOI: 10.48550/arXiv.1804.00518.

21. Fan Q., Yang S., Li H., Chen X. LUD-YOLO: A Novel Lightweight Object Detection Network for UAV Vision Systems. *Information Sciences*. 2025. Vol. 690. P. 121755. DOI: 10.1016/j.ins.2024.121366.

22. Wu Y., Zhang T., Li H. Research Advances on Deep Learning-Based Small Object Detection in UAV Aerial Images. *Acta Aeronautica et Astronautica Sinica*. 2025. Vol. 46(3). P. 1–22. DOI: 10.7527/S1000-6893.2024.30848.

23. Alsheikhy A.A., Barr M., Boubaker S., Said Y. DV-YOLO: A Deep Learning Framework for Small-Object Detection in UAV-Based Remote Sensing Imagery. *AIMS Mathematics*. 2026. Vol. 11(4). P. 12043–12063. DOI: 10.3934/math.2026494.

24. Zeng W., Mao G., Li M., Yin S. Deep Learning-Based Object Detection: A Comprehensive Review of YOLO, RCNN, and SSD Series. *Electronic Research Archive*. 2026. Vol. 34(4). P. 2674–2731. DOI: 10.3934/era.2026124.

25. Zhao C., Shi Q., Liu Y. Deep Learning-Based Object Detection in Maritime UAV Scenarios: A Review. *Journal of Marine Science and Engineering*. 2023. Vol. 11(11). P. 1–29. DOI: 10.48550/arXiv.2311.07955.

26. Shin J., Kim J., Park H. UAV-Assisted and Deep Learning-Driven Object Detection for Autonomous Systems. *Proceedings of the ACM Symposium on Applied Computing*. 2022. P. 1147–1154. DOI: 10.1145/3555661.3560856.

27. Zhang K., Liu P., Wang H. Navigation Mark Detection Based on Deep Learning Models in UAV Images. *Discover Artificial Intelligence*. 2025. Vol. 5(1). P. 1–15. DOI: 10.1007/s43762-025-00229-2.

28. Ciccone F., Mancini S., Bianchi L. Real-Time Search and Rescue with Drones: A Deep Learning Approach. *Drones*. 2025. Vol. 9(8). P. 514. DOI: 10.3390/drones9080514.

29. Goodfellow I., Bengio Y., Courville A. Deep Learning. *MIT Press*. 2016. Vol. 1. P. 1–775. DOI: 10.1007/s10710-017-9314-z.

30. LeCun Y., Bengio Y., Hinton G. Deep Learning. *Nature*. 2015. Vol. 521. P. 436–444. DOI: 10.1038/nature14539.

31. Shorten C., Khoshgoftaar T.M. A Survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*. 2019. Vol. 6(60). P. 1–48. DOI: 10.1186/s40537-019-0197-0.

32. Chollet F. Xception: Deep Learning with Depthwise Separable Convolutions. *IEEE Conference on Computer Vision and Pattern Recognition*. 2017. P. 1800–1807. DOI: 10.1109/CVPR.2017.195.

33. Howard A., Sandler M., Chu G., Chen L.C., Chen B., Tan M., Wang W., Zhu Y., Pang R., Vasudevan V., Le Q. Searching for MobileNetV3. *IEEE International Conference on Computer Vision*. 2019. P. 1314–1324. DOI: 10.1109/ICCV.2019.00140.

34. Lin T.Y., Maire M., Belongie S., Hays J., Perona P., Ramanan D., Dollár P., Zitnick C.L. Microsoft COCO: Common Objects in Context. *European Conference on Computer Vision*. 2014. Vol. 8693. P. 740–755. DOI: 10.1007/978-3-319-10602-1_48.

35. Everingham M., Van Gool L., Williams C.K.I., Winn J., Zisserman A. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*. 2010. Vol. 88(2). P. 303–338. DOI: 10.1007/s11263-009-0275-4.

36. Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. *IEEE Conference on Computer Vision and Pattern Recognition*. 2016. P. 779–788. DOI: 10.1109/CVPR.2016.91.

37. Girshick R. Fast R-CNN. *IEEE International Conference on Computer Vision*. 2015. P. 1440–1448. DOI: 10.1109/ICCV.2015.169.

38. He K., Zhang X., Ren S., Sun J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *IEEE International Conference on Computer Vision*. 2015. P. 1026–1034. DOI: 10.1109/ICCV.2015.123.

39. Ioffe S., Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *International Conference on Machine Learning*. 2015. Vol. 37. P. 448–456. DOI: 10.48550/arXiv.1502.03167.

40. Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*. 2014. Vol. 15. P. 1929–1958. DOI: https://dl.acm.org/doi/10.5555/2627435.2670313.

41. Krizhevsky A., Sutskever I., Hinton G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*. 2017. Vol. 60(6). P. 84–90. DOI: 10.1145/3065386.

42. Hinton G., Vinyals O., Dean J. Distilling the Knowledge in a Neural Network. *Computer Science*. 2015. Vol. 1. P. 1–9. DOI: 10.48550/arXiv.1503.02531.

43. Han S., Mao H., Dally W.J. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *International Conference on Learning Representations*. 2016. Vol. 1. P. 1–14. DOI: 10.48550/arXiv.1510.00149.

44. Jacob B., Kligys S., Chen B., Zhu M., Tang M., Howard A., Adam H., Kalenichenko D. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. *IEEE Conference on Computer Vision and Pattern Recognition*. 2018. P. 2704–2713. DOI: 10.1109/CVPR.2018.00286.

45. Zoph B., Vasudevan V., Shlens J., Le Q.V. Learning Transferable Architectures for Scalable Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition*. 2018. P. 8697–8710. DOI: 10.1109/CVPR.2018.00907.

46. Huang G., Liu Z., Van Der Maaten L., Weinberger K.Q. Densely Connected Convolutional Networks. *IEEE Conference on Computer Vision and Pattern Recognition*. 2017. P. 4700–4708. DOI: 10.1109/CVPR.2017.243.

47. Ronneberger O., Fischer P., Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Lecture Notes in Computer Science*. 2015. Vol. 9351. P. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

48. Long J., Shelhamer E., Darrell T. Fully Convolutional Networks for Semantic Segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*. 2015. P. 3431–3440. DOI: 10.1109/CVPR.2015.7298965.

49. Chen L.C., Papandreou G., Kokkinos I., Murphy K., Yuille A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2018. Vol. 40(4). P. 834–848. DOI: 10.1109/TPAMI.2017.2699184.

50. Carion N., Massa F., Synnaeve G., Usunier N., Kirillov A., Zagoruyko S. End-to-End Object Detection with Transformers. *European Conference on Computer Vision*. 2020. Vol. 12346. P. 213–229. DOI: 10.1007/978-3-030-58452-8_13.

51. Zhang Z., Sabuncu M. Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. *Advances in Neural Information Processing Systems*. 2018. Vol. 31. P. 8778–8788. DOI: 10.48550/arXiv.1805.07836.

52. Müller R., Kornblith S., Hinton G. When Does Label Smoothing Help? *Advances in Neural Information Processing Systems*. 2019. Vol. 32. P. 4694–4703. DOI: 10.48550/arXiv.1906.02629.

