

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)
Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та
робототехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)
Розроблення програмного модуля розпізнавання об'єктів для системи
комп'ютерного зору FPV дрону
(тема)

Виконав:
здобувач 2 року навчання,
групи КТРСМ-24-2
Ілля ПАРА
(власне ім'я, прізвище)

Спеціальність 174 Автоматизація,
комп'ютерно-інтегровані технології
та робототехніка
(код і повна назва спеціальності)
Тип програми освітньо-професійна
Освітня програма Комп'ютеризовані
та робототехнічні системи
(повна назва освітньої програми)

Керівник проф. Владислав ЄВССЄВ
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри КІТАР

(підпис)

Ігор НЕВЛЮДОВ

(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та
робототехніки
Рівень вищої освіти другий (магістерський)
Спеціальність 174 Автоматизація, комп'ютерно-інтегровані технології та
робототехніка
(код і повна назва)
Тип програми освітньо-професійна
Освітня програма Комп'ютеризовані та робототехнічні системи
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«01» вересня 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

здобувачеві _____

Пара Іллі Ігоровичу

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розроблення програмного модуля розпізнавання об'єктів
для системи комп'ютерного зору FPV дрону»

затверджена наказом університету від 10.11.2025 р. № 1018 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 23.12.2025 р.

3. Вихідні дані до роботи 3.1 Розпізнавання об'єктів на базі потокового
відео, отриманого з камер FPV дрону

4. Перелік питань, що потрібно опрацювати в роботі 4.1 Вступ; 4.2 Аналіз
систем комп'ютерного зору First Person View, які використовуються на дронах;
4.3 Аналіз методів передачі потокового відео в системах комп'ютерного зору
First Person View; 4.4 Аналіз апаратного забезпечення систем First Person View;
4.5 Аналіз нейронних мереж в системах First Person View; 4.6 Розробка
математичної моделі опису відопотоку з камери First Person View; 4.7 Розробка
програми для моделювання відстеження об'єктів; 4.8 Аналіз отриманих
результатів моделювання; 4.9 Обґрунтування та вибір середовища та мови
програмування; 4.10 Розробка алгоритму роботи системи ідентифікації та
розпізнавання об'єктів; 4.11 Реалізація функцій ідентифікації та розпізнавання з
використанням нейронної мережі; 4.12 Проведення експериментів та аналіз
отриманих результатів; 4.13 Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Демонстраційний матеріал, представлений у форматі презентації PowerPoint (*.ppt). – 14 с.ф. А 4.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз сучасних систем комп'ютерного зору FPV	02.09.2025-15.09.2025	виконано
2	Розробка математичної моделі опису відопотоку з камери FPV	16.09.2025-26.09.2025	виконано
3	Розробка програми для моделювання відстеження об'єктів	27.09.2025-05.10.2025	виконано
4	Аналіз отриманих результатів моделювання	06.10.2025-11.10.2025	виконано
5	Обґрунтування та вибір середовища та мови програмування	12.10.2025-24.10.2025	виконано
6	Розробка алгоритму роботи системи ідентифікації та розпізнавання об'єктів	25.10.2025-08.11.2025	виконано
7	Реалізація функцій ідентифікації та розпізнавання з використанням нейронної мережі	09.11.2025-10.11.2025	виконано
8	Проведення експериментів та аналіз отриманих результатів	11.11.2025-28.11.2025	виконано
11	Оформлення пояснювальної записки	29.11.2025-07.12.2025	виконано

Дата видачі завдання 01.09.2025 р.

Здобувач _____

Керівник роботи _____

(підпис)

Ілля ПАРА

(власне ім'я, прізвище)

проф. Владислав ЄВСЄЄВ

(посада, власне ім'я, прізвище)

Я, Пара Ілля Ігорович, як здобувач вищої освіти ХНУРЕ, розумію і підтримую політику закладу з академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Я не використовував(ла) штучний інтелект для підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

" 22 " грудня 2025 р.

A handwritten signature in dark blue ink, appearing to read 'Ilya Para', written in a cursive style.

Ілля ПАРА

РЕФЕРАТ

Пояснювальна записка: 90 с., 2 табл., 23 рис., 3 дод., 21 джерело.

FPV ДРОН, РОЗПІЗНАВАННЯ ОБ'ЄКТІВ, КОМП'ЮТЕРНИЙ ЗІР, ПОТОКОВЕ ВІДЕО, ІДЕНТИФІКАЦІЯ, АЛГОРИТМИ ОБРОБКИ ЗОБРАЖЕНЬ, МОДЕЛЮВАННЯ.