53. Zhang H., Cisse M., Dauphin Y.N., Lopez-Paz D. mixup: Beyond Empirical Risk Minimization. *International Conference on Learning Representations*. 2018. Vol. 1. P. 1–13. DOI: 10.48550/arXiv.1710.09412.

54. Yun S., Han D., Oh S.J., Chun S., Choe J., Yoo Y. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. *IEEE International Conference on Computer Vision*. 2019. P. 6023–6032. DOI: 10.1109/ICCV.2019.00612.

55. Touvron H., Cord M., Douze M., Massa F., Sablayrolles A., Jégou H. Training Data-Efficient Image Transformers & Distillation Through Attention. *International Conference on Machine Learning*. 2021. Vol. 139. P. 10347–10357. DOI: 10.48550/arXiv.2012.12877.

56. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser Ł., Polosukhin I. Attention Is All You Need. *Advances in Neural Information Processing Systems*. 2017. Vol. 30. P. 5998–6008. DOI: 10.48550/arXiv.1706.03762.

57. Brown T., Mann B., Ryder N., Subbiah M., Kaplan J., Dhariwal P., Neelakantan A., Shyam P., Sastry G., Askell A. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*. 2020. Vol. 33. P. 1877–1901. DOI: 10.48550/arXiv.2005.14165.

58. Devlin J., Chang M.W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of NAACL-HLT*. 2019. Vol. 1. P. 4171–4186. DOI: 10.48550/arXiv.1810.04805.

59. Radford A., Metz L., Chintala S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *International Conference on Learning Representations*. 2016. Vol. 1. P. 1–16. DOI: 10.48550/arXiv.1511.06434.

60. Isola P., Zhu J.Y., Zhou T., Efros A.A. Image-to-Image Translation with Conditional Adversarial Networks. *IEEE Conference on Computer Vision and Pattern Recognition*. 2017. P. 1125–1134. DOI: 10.1109/CVPR.2017.632.

61. Zhu J.Y., Park T., Isola P., Efros A.A. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. *IEEE International Conference on Computer Vision*. 2017. P. 2242–2251. DOI: 10.1109/ICCV.2017.244.

62. Goodfellow I.J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y. Generative Adversarial Nets. *Advances in Neural Information Processing Systems*. 2014. Vol. 27. P. 2672–2680. DOI: 10.48550/arXiv.1406.2661.

63. Ultralytics YOLO Docs: Real-time Object Detection and Image Segmentation. Режим доступу: https://docs.ultralytics.com/.

64. Redmon J., Farhadi A. YOLOv3: An Incremental Improvement. *arXiv*. 1804.02767. 2018. DOI: 10.48550/arXiv.1804.02767.

65. PX4 Autopilot User Guide: Open Source Autopilot for Drones. Режим доступу: https://docs.px4.io/main/en/.

66. YOLO26: Next-generation real-time object detection with NMS-free inference. Ultralytics Documentation. Режим доступу: https://docs.ultralytics.com/models/yolo26/.

67. Offboard Mode (Generic/All) – PX4 Guide. Режим доступу: https://docs.px4.io/main/en/flight_modes/offboard.html.

68. uXRCE-DDS (PX4-ROS 2/DDS Bridge) – PX4 Middleware. Режим доступу: https://docs.px4.io/main/en/middleware/uxrce_dds.html.

69. YOLO11: Real-time object detection, segmentation and classification. Ultralytics Documentation. Режим доступу: https://docs.ultralytics.com/models/yolo11/.

70. Companion Computers: Pixhawk + Companion Setup. PX4 User Guide. Режим доступу: https://docs.px4.io/main/en/companion_computer/pixhawk_companion.html.

71. MAVLink Messaging Protocol for Drones and Autopilots. Режим доступу: https://mavlink.io/en/.

72. Ultralytics Licensing: AGPL-3.0 and Enterprise Licenses. Режим доступу: https://ultralytics.com/license.

73. Kucherov D. P., Kozub A. N., Kostyna O. N. Group Behavior of UAVs in Obstacles Presence. *Proc. of IEEE 4th Int. Conf. "Methods and Systems of Navigation and Motion Control (MSNMC)*. Kyiv, Ukraine, October 18-20 2016. pp. 51-54. DOI: 10.1109/MSNMC.2016.7783104

74. Коваленко Н. П., Лисенко Ю. І. Застосування нейронних мереж у задачі автоматичної класифікації зображень. *Матеріали II Міжнародної науково-технічної конференції "Проблеми та перспективи розвитку сучасної техніки і технологій"*. Київ, 2017. С. 111-114.

75. Родіонов О. С., Черкаський В. Ю. Методи обробки та аналізу знімків аерозвідки для задач військового застосування. *Військова техніка*. 2019. № 1 (21). С. 43-49.

76. Березін, А. В., Гудков, О. М., Кіпріянов, І. В. Автоматизована система розпізнавання військової техніки на зображеннях. *Збірник наукових праць Харківського національного університету імені В. Н. Каразіна. Серія: Радіофізика та електроніка*. ХНУ. 2018. Випуск 23. С. 122-131.

77. Абрамов С.В., Лупаленко О.В., Манжай О.В. Аналіз автоматизованих систем виявлення та розпізнавання об’єктів збройних сил Російської Федерації.

Scientific Collection «InterConf» 95. Scientific goals and purposes in XXI century. 2022. С. 893–905. DOI: 10.51582/interconf.19-20.01.2022.098

78. Mustafin R., Isakov I., Kussainov T. Recognition of military equipment on aerial photographs using convolutional neural networks. *International Journal of Engineering & Technology*. 2018. 7(4.6). P. 29-33. DOI: 10.23939/csn2025.01.047

79. Tomè D., Monti F., Baroffio L., Bondi L., Tagliasacchi M., Tubaro S. Deep Convolutional Neural Networks for pedestrian detection. Signal Processing: Image Communication. Vol. 47. 2016. pp. 482-489. DOI: 10.1016/j.image.2016.05.007

80. Chen L.C., Papandreou G., Kokkinos I., Murphy K., Yuille, A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2018. 40(4). P. 834-848. DOI: 0.1109/TPAMI.2017.2699184

81. DeepLearning 0.1. LISA Lab. Посилання: https://github.com/lisa-lab/DeepLearningTutorials

82. Habibi A.H., Elnaz J.H. Guide to convolutional neural networks: a practical application to traffic-sign detection and classification. *Cham: Springer International Publishing*. 2017. 348 P.

83. Nguyen H. Fast object detection framework based on mobilenetv2 architecture and enhanced feature pyramid. *Journal of Theoretical and Applied Information Technology*. 2020. 98 (05). P. 145-156. Постійний доступ: https://www.semanticscholar.org/paper/FAST-OBJECT-DETECTION-FRAMEWORK-BASED-ON-AND-Nguyen/20b93ed7b79c5f4afd1f52d2f555ecc5599ba2ee

84. Matusugu M., Katsuhiko M., Yusuke M., Yuji K. Designing Convolutional Neural Network Architecture Using Genetic Algorithms. *International Journal of Advanced Network, Monitoring and Controls*. 2021. Volume 6. 555–559. DOI: 10.21307/ijanmc-2021-024

85. Гріцай І.С. Кіберфізична система управління квадрокоптерами для розпізнавання об’єктів за допомогою машинного навчання. 17-а Міжнародна студентська науково-технічна конференція «Перспективні мережеві та комп’ютерні технології» (ПЕРСИК 2026). Харків, ХАІ. 2026. с. 51.

ДОДАТОК А

(обов’язковий)

Код модуля навчання нейронної мережі

```
import os
import numpy as np
import tensorflow as tf
import keras_tuner as kt
from tensorflow import keras
from tensorflow.keras import layers
from sklearn.model_selection import KFold
from sklearn.metrics import accuracy_score, matthews_corrcoef, f1_score,
classification_report

DATASET_DIR = "dataset"
IMG_SIZE = (128, 128)
BATCH_SIZE = 32
EPOCHS = 25
NUM_CLASSES = 8
SEED = 42

def load_dataset(dataset_dir):
    dataset = keras.utils.image_dataset_from_directory(
        dataset_dir,
        labels="inferred",
        label_mode="categorical",
        image_size=IMG_SIZE,
        batch_size=None,
        shuffle=True,
        seed=SEED
    )
    images = []
```

```
    labels = []
    for image, label in dataset:
        images.append(image.numpy())
        labels.append(label.numpy())
    images = np.array(images, dtype="float32")
    labels = np.array(labels, dtype="float32")
    return images, labels

def build_model(hp):
    model = keras.Sequential()
    model.add(layers.InputLayer(input_shape=(IMG_SIZE[0], IMG_SIZE[1], 3)))
    model.add(layers.Rescaling(1.0 / 255))
    conv_blocks = hp.Int("conv_blocks", min_value=2, max_value=4, step=1)
    for i in range(conv_blocks):
        filters = hp.Choice(f"filters_{i}", values=[16, 32, 64])
        model.add(layers.Conv2D(filters, kernel_size=(3, 3), activation="relu",
padding="same"))
        model.add(layers.MaxPooling2D(pool_size=(2, 2)))
    model.add(layers.Flatten())
    dense_units = hp.Choice("dense_units", values=[32, 64, 128])
    model.add(layers.Dense(dense_units, activation="relu"))
    dropout_rate = hp.Choice("dropout_rate", values=[0.2, 0.3, 0.4])
    model.add(layers.Dropout(dropout_rate))
    model.add(layers.Dense(NUM_CLASSES, activation="softmax"))
    learning_rate = hp.Float(
        "learning_rate",
        min_value=1e-4,
        max_value=1e-2,
        sampling="log"
    )
    model.compile(
```

```
        optimizer=keras.optimizers.Adam(learning_rate=learning_rate),
        loss="categorical_crossentropy",
        metrics=["categorical_accuracy", keras.metrics.AUC(name="auc")]
    )
    return model

def tune_hyperparameters(x_train, y_train, x_val, y_val):
    tuner = kt.GridSearch(
        build_model,
        objective="val_categorical_accuracy",
        max_trials=20,
        directory="tuner_results",
        project_name="uav_object_recognition"
    )
    tuner.search(
        x_train,
        y_train,
        validation_data=(x_val, y_val),
        epochs=10,
        batch_size=BATCH_SIZE
    )
    best_hp = tuner.get_best_hyperparameters(num_trials=1)[0]
    return best_hp

def train_final_model(x_train, y_train, x_val, y_val, best_hp):
    model = build_model(best_hp)
    callbacks = [
        keras.callbacks.EarlyStopping(
            monitor="val_loss",
            patience=5,
            restore_best_weights=True
```

```
        ),
        keras.callbacks.ModelCheckpoint(
            "trained_model.keras",
            monitor="val_categorical_accuracy",
            save_best_only=True
        )
    ]
    history = model.fit(
        x_train,
        y_train,
        validation_data=(x_val, y_val),
        epochs=EPOCHS,
        batch_size=BATCH_SIZE,
        callbacks=callbacks
    )
    return model, history

def cross_validate_model(images, labels, best_hp):
    kfold = KFold(n_splits=5, shuffle=True, random_state=SEED)
    scores = []
    for fold, (train_idx, val_idx) in enumerate(kfold.split(images), start=1):
        print(f"Fold {fold}")
        x_train, x_val = images[train_idx], images[val_idx]
        y_train, y_val = labels[train_idx], labels[val_idx]
        model = build_model(best_hp)
        model.fit(
            x_train,
            y_train,
            validation_data=(x_val, y_val),
            epochs=EPOCHS,
            batch_size=BATCH_SIZE,
```

```
            verbose=1
        )
        predictions = model.predict(x_val)
        y_pred = np.argmax(predictions, axis=1)
        y_true = np.argmax(y_val, axis=1)
        accuracy = accuracy_score(y_true, y_pred)
        scores.append(accuracy)
        print(f"Validation accuracy: {accuracy:.4f}")
    print(f"Mean validation accuracy: {np.mean(scores):.4f}")
    return scores

def evaluate_model(model, x_test, y_test):
    predictions = model.predict(x_test)
    y_pred = np.argmax(predictions, axis=1)
    y_true = np.argmax(y_test, axis=1)
    accuracy = accuracy_score(y_true, y_pred)
    mcc = matthews_corrcoef(y_true, y_pred)
    f1 = f1_score(y_true, y_pred, average="weighted")
    print("Test accuracy:", accuracy)
    print("MCC:", mcc)
    print("F1-score:", f1)
    print(classification_report(y_true, y_pred))

def main():
    images, labels = load_dataset(DATASET_DIR)
    total_count = len(images)
    test_count = int(total_count * 0.10)
    x_test = images[:test_count]
    y_test = labels[:test_count]
    x_data = images[test_count:]
    y_data = labels[test_count:]
```

```
    val_count = int(len(x_data) * 0.20)
    x_val = x_data[:val_count]
    y_val = y_data[:val_count]
    x_train = x_data[val_count:]
    y_train = y_data[val_count:]
    print("Total images:", total_count)
    print("Train images:", len(x_train))
    print("Validation images:", len(x_val))
    print("Test images:", len(x_test))
    best_hp = tune_hyperparameters(x_train, y_train, x_val, y_val)
    print("Best hyperparameters:")
    print("conv_blocks:", best_hp.get("conv_blocks"))
    print("dense_units:", best_hp.get("dense_units"))
    print("learning_rate:", best_hp.get("learning_rate"))
    model, history = train_final_model(x_train, y_train, x_val, y_val, best_hp)
    cross_validate_model(x_data, y_data, best_hp)
    evaluate_model(model, x_test, y_test)
    model.save("uav_object_classification_model.keras")

if __name__ == "__main__":
    main()
```

ДОДАТОК Б

(обов’язковий)

Презентація

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп’ютерної інженерії та інформаційних систем

Кіберфізична система управління квадрокоптерами для розпізнавання об'єктів за допомогою машинного навчання

Здобувач: Гріцай Ілля Сергійович

Керівник: д.т.н., професор Федоров Євген Євгенович

1

АКТУАЛЬНІСТЬ ДОСЛІДЖЕННЯ

Актуальність обумовлена необхідністю створення інтелектуальних кіберфізичних систем, здатних забезпечувати автоматизований аналіз зображень та відеоданих у режимі реального часу. Існуючі системи розпізнавання об’єктів часто потребують значних обчислювальних ресурсів, мають складність інтеграції з автономними системами керування або демонструють недостатню ефективність при роботі зі зображеннями аерозйомки, що містять шуми, спотворення та складні умови освітлення.