Мета роботи – підвищення ефективності та швидкості розпізнавання об'єктів на базі потокового відео, отриманого з камер FPV дрону.

Об'єкт дослідження – процес розпізнавання та ідентифікації об'єктів для безпілотних літаючих апаратів.

Предмет дослідження – моделі, методи, алгоритмічне та програмне забезпечення розпізнавання та ідентифікації об'єктів.

В кваліфікаційній роботі було проведено аналіз сучасних систем комп'ютерного зору FPV. Розроблено математичну модель опису відопотоку з камери FPV. Розроблено програму для моделювання відстеження об'єктів та проаналізовано отримані результати моделювання. Проведено обґрунтування та вибір середовища та мови програмування, розроблено алгоритм роботи системи ідентифікації та розпізнавання об'єктів та реалізовано функції ідентифікації та розпізнавання з використанням нейронної мережі. Проведено експерименти та аналіз отриманих результатів.

Отримані результати кваліфікаційної роботи можна віднести до Цілі сталого розвитку 9 «Промисловість, інновації та інфраструктура», а саме 9.5. Створити фінансову та інституційну системи (інноваційну інфраструктуру), що забезпечуватимуть розвиток наукових досліджень та науковотехнічних (експериментальних) розробок.

ABSTRACT

Explanatory note: 90 p., 2 tables, 23 figures, 3 appendices, 21 sources.

FPV DRONE, OBJECT RECOGNITION, COMPUTER VISION, STREAMING VIDEO, IDENTIFICATION, IMAGE PROCESSING ALGORITHMS, MODELING.

The purpose of the work is to increase the efficiency and speed of object recognition based on streaming video obtained from FPV drone cameras.

The object of the study is the process of object recognition and identification for unmanned aerial vehicles.

The subject of the study is models, methods, algorithmic and software for object recognition and identification.

In the qualification work, an analysis of modern FPV computer vision systems was conducted. A mathematical model for describing the video stream from an FPV camera was developed. A program for object tracking modeling was developed and the obtained modeling results were analyzed. The justification and selection of the environment and programming language were carried out, the algorithm for the object identification and recognition system was developed and the identification and recognition functions were implemented using a neural network. Experiments were conducted and the results obtained were analyzed.

The obtained results of the qualification work can be attributed to Sustainable Development Goal 9 "Industry, Innovation and Infrastructure", namely 9.5. Create a financial and institutional system (innovation infrastructure) that will ensure the development of scientific research and scientific and technical (experimental) developments.

ЗМІСТ

Перелік умовних скорочень	9
Вступ.....	10
1 Аналіз сучасних систем комп'ютерного зору first person view.....	12
1.1 Аналіз систем комп'ютерного зору First Person View, які використовуються на дронах	12
1.2 Аналіз методів передачі потокового відео в систем комп'ютерного зору First Person View	14
1.3 Аналіз апаратного забезпечення систем First Person View.....	17
1.4 Аналіз нейронних мереж в системах First Person View	19
1.5 Постановка задач досліджень	21
2 Розробка математичних моделей системи ідентифікації та розпізнавання об'єктів.....	22
2.1 Розробка математичної моделі опису відопотоку з камери First Person View	22
2.2 Розробка програми для моделювання відстеження об'єктів.....	32
2.3 Аналіз отриманих результатів моделювання	36
2.4 Висновки до 2 розділу	44
3 Розробка системи ідентифікації та розпізнавання об'єктів для системи комп'ютерного зору FPV дрону	46
3.1 Обґрунтування та вибір середовища та мови програмування.....	46
3.2 Розробка алгоритму роботи системи ідентифікації та розпізнавання об'єктів.....	47
3.3 Реалізація функцій ідентифікації та розпізнавання з використанням нейронної мережі.....	49
3.4 Проведення експериментів та аналіз отриманих результатів	52
3.5 Охорона праці	57
3.6 Висновки до 3 розділу	58
Висновки	59

Перелік джерел посилання	60
Додаток А Код численного моделювання відстеження об'єктів	63
Додаток Б Апробація результатів кваліфікаційної роботи	80
Додаток В Демонстраційний матеріал	89

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

CMOS – complementary metal-oxide-semiconductor;

FPV – first person view;

GPU – graphics processing unit;

IoU – intersection over union, показник перекриття об'єктів;

LTE – long term evolution;

NMS – non-maximum suppression;

PID – proportional–integral–derivative;

SoC – спеціалізована система-на-кристалі.