2

ОБ’ЄКТ, ПРЕДМЕТ ТА МЕТА ДОСЛІДЖЕННЯ

Об’єктом дослідження є процес автоматизованого управління квадрокоптерами на основі аналізу відеоданих та методів машинного навчання.

Предметом дослідження є методи та програмні засоби розпізнавання об’єктів у кіберфізичних системах управління безпілотними літальними апаратами.

Метою роботи є розробка кіберфізичної системи управління квадрокоптерами для розпізнавання об’єктів за допомогою технологій машинного навчання та комп’ютерного зору.

Завдання, які потрібно вирішити в межах роботи:

– провести аналіз сучасних методів та систем розпізнавання об’єктів у БПЛА;

– дослідити підходи до побудови кіберфізичних систем управління квадрокоптерами;

– виконати аналіз моделей машинного навчання та алгоритмів комп’ютерного зору;

– розробити функціональну та інформаційну архітектуру системи;

– реалізувати програмний модуль розпізнавання об’єктів;

– сформувати та підготувати навчальний датасет;

– виконати навчання та налаштування нейронної мережі;

– провести тестування та оцінювання ефективності роботи системи.

Завдання, які потрібно вирішити в межах роботи:

– провести аналіз сучасних методів та систем розпізнавання об’єктів у БПЛА;

– дослідити підходи до побудови кіберфізичних систем управління квадрокоптерами;

– виконати аналіз моделей машинного навчання та алгоритмів комп’ютерного зору;

– розробити функціональну та інформаційну архітектуру системи;

– реалізувати програмний модуль розпізнавання об’єктів;

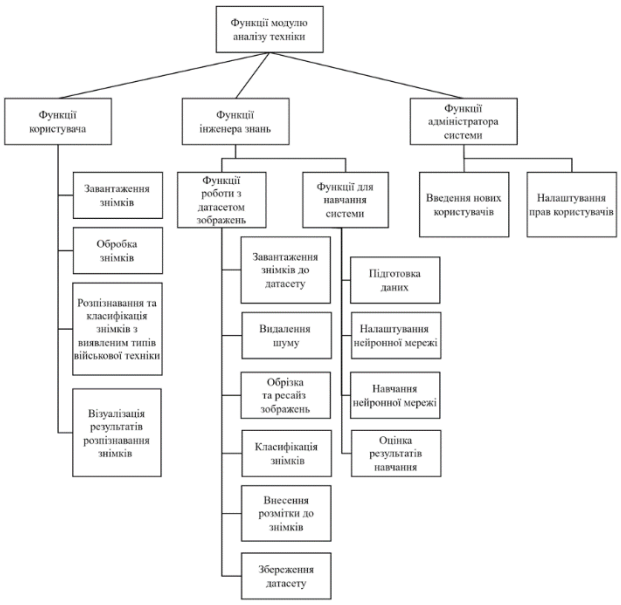
– сформувати та підготувати навчальний датасет;

– виконати навчання та налаштування нейронної мережі;

– провести тестування та оцінювання ефективності роботи системи.

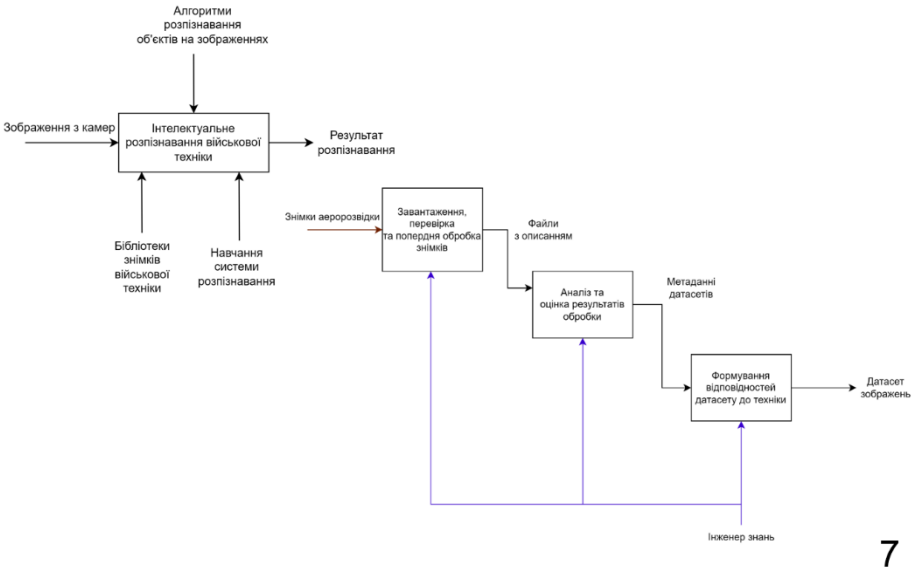
5

Дерево функцій кіберфізичної системи управління БПЛА з модулем комп’ютерного зору



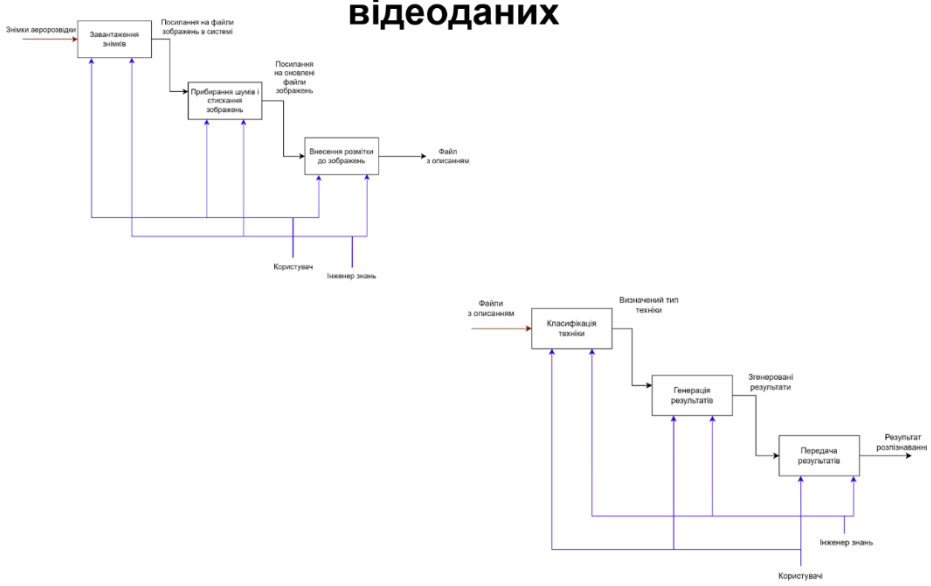
6

Контекстні діаграми програмного модуля



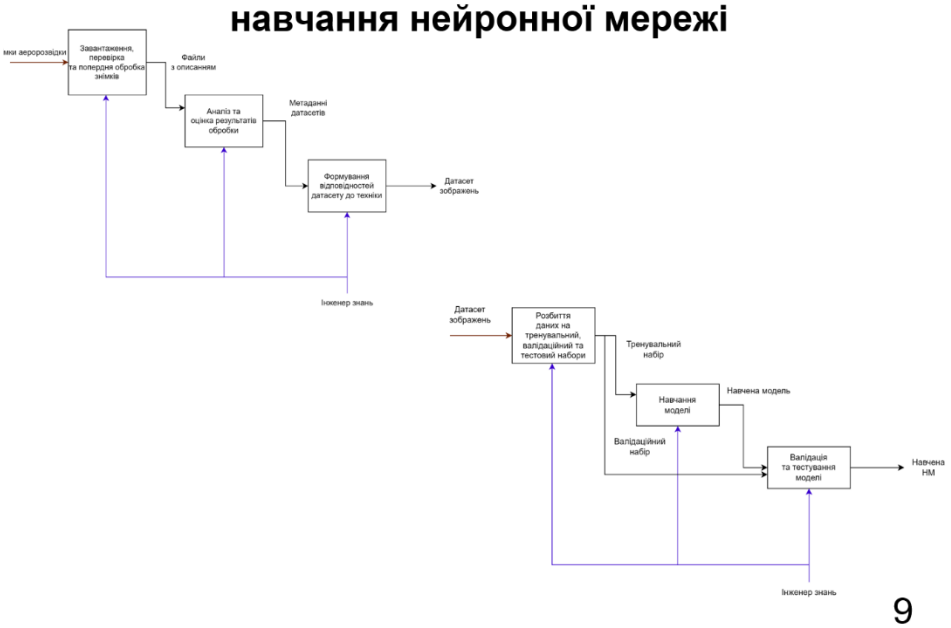
7

Діаграми IDEF0 рівень 1 процесів обробки відеоданих



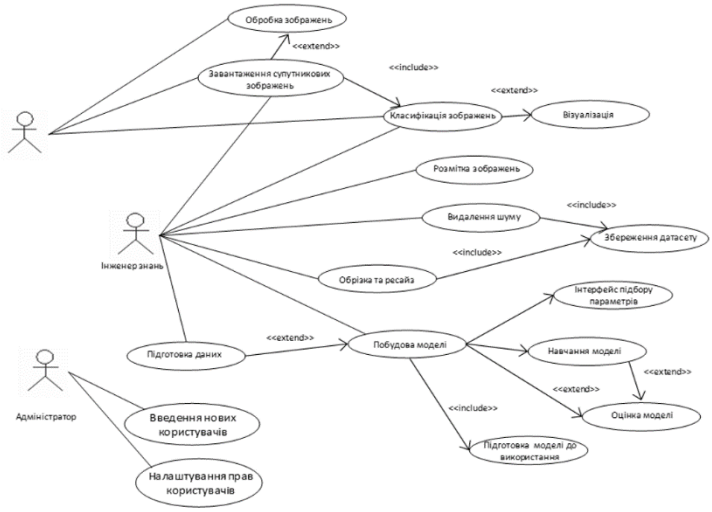
8

Діаграми IDEF0 рівень 1 процесів для навчання нейронної мережі



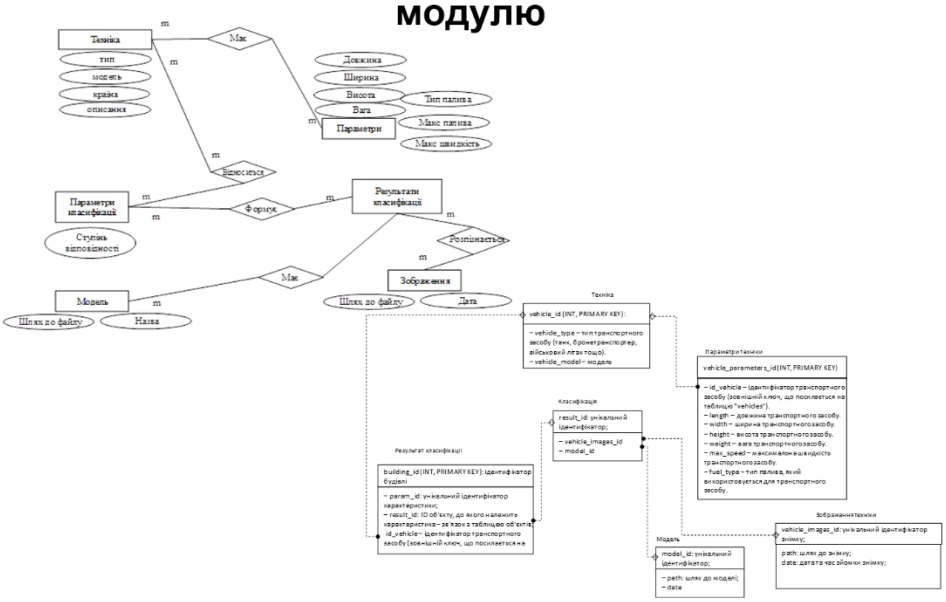
9

UML-діаграма варіантів використання модуля комп’ютерного зору



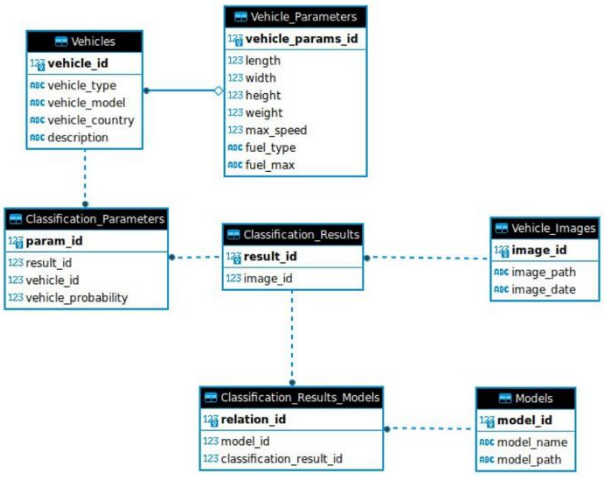
10

Проєктування бази данних програмного модулю



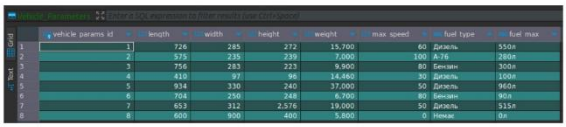
11

Проєктування бази данних програмного модулю

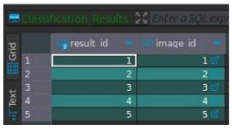


12

Наповнення таблиць бази даних у системі розпізнавання

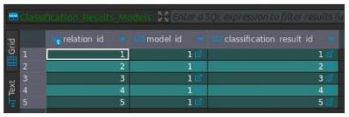


vehicle_params_id	length	width	height	weight	max_speed	fuel_type	fuel_max
1	726	285	272	15,700	60	Дизель	550л
2	575	235	239	7,000	100	А-76	280л
3	756	283	223	9,900	80	Бензин	300л
4	410	97	96	14,460	30	Дизель	100л
5	934	330	240	37,000	50	Дизель	960л
6	704	250	248	6,700	80	Бензин	90л
7	653	312	2,576	19,000	50	Дизель	515л
8	600	900	400	5,800	0	Немає	0л