ВСТУП

Сучасні безпілотні літальні апарати активно використовуються у багатьох сферах, включаючи аерофотозйомку, моніторинг об'єктів, пошуково-рятувальні операції та інспекцію важкодоступних територій. Для підвищення ефективності їх роботи важливим є забезпечення швидкого та точного розпізнавання об'єктів у режимі реального часу. Технології комп'ютерного зору та нейронних мереж дозволяють автоматизувати процес аналізу відеопотоку з камер FPV-дронів, зменшуючи навантаження на оператора та підвищуючи безпеку й автономність польоту, проте існуючі підходи часто обмежені низькою швидкістю, високими вимогами до обчислювальних ресурсів та нестабільністю роботи в умовах змінного освітлення або шумів у відеопотоці. Саме тому, розроблення спеціалізованого програмного модуля для ідентифікації та розпізнавання об'єктів у FPV-системах є актуальним завданням, оскільки воно дозволяє підвищити точність та швидкість обробки даних, забезпечити стабільність роботи системи у складних умовах і сприяє розвитку автономних та напіваавтономних безпілотних технологій.

Мета роботи – підвищення ефективності та швидкості розпізнавання об'єктів на базі потокового відео, отриманого з камер FPV дрону.

Об'єкт дослідження – процес розпізнавання та ідентифікації об'єктів для безпілотних літаючих апаратів.

Предмет дослідження – моделі, методи, алгоритмічне та програмне забезпечення розпізнавання та ідентифікації об'єктів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проведено аналіз систем комп'ютерного зору First Person View, які використовуються на дронах;
- проведено аналіз методів передачі потокового відео в систем комп'ютерного зору First Person View;
- проведено аналіз апаратного забезпечення систем First Person View;
- проведено аналіз нейронних мереж в системах First Person View;

- розробити математичну модель опису відопотоку з камери First Person View;
- розробити програму для моделювання відстеження об'єктів;
- проведено аналіз отриманих результатів моделювання;
- проведено аналіз обґрунтування та вибір середовища та мови програмування;
- розробити алгоритм роботи системи ідентифікації та розпізнавання об'єктів;
- реалізовано функції ідентифікації та розпізнавання з використанням нейронної мережі;
- проведено експерименти та аналіз отриманих результатів.

Кваліфікаційна робота виконана згідно ДСТУ 3008–15 [1], методичних вказівок [2], також, отримали апробацію на науковій конференції [3].

1 АНАЛІЗ СУЧАСНИХ СИСТЕМ КОМП'ЮТЕРНОГО ЗОРУ FIRST PERSON VIEW

1.1 Аналіз систем комп'ютерного зору First Person View, які використовуються на дронах

Системи комп'ютерного зору First Person View (FPV) є ключовим елементом сучасних безпілотних літальних апаратів, оскільки забезпечують оператору або бортовому обчислювальному модулю візуальну інформацію з перспективи «першої особи» в режимі реального часу, такі системи поєднують апаратні засоби захоплення зображень, канали передачі відеопотоку та програмні модулі обробки і аналізу візуальних даних, що дозволяє реалізувати задачі навігації, спостереження, розпізнавання та ідентифікації об'єктів [4].

Апаратна складова FPV-систем комп'ютерного зору зазвичай представлена мініатюрними камерами з широким кутом огляду, які здатні працювати в умовах змінного освітлення та високих динамічних навантажень. Найпоширенішими є CMOS-камери з роздільною здатністю від 720p до 4K, які забезпечують баланс між якістю зображення та обчислювальними витратами. Важливим фактором є частота кадрів, що для FPV-дронів може досягати 60–120 кадрів за секунду, оскільки затримки у відеопотоці безпосередньо впливають на ефективність керування та точність розпізнавання об'єктів. Передача відеоданих у FPV-системах здійснюється за допомогою аналогових або цифрових каналів зв'язку. Аналогові системи вирізняються мінімальною затримкою, однак мають нижчу якість зображення та схильність до завад, а цифрові FPV-системи, зокрема на базі Wi-Fi, LTE або спеціалізованих радіопротоколів, забезпечують вищу роздільну здатність та стабільність сигналу, але потребують ефективних алгоритмів стиснення та декодування відео. Для задач комп'ютерного зору особливо важливою є мінімізація затримок та втрат інформації під час передачі відеопотоку [5].