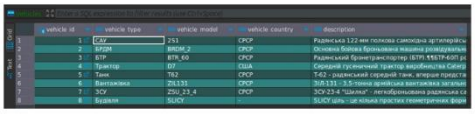


result_id	image_id
1	1
2	2
3	3
4	4
5	5

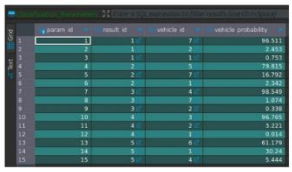
model_id	model_name	model_path
1	wartech_classifier	models/model1.h5
2	wartech_classifier	models/model2.h5



relation_id	model_id	classification_result_id
1	1	1
2	1	2
3	1	3
4	1	4
5	1	5



vehicle_id	vehicle_type	vehicle_model	vehicle_country	description
1	САУ	2S1	СРСР	Радянська 122-мм гаубична самохідна артилерійська
2	БРДМ	BRDM_2	СРСР	Основна бойова броньована машина розвідувальна
3	БТР	BTR_60	СРСР	Радянський бронетранспортер (БТР) 'БТР-60П' ра
4	Трактор	DT	США	Середній гусеничний трактор виробництва Caterp
5	Танк	T62	СРСР	Т-62 - радянський середній танк, вперше предста
6	Вантажівка	ZIL135	СРСР	ЗИЛ-135 - 3,5-тонна армійська вантажівка загальн
7	ЗСУ	ZSU_23_4	СРСР	ЗСУ-23-4 "Шилка" - легкоброньована радянська са
8	Будівля	SLICY	-	SLICY ціль - це кілька простих геометричних форм



param_id	result_id	vehicle_id	vehicle_probability
1	1	7	99.511
2	1	2	2.455
3	1	1	0.753
4	2	5	79.815
5	2	7	16.792
6	2	1	2.342
7	3	4	99.549
8	3	7	1.976
9	3	2	0.338
10	4	3	96.766
11	4	2	3.221
12	4	1	0.314
13	5	6	61.179
14	5	1	30.24
15	5	4	5.444

13

Проєктування інтерфейсу програмного модуля

Прототип вікна завантаєення знімків

Прототип вікна результатів розпізнавання знімків

Прототип вікна додавання класів військової техніки

14

Підготовка до навчання моделі

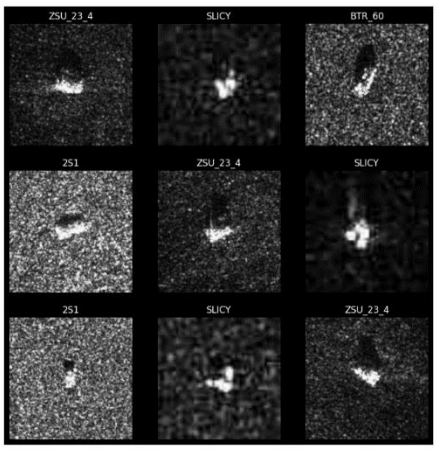
Завантаження дата сету техніки

```
Found 6672 files belonging to 8 classes.
Using 5338 files for training.
Found 6672 files belonging to 8 classes.
Using 1334 files for validation.
Found 747 files belonging to 8 classes.
```

Формування моделі

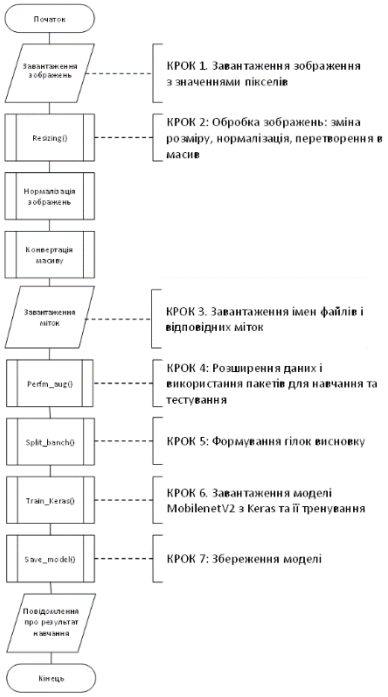
```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 rescaling_2 (Rescaling)     (None, 128, 128, 1)       0

 conv2d_6 (Conv2D)           (None, 126, 126, 32)      320

 max_pooling2d_6 (MaxPooling  (None, 63, 63, 32)       0
 2D)

 conv2d_7 (Conv2D)           (None, 61, 61, 32)        9248

 max_pooling2d_7 (MaxPooling  (None, 30, 30, 32)       0
 2D)

 conv2d_8 (Conv2D)           (None, 28, 28, 32)        9248

 max_pooling2d_8 (MaxPooling  (None, 14, 14, 32)       0
 2D)

 flatten_2 (Flatten)         (None, 6272)              0

 dense_4 (Dense)             (None, 32)                200736

...
Total params: 219,816
Trainable params: 219,816
Non-trainable params: 0
```

Приклади знімків військової техніки



15

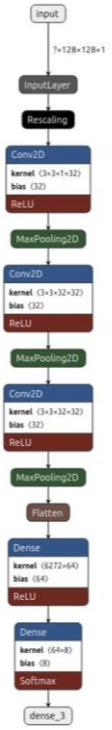
Схема алгоритму попередньої обробки і тренування моделі



16

Архітектура Нейронної Мережі

1. Inputlayer: початковий шар мережі, який визначає формат та розмір вхідних зображень;
2. Rescaling: шар, який виконує масштабування вхідних зображень для приведення їх значень до певного діапазону;
3. Conv2D: шар виконує згортку на вхідних зображеннях з використанням фільтрів (ядер) розміром 3x3 з одним вхідним каналом і 32 вихідними каналами. Кожен фільтр проходить по зображенню і виконує обчислення, що допомагають виявити різні ознаки та шаблони;
4. MaxPooling2D: шар виконує пулінг (зменшення розміру) на виході попереднього згорткового шару. В даній архітектурі використовується максимальне значення з певної області для стиснення інформації та забезпечення інваріантності до малих зміщень та змін масштабу;
5. Conv2D: шар є аналогічним до попереднього згорткового шару, але з використанням 32 вхідних та 32 вихідних каналів. Він допомагає виявити більш складні ознаки та шаблони на зображенні;
6. MaxPooling2D: шар є аналогічним до попереднього пулінгового шару і здійснює подальше зменшення розміру вихідних даних;
7. Conv2D: шар є аналогічним до попередніх згорткових шарів, але знову використовує 32 вхідних та 32 вихідних канали і виконує згортку на виході попередніх шарів, щоб виявити ще більш складні ознаки та шаблони;
8. MaxPooling2D: шар є аналогічним до попереднього пулінгового шару;
9. Flatten: шар перетворює вихідні дані з попереднього шару в одновимірний вектор, який може бути використаний як вхід для повністю зв'язаного шару;
10. Dense: шар є повністю зв'язаним шаром з 64 нейронами. Використовує функцію активації ReLU. Його роль полягає в отриманні важливих ознак з попередніх шарів та виконанні подальшого аналізу;
11. Dense: шар є повністю зв'язаним шаром з 8 нейронами. Використовує функцію активації Softmax. Використовується для класифікації зображень та вироблення остаточних рішень щодо класу, до якого належить вхідне зображення.

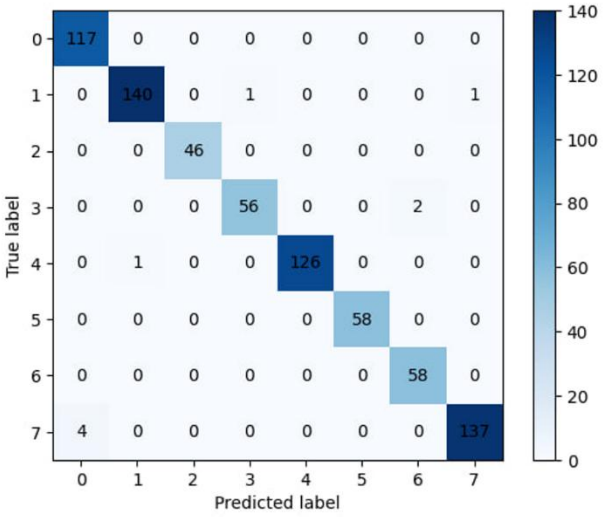


Підбір гіперпараметрів нейронної мережі

- Для того щоб підібрати гіперпарматри було використано функцію GridSearch, класу Tuner, з бібліотеки Keras.
- В результаті було вибрано найкращі парметри:
  - 'conv_blocks': 3,
  - 'filters_0': 32,
  - 'filters_1': 32,
  - 'filters_2': 32,
  - 'dense_units': 32,
  - 'learning_rate': 0.001995262314968881

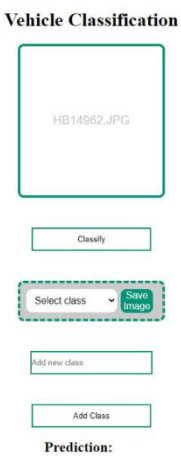
Інформація про вірну класифікацію техніки

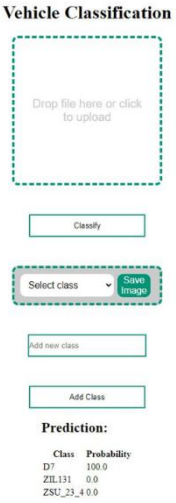
Точність:
98,8%



19

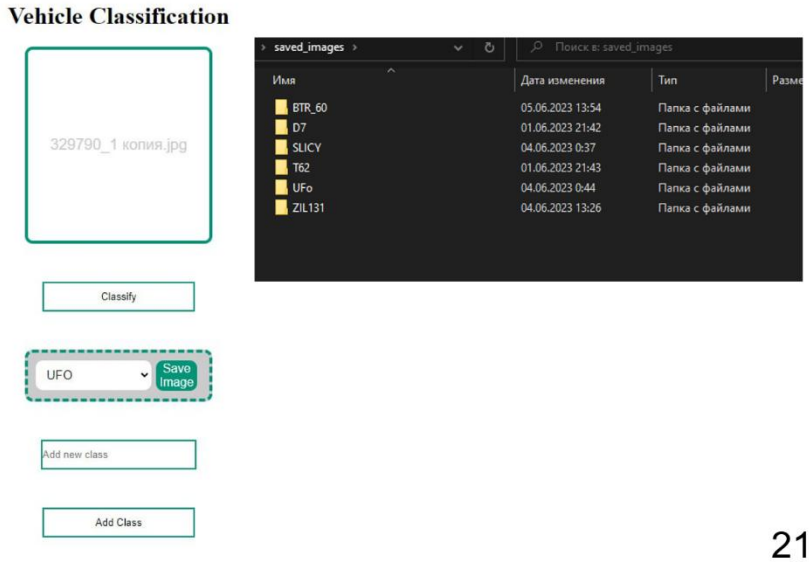
Інтерфейс програмного модуля





20

Додавання нового класу військової техніки



21

Висновки

1) **Проведено аналіз** сучасних методів розпізнавання і класифікації зображень;

2) **Проведено аналіз** методів побудови і навчання штучних нейронних мереж для розпізнавання зображень та розглянуті існуючи програмні системи розпізнавання зображень з використанням нейронних мереж;

3) **Розроблено програмний модуль** визначення військової техніки за знімками аеророзвідки, який забезпечує завантаження, попередня обробку і сегментацію зображень, класифікацію визначених на зображеннях об’єктів та представлення результатів розпізнавання наявних на зображенні типів військової техніки через інтерфейс користувача;

4) **Проведено навчання** нейронної мережі для розпізнавання військової техніки за знімками аеророзвідки та оцінено якість навчання;

5) **Проведено тестування** програмного модуля та надано описання режимів його використання.

22

ДОДАТОК В

(обов’язковий)

Опублікована теза 17-й Міжнародній студентській науково-технічній конференції «Перспективні мережеві та комп’ютерні технології» – ПЕРСИК 2026.

УДК 004.932.72'1

КІБЕРФІЗИЧНА СИСТЕМА УПРАВЛІННЯ КВАДРОКОПТЕРАМИ ДЛЯ РОЗПІЗНАВАННЯ ОБ’ЄКТІВ ЗА ДОПОМОГОЮ МАШИННОГО НАВЧАННЯ

Гріцай І.С., студент гр. КІ2м-24-1

Хмельницький національний університет

Анотація. Сучасний етап розвитку безпілотних літальних апаратів як складних кіберфізичних систем зумовлює потребу в інтеграції інтелектуальних алгоритмів безпосередньо в контур керування. Основним викликом при проектуванні таких систем є обмежена обчислювальна потужність бортового обладнання. У роботі запропоновано вирішення цієї проблеми через впровадження клієнт-серверної архітектури «сенсор—сервер—автопілот», що дозволяє перенести ресурсомістку обробку візуальних даних на зовнішній супровідний комп’ютер або виділений сервер. Це забезпечує підвищення інтелектуальної автономності апарату без критичного збільшення його маси та енергоспоживання.

Метою дослідження є розробка програмно-апаратного комплексу для автономного виявлення об'єктів, здатного функціонувати в режимі реального часу з мінімальною затримкою. Технічну основу запропонованого рішення становить модель YOLO26, яка була обрана завдяки впровадженню технології NMS-free інференсу. Такий підхід дозволяє усунути обчислювальне «вузьке місце» на стадії постобробки, що є критичним для стабільної роботи на периферійних пристроях. Математичний апарат системи базується на використанні бінарної крос-ентропії для класифікації та метрики CIoU для точної регресії обмежувальних рамок.

Програмна реалізація системи базується на середовищі ROS 2, де взаємодія між вузлами здійснюється за стандартом DDS. Для інтеграції високорівневого інтелекту з польотним стеком PX4 використано міст uXRCE-DDS, що забезпечує пряму трансляцію повідомлень uORB. Важливим етапом оптимізації стала квантизація ваг нейромережі до формату INT8 за допомогою бібліотеки TensorRT. Методика базується на лінійному афінному перетворенні та мінімізації розбіжності Кулльбака-Лейблера для збереження точності моделі при значному прискоренні обчислень.