Програмне забезпечення систем комп'ютерного зору FPV-дронів включає етапи попередньої обробки зображень, аналізу сцени та розпізнавання об'єктів. На етапі попередньої обробки застосовуються методи фільтрації шумів, корекції спотворень об'єктива, нормалізації яскравості та контрасту. Ці операції дозволяють підвищити якість вхідних даних і створити сприятливі умови для подальшого застосування алгоритмів машинного навчання та глибоких нейронних мереж.

У сучасних FPV-системах все ширше використовуються алгоритми комп'ютерного зору на основі глибокого навчання, зокрема згорткові нейронні мережі для детекції та класифікації об'єктів. Такі підходи дозволяють у реальному часі виявляти людей, транспортні засоби, перешкоди та інші об'єкти. Однак впровадження нейронних мереж на борту дрона пов'язане з обмеженнями обчислювальних ресурсів, енергоспоживання та масогабаритних характеристик апаратної платформи, що зумовлює необхідність оптимізації моделей і алгоритмів. Окрему увагу слід приділити архітектурі обробки даних FPV-систем комп'ютерного зору, можливі два основні підходи: бортова обробка відео та обробка на наземній станції, яка забезпечує автономність дрона та зменшення залежності від каналу зв'язку, однак потребує використання високопродуктивних вбудованих обчислювальних платформ та обробка на наземній станції дозволяє застосовувати більш складні алгоритми, проте є чутливою до затримок і втрати зв'язку [6]. Для простоти представлення, наведемо порівняння цих підходів у вигляді таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика двох підходів до архітектури обробки даних у FPV-системах комп'ютерного зору дронів

Критерій порівняння	Бортова обробка відео	Обробка на наземній станції
1	2	3
Місце обробки даних	Безпосередньо на борту FPV-дрона	На зовнішній наземній обчислювальній станції
Автономність системи	Висока автономність, дрон може працювати незалежно від стабільності каналу зв'язку	Низька автономність, робота системи залежить від якості зв'язку

Продовження таблиці 1.1

1	2	3
Затримка обробки	Мінімальна затримка, що важливо для керування в реальному часі	Можливі значні затримки через передачу відеопотоку
Вимоги до каналу зв'язку	Менші вимоги, передаються лише результати обробки	Високі вимоги, необхідна передача потокового відео високої якості
Обчислювальні ресурси	Обмежені ресурсами вбудованих платформ	Практично не обмежені, можливе використання потужних GPU
Складність алгоритмів	Обмежена, потребує оптимізованих та полегшених моделей	Можливе використання складних та ресурсомістких алгоритмів
Енергоспоживання	Збільшене навантаження на акумулятор дрона	Не впливає на енергоспоживання дрона
Стійкість до втрати зв'язку	Висока, система продовжує функціонувати	Низька, при втраті зв'язку обробка стає неможливою
Сфера застосування	Автономні FPV-дрони, навігація, уникнення перешкод	Дослідницькі задачі, моніторинг, аналітичні системи

Системи комп'ютерного зору FPV-дронів характеризуються складною взаємодією апаратних та програмних компонентів, а їх ефективність значною мірою залежить від швидкодії алгоритмів обробки відеопотоку. Основними проблемами залишаються забезпечення роботи в реальному часі, стійкість до змін умов освітлення та обмеженість обчислювальних ресурсів. Це обґрунтовує доцільність розроблення спеціалізованого програмного модуля розпізнавання об'єктів, оптимізованого для потокового відео FPV-дрону, що відповідає меті та завданням даної кваліфікаційної роботи [7].

1.2 Аналіз методів передачі потокового відео в систем комп'ютерного зору First Person View

Передача потокового відео в системах комп'ютерного зору є критично важливим компонентом, оскільки безпосередньо впливає на затримку, якість зображення та ефективність алгоритмів розпізнавання об'єктів. У FPV-дронах застосовуються різні методи передачі відеоданих, кожен з яких має власні

переваги та обмеження. Проведемо аналіз, який дозволить оцінити доцільність використання того чи іншого підходу залежно від умов експлуатації та вимог системи комп'ютерного зору.

Аналогові методи передачі відео традиційно використовуються у класичних FPV-системах завдяки надзвичайно малій затримці сигналу. Відеопотік передається у реальному часі без складних процедур кодування та декодування, що забезпечує швидку реакцію оператора або системи керування, проте аналоговий сигнал характеризується низькою роздільною здатністю, чутливістю до електромагнітних завад та поступовою деградацією якості зображення зі збільшенням відстані, що негативно впливає на точність алгоритмів комп'ютерного зору [8].

Цифрові методи передачі відео, зокрема на базі стандартів Wi-Fi, спеціалізованих FPV-протоколів або стільникових мереж, забезпечують значно вищу якість зображення та стабільність сигналу. Використання алгоритмів стиснення дозволяє передавати відео з високою роздільною здатністю, що є важливим для задач детекції та ідентифікації об'єктів, однак цифрові системи потребують додаткового часу на кодування, передачу та декодування відео, що призводить до зростання затримок і може обмежувати їх застосування в динамічних FPV-сценаріях.

Гібридні методи передачі відео поєднують переваги аналогових і цифрових підходів. У таких системах низькороздільне відео або телеметричні дані можуть передаватися з мінімальною затримкою, тоді як високоякісний цифровий потік використовується для детального аналізу та запису, що дозволяє знизити загальну затримку керування та одночасно забезпечити достатню якість зображення для алгоритмів комп'ютерного зору. Недоліком даного методу є підвищена складність апаратної та програмної реалізації [9].

З позиції систем комп'ютерного зору FPV-дронів ключовими критеріями оцінки методів передачі відео є затримка, пропускна здатність каналу, стійкість до завад та вплив на обчислювальні ресурси. Аналогові системи демонструють мінімальну затримку, проте обмежують можливості точного розпізнавання через низьку якість зображення. Цифрові методи, навпаки, забезпечують високу

інформативність відеопотоку, але потребують оптимізації для зменшення затримок. Гібридні підходи є компромісним рішенням, що дозволяє адаптувати систему до різних умов експлуатації. Розглянемо порівняння існуючих методів передачі потокового відео в FPV-системах комп'ютерного зору у вигляді таблиці з основними з чисельними характеристиками, які найчастіше використовуються для інженерного та наукового аналізу [10].

Таблиця 1.2 – Порівняння методів передачі потокового відео в FPV-системах комп'ютерного зору

Критерій	Аналогова передача відео	Цифрова передача відео	Гібридна передача відео
Середня затримка сигналу, мс	5 – 20	80 – 250	20 – 80
Роздільна здатність відео	480p – 576p	720p – 4K	480p – 1080p
Частота кадрів, кадр/с	25 – 60	30 – 120	30 – 90
Пропускна здатність каналу, Мбіт/с	5 – 10	10 – 50	8 – 30
Якість зображення (SNR), дБ	25 – 35	35 – 45	30 – 40
Стійкість до завад	Низька	Висока	Середня
Чутливість до втрати пакету	Відсутня	Висока	Середня
Навантаження на обчислювальні ресурси	Мінімальне	Високе (кодування/декодування)	Середнє
Споживання енергії, Вт	1 – 2	3 – 8	2 – 5
Дальність стабільної передачі, км	1 – 5	2 – 10	2 – 7
Придатність для CV-алгоритмів	Низька	Висока	Середня–висока
Придатність для автономних дронів	Середня	Висока	Висока

Чисельний аналіз підтвердив, що аналогова передача відео забезпечує мінімальну затримку, однак істотно обмежує точність алгоритмів комп'ютерного зору через низьку роздільну здатність та нестабільну якість зображення, цифрові методи передачі демонструють найкращі показники якості відеопотоку та є найбільш придатними для задач розпізнавання та ідентифікації

об'єктів, проте потребують значних обчислювальних ресурсів і мають більші затримки, а гібридні системи забезпечують компроміс між швидкістю та інформативністю відео і є перспективним напрямом для FPV-дронів з елементами автономного комп'ютерного зору. В цілому, вибір методу передачі потокового відео для FPV-систем комп'ютерного зору повинен здійснюватися з урахуванням поставлених задач, для автономних та швидкісних FPV-дронів доцільним є використання методів з мінімальною затримкою та частковою бортовою обробкою відео, а для систем, орієнтованих на високоточне розпізнавання та аналітичну обробку, більш ефективними є цифрові або гібридні методи передачі відеоданих, що забезпечують баланс між швидкістю та якістю зображення [11].

1.3 Аналіз апаратного забезпечення систем First Person View

Системи FPV, що застосовуються на безпілотних літальних апаратах, повинні відповідати комплексу вимог, обумовлених особливостями експлуатації в реальному часі. До основних потреб такої системи належать мінімальна затримка відеосигналу, стабільність роботи в умовах обмеженого каналу зв'язку, висока якість зображення для коректної роботи алгоритмів комп'ютерного зору та низьке енергоспоживання. Водночас апаратне забезпечення FPV-систем має бути компактним, легким і стійким до вібрацій, перепадів температур та електромагнітних завад. Проблематика реалізації полягає у необхідності досягнення балансу між продуктивністю обчислювальних модулів, якістю відеопотоку та обмеженими ресурсами бортового живлення, що часто призводить до компромісних інженерних рішень і ускладнює впровадження ресурсомістких алгоритмів розпізнавання об'єктів [12].

Ключовим елементом FPV-системи є камера, яка забезпечує захоплення зображення з перспективи першої особи. У сучасних дронах переважно застосовуються CMOS-камери, що відзначаються низьким енергоспоживанням, високою швидкістю зчитування та компактними розмірами. Для FPV-застосувань важливими параметрами є роздільна здатність, частота кадрів, кут

огляду та світлочутливість сенсора. Ширококутні об'єктиви з кутом огляду 120–170° дозволяють охоплювати значну частину простору перед дроном, однак потребують корекції геометричних спотворень під час програмної обробки.

Обчислювальний модуль виконує функції попередньої обробки відео, кодування потоку та, у випадку бортової реалізації комп'ютерного зору, запуск алгоритмів розпізнавання об'єктів. У FPV-дронах можуть використовуватися як мікроконтролери, так і вбудовані одноплатні комп'ютери або спеціалізовані системи-на-кристалі (SoC). Вибір платформи визначається вимогами до продуктивності, масогабаритних характеристик та енергоспоживання. Для задач реального часу особливо важливою є наявність апаратного прискорення обробки відео та підтримки нейронних обчислень [13].

Модулі передачі відеосигналу є ще одним критичним компонентом апаратного забезпечення FPV-систем. Вони можуть бути реалізовані у вигляді аналогових відеопередавачів або цифрових трансиверів, що підтримують різні протоколи зв'язку. Основними апаратними характеристиками таких модулів є вихідна потужність, робочий діапазон частот, стабільність сигналу та сумісність з приймальними пристроями. Для цифрових FPV-систем додатково важливими є можливості апаратного кодування та декодування відео.

Антени відіграють важливу роль у забезпеченні стабільності та дальності FPV-зв'язку. Залежно від конфігурації системи застосовуються всеспрямовані або спрямовані антени, які впливають на діаграму спрямованості та рівень завадостійкості. Неправильний вибір або розташування антен може призвести до значних втрат сигналу та зниження якості відеопотоку, що негативно позначається на роботі систем комп'ютерного зору [14].

Важливим аспектом апаратного забезпечення FPV-систем є система живлення. Відеокамера, обчислювальний модуль та передавачі створюють додаткове навантаження на акумулятор дрона, що зменшує тривалість польоту. Тому апаратні компоненти повинні характеризуватися високою енергоефективністю та стабільністю роботи при змінних рівнях напруги, а також необхідно враховувати електромагнітну сумісність компонентів для уникнення взаємних завад. Ефективність роботи FPV-дрона значною мірою визначається

збалансованістю між якістю відеозахоплення, обчислювальною продуктивністю та характеристиками каналу передачі даних. Обмеження за масою, енергоспоживанням і габаритами зумовлюють необхідність ретельного вибору апаратних компонентів та оптимізації програмних алгоритмів, що підтверджує актуальність розроблення програмного модуля розпізнавання об'єктів, адаптованого до реальних апаратних можливостей FPV-дронів, що відповідає меті даної кваліфікаційної роботи [15].

1.4 Аналіз нейронних мереж в системах First Person View

Системи FPV, що використовуються на безпілотних літальних апаратах, функціонують у динамічному середовищі та потребують швидкої й надійної обробки візуальної інформації в реальному часі. У зв'язку з цим до нейронних мереж, які застосовуються в FPV-системах, висувається низка специфічних вимог. Основними потребами є висока швидкість обробки відеопотоку, стійкість до змін освітлення та ракурсів зйомки, здатність працювати з шумними та частково спотвореними зображеннями, а також помірні вимоги до обчислювальних ресурсів і енергоспоживання. Крім того, нейронна мережа повинна забезпечувати достатню точність розпізнавання об'єктів за умов обмеженої роздільної здатності та можливих втрат якості відеосигналу.

У FPV-системах нейронні мережі можуть виконувати широкий спектр задач комп'ютерного зору, до основних з них належать детекція об'єктів у кадрі, їх класифікація та ідентифікація, відстеження рухомих цілей, оцінка відстані до об'єктів, а також сегментація зображення для виділення перешкод або зон інтересу. У більш складних системах нейронні мережі можуть застосовуватися для задач навігації, уникнення зіткнень і підтримки автономного польоту. Актуальність використання нейронних мереж у FPV-дронах зумовлена їх здатністю автоматично виявляти складні візуальні ознаки та адаптуватися до змінних умов середовища, що практично неможливо реалізувати за допомогою класичних алгоритмів обробки зображень [16].

Серед існуючих типів нейронних мереж у системах FPV найбільш поширеними є згорткові нейронні мережі, які забезпечують ефективну обробку зорових даних. Класичні архітектури, орієнтовані на класифікацію зображень, відзначаються високою точністю, однак мають значну обчислювальну складність і обмежену придатність для роботи в реальному часі на борту дрона. Для задач детекції об'єктів у FPV-системах частіше застосовуються одноетапні моделі, що поєднують процес виявлення та класифікації в одному циклі обробки, забезпечуючи кращий баланс між швидкістю та точністю.

Наступними є двоетапні моделі розпізнавання об'єктів, які характеризуються високою точністю локалізації, оскільки на першому етапі формують пропозиції областей інтересу, а на другому виконують детальну класифікацію. Проте такі підходи вимагають значних обчислювальних ресурсів і, як правило, застосовуються у наземних станціях або офлайн-аналізі відеоданих. Одноетапні нейронні мережі, навпаки, більш придатні для FPV-дронів, оскільки дозволяють досягти низької затримки та стабільної роботи в режимі реального часу, що є критично важливим для систем комп'ютерного зору першої особи [17].

Окрему категорію становлять полегшені та оптимізовані нейронні мережі, розроблені спеціально для вбудованих систем. Вони використовують зменшену кількість параметрів, квантування вагових коефіцієнтів та апаратне прискорення обчислень. Такі мережі демонструють дещо нижчу точність порівняно з повнорозмірними моделями, однак забезпечують прийнятний рівень розпізнавання за значно менших витрат обчислювальних ресурсів, що робить їх перспективними для FPV-застосувань.

Процес використання нейронних мереж у FPV-системах комп'ютерного зору включає декілька основних етапів. На першому етапі здійснюється формування та підготовка навчальної вибірки, яка повинна містити зображення, максимально наближені до реальних умов польоту дрона. Далі виконується навчання нейронної мережі з використанням відповідних алгоритмів оптимізації та функцій втрат, а вже після цього модель проходить етап тестування та валідації для оцінки точності та стійкості до різних умов зйомки. Заключним

етапом є інтеграція навченої мережі у програмне забезпечення FPV-дрона та її оптимізація для роботи в реальному часі. Використання нейронних мереж у системах First Person View є невід'ємною складовою сучасних систем комп'ютерного зору для дронів. Вибір конкретної архітектури та методів оптимізації повинен ґрунтуватися на вимогах до швидкодії, точності та апаратних обмеженнях FPV-платформи, що безпосередньо узгоджується з метою даної кваліфікаційної роботи [18].

1.5 Постановка задач досліджень

Для досягнення мети кваліфікаційної роботи, яка полягає у підвищенні ефективності та швидкості розпізнавання об'єктів на базі потокового відео FPV-дрону, у процесі дослідження необхідно виконати наступні завдання:

- розробити математичну модель відеопотоку, що надходить з камери First Person View, з урахуванням його часових, просторових та шумових характеристик, що дозволить формалізувати процес отримання та обробки зорової інформації в умовах реального часу;

- розробити програмну модель для відстеження об'єктів у відеопотоці, яка забезпечує аналіз динаміки руху об'єктів та їх стійке виявлення в послідовності кадрів. На основі результатів моделювання необхідно виконати аналіз ефективності обраних методів та оцінити їх придатність для використання у FPV-системах;

- побудова алгоритму роботи системи ідентифікації та розпізнавання об'єктів. Окремою задачею є реалізація функцій розпізнавання з використанням нейронної мережі, адаптованої до обмежень апаратного забезпечення FPV-дрону;

- постановка та проведення експериментів для перевірки працездатності розробленої системи, аналіз отриманих результатів за показниками точності та швидкодії, а також формулювання висновків щодо ефективності запропонованого підходу.

2 РОЗРОБКА МАТЕМАТИЧНИХ МОДЕЛЕЙ СИСТЕМИ ІДЕНТИФІКАЦІЇ ТА РОЗПІЗНАВАННЯ ОБ'ЄКТІВ

2.1 Розробка математичної моделі опису відопотоку з камери First Person View

Відопотік камери First Person View розглядається як дискретизований у часі просторово-часовий процес інтенсивностей, що формується оптикою, сенсором та трактом стиснення, і зазнає впливу шумів, спотворень лінзи, розмиття руху та затримок передачі. Для подальшого керування та навігації дрона цей потік має бути приведений до узгоджених систем координат через калібрування й проєкції, із врахуванням параметрів внутрішньої та зовнішньої геометрії камери. Ключовими є моделювання часової дискретизації й асинхроній, компенсування латентності, а також статистичний опис невизначеності вимірювань. Така постановка дозволяє побудувати узагальнену математичну схему перетворень від “сирого” кадру до метрично інтерпретованих ознак. У підсумку це створює підґрунтя для розроблення математичних моделей опису відопотоку FPV, включно з моделями проєкції, шуму, згладжування, відстеження та компенсації затримок.

Формалізуємо дані отримані з FPV , як дискретизацію безперервного відєопотоку камерою дрона з урахуванням пропуску кадрів (frame skipping), що напряду впливає на затримку, частоту оновлення оцінок та стабільність керування. Нехай безперервний потік інтенсивностей $I(x, y, t)$ дискретизується в часі з періодом $\Delta t = 1/FPS$:

$$\mathbf{X}_k = I(\cdot, \cdot, t_k), \quad t_k = k\Delta t, \quad k \in \mathbb{N}, \quad (2.1)$$

де $I(\cdot, \cdot, t_k)$ – безперервна функція інтенсивності зображення у момент часу t , яка описує яскравість кожної точки просторових координат (пікселів) у потоці відео,

тобто це фактично це тривимірне поле $I(x, y, t)$, де x, y - координати пікселя, а t – час;

X_k – дискретизоване зображення (кадр) відеопотоку, отримане у момент часу t_k , що є матрицею інтенсивностей або кольорових значень, що становить миттєвий стан відеопотоку;

t_k – дискретні часові відмітки, які визначають моменти зйомки кадрів, при умові що кожен наступний кадр формується через інтервал Δt ;

Δt – період дискретизації або часовий інтервал між двома послідовними кадрами, що обернено пропорційний частоті кадрів $\Delta t = 1/FPS$;

$k \in \mathbb{N}$ – натуральний індекс кадру, який нумерує послідовність дискретних моментів часу, тобто $k = 0, 1, 2, \dots$

Вираз (2.1) дає можливість описати перехід від безперервного потоку світлових інтенсивностей $I(x, y, t)$ до послідовності дискретних кадрів X_k , що є основою для математичного моделювання цифрового відеопотоку FPV-системи.

Для зменшення роздільної здатності для прискорення моделі та приведення до цілочисельного типу, як того вимагає SavedModel. SavedModel – це універсальний формат збереження навченої моделі у середовищі TensorFlow, який містить як архітектуру нейронної мережі, так і всі її параметри (ваги, граф обчислень, сигнатури викликів та метадані), необхідні для подальшого використання без повторного навчання. Нехай початковий кадр $X_k \in \mathbb{R}^{H \times W \times 3}$. Припустимо, що початковий кадр ресайзиться афінним перетворенням до $\tilde{X}_k \in \mathbb{R}^{h \times w \times 3}$, $(h, w) = (320, 320)$:

$$\tilde{X}_k(u, v) = X_k(\alpha u, \beta v), \alpha = H/h, \beta = W/w, \quad (2.2)$$

де $X_k(x, y)$ – оригінальний кадр відеопотоку з камери FPV, визначений у координатах вихідного зображення розміром $H \times W$;

$\tilde{X}_k(u, v)$ – масштабована версія кадру X_k , яка приведена до розміру $h \times w$, що відповідає вхідному формату нейронної мережі;

$\alpha = H/h$ – коефіцієнт масштабування по вертикалі, який показує, скільки пікселів оригінального зображення відповідає одному пікселю у зменшеному кадрі;

$\beta = W/w$ – коефіцієнт масштабування по горизонталі, що визначає співвідношення між шириною оригінального кадру та новим розміром;

(u, v) – координати пікселя у масштабованому зображенні, а $(\alpha u, \beta v)$ – відповідна точка в оригінальному кадрі.

Модель (2.2) знижує обчислювальне навантаження, зберігає пропорції об'єктів та забезпечує коректне позиціонування детектованих областей при подальшій реконструкції координат у вихідному масштабі. Далі відбувається квантування:

$$\hat{X}_k = \text{round}(\tilde{X}_k) \in \{0, \dots, 255\}^{h \times w \times 3}, \quad (2.3)$$

де