Експериментальні дані, отримані під час тестування на платформі NVIDIA Jetson Nano, підтвердили здатність системи підтримувати частоту обробки кадрів на рівні 15–25 FPS для моделей класу YOLOv8n/11n. При використанні спеціалізованих прискорювачів Google Coral показник продуктивності перевищує 30 FPS. Встановлено, що квантизація INT8 забезпечує чотирикратне скорочення об’єму пам’яті при втраті точності розпізнавання у межах лише 1–2%. Алгоритм візуального сервокерування трансформує екранні координати об'єкта у навігаційні команди TrajectorySetpoint, які передаються автопілоту в режимі Offboard через протокол MAVLink для здійснення автономного маневрування.

Розроблена кіберфізична система демонструє високу ефективність у задачах автономного стеження за динамічними цілями. Поєднання архітектури YOLO26 та оптимізаційних інструментів TensorRT дозволяє досягти необхідного балансу між точністю детекції та швидкістю прийняття рішень у контурі керування БПЛА.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Здобувач: Ілля ГРІЦАЙ

Тема: Кіберфізична система управління квадрокоптерами для розпізнавання об'єктів за допомогою машинного навчання

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи магістра:

Кількість листів креслень – ; кількість сторінок записки 83

1. Короткий зміст роботи та прийнятих рішень У кваліфікаційній роботі розроблено кіберфізичну систему управління квадрокоптерами для розпізнавання об'єктів за допомогою машинного навчання. У роботі запропоновано архітектуру програмно-апаратного комплексу для автоматизованого аналізу зображень та відеопотоку з бортових систем БПЛА, реалізовано модулі комп'ютерного зору та класифікації об'єктів із використанням згорткових нейронних мереж. Також розроблено програмний модуль обробки даних аерозйомки, систему збереження результатів класифікації та механізми навчання й тестування нейронної мережі.

2. Висновок про відповідність роботи дипломному завданню
Кваліфікаційна робота магістра відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено аналіз сучасних кіберфізичних систем управління квадрокоптерами та методів розпізнавання об'єктів за допомогою машинного навчання. Досліджено сучасні моделі комп'ютерного зору, архітектури нейронних мереж та особливості використання БПЛА для задач аеророзвідки й автоматизованого моніторингу територій. У другому розділі розроблено математичні та алгоритмічні моделі функціонування системи управління квадрокоптерами. Сформовано структурно-функціональну архітектуру кіберфізичної системи, досліджено підходи до інтеграції нейронних мереж у структуру автономного керування БПЛА та виконано аналіз методів оптимізації моделей машинного навчання для роботи у режимі реального часу. У третьому розділі розроблено програмне та інформаційне забезпечення системи. Побудовано функціональні моделі у нотації IDEF0, спроєктовано концептуальну, логічну та фізичну моделі бази даних, реалізовано алгоритми

попередньої обробки зображень, формування датасету та навчання нейронної мережі. Також виконано підбір гіперпараметрів моделі та оцінювання ефективності навчання із використанням сучасних методів машинного навчання. У четвертому розділі наведено опис програмної реалізації та результатів тестування розробленої системи. Реалізовано користувацький інтерфейс програмного комплексу, проведено навчання нейронної мережі та оцінювання якості класифікації об'єктів за допомогою метрик точності, матриці помилок та графіків навчання. Досліджено ефективність роботи системи на тестових наборах зображень аерозйомки.

4. Позитивні сторони роботи: Позитивною стороною роботи є комплексний підхід до розробки кіберфізичної системи управління квадрокоптерами, який поєднує методи машинного навчання, комп'ютерного зору та технології обробки відеоданих у режимі реального часу. У роботі реалізовано повний цикл створення системи – від аналізу предметної області та проєктування архітектури до програмної реалізації, навчання нейронної мережі та тестування результатів. Практичну цінність мають реалізовані алгоритми класифікації об'єктів та інтеграція програмного модуля у структуру кіберфізичної системи БПЛА.

5. Негативні сторони роботи: До недоліків роботи можна віднести недостатньо детальне дослідження роботи системи в умовах реального відеопотоку з квадрокоптера, а також обмежений обсяг експериментального аналізу впливу різних погодних умов та рівня шуму на точність класифікації об'єктів.

6. Оцінка графічного оформлення та пояснювальної записки роботи: –

7. Відгук про роботу в цілому: У цілому кваліфікаційна робота виконана на достатньому науково-технічному рівні, має практичне значення та демонструє вміння здобувача самостійно вирішувати задачі проєктування кіберфізичних систем управління квадрокоптерами із використанням сучасних методів машинного навчання та комп'ютерного зору.

8. Інші зауваження: –

9. Оцінка кваліфікаційної роботи магістра:

Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи магістра вважаю, що робота заслуговує оцінки «добре» 75.00 (С)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Едуард Манцис к.т.н., доц. КН

"1" травня 2026р.

Зав. кафедри КІІС
д-р. філософії Ользі ПАВЛОВІЙ

Ілля ГРІЦАЙ

ПІБ здобувача вищої освіти

ФІТ, 2 курсу, групи КІ2м-24-1

ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

1 травня 2026 року

Грицай І

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Кіберфізична система управління квадрокоптерами для розпізнавання об'єктів за допомогою машинного навчання
Автор Ілля ГРІЦАЙ
Освітня програма Комп'ютерна інженерія та програмування
Рівень вищої освіти другий (магістерський)
Спеціальність 123 Комп'ютерна інженерія
Науковий керівник: к.т.н., доцент Євген ФЕДОРОВ

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;

2) окремі виявлені збіги є загальновживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;

3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

4) значна частина знайденого плагіату відноситься до списку використаних джерел

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 11,34%; та системою Anti-Plagiarism складає 4%, що, з урахуванням наведених обгрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

25.04.2026

Завідувач кафедри _____ Підпис — Ольга ПАВЛОВА (Ім'я, ПРІЗВИЩЕ)

Гарант освітньої програми _____ Підпис — Олег САВЕНКО (Ім'я, ПРІЗВИЩЕ)

Керівник кваліфікаційної роботи _____ Підпис — Євген ФЕДОРОВ (Ім'я, ПРІЗВИЩЕ)

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Ілля ГРІЦАЙ

Співавтор:

Назва: Кіберфізична система управління квадрокоптерами для розпізнавання об'єктів за допомогою машинного навчання

Експерт: Євген ФЕДОРОВ

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:11.34%

Коефіцієнт подібності 2:3.72%

Мікропробіли: 439

Заміна букв: 3

Інтервали: 0

Білі знаки: 6

Дата створення звіту: 2026-05-14 23:09:15.0

Після аналізу Звіту подібності констатую наступне:

☒ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

2026-05-15 Доцент Андрій Нічепорук

Дата експерт

Anti-Plagiarism (http://ap.km.ua) v-15.701

Максимальне співпадіння з одним документом 4.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 16%**

	Документ		Сумарний збіг по Базі Даних	
ID: 271508 Назва: МКР Кіберфізична система управління квадрокоптерами для розпізнавання об'єктів за допомогою машинного навчання Додано в БД: 2026-05-14 Автора: Ілля ГРІЦАЙ Керівники: Євген ФЕДОРОВ Консультанти: Опоненти:	Символи	Лексеми	Символи	Лексеми
	109103	932	10234 (9%)	108 (12%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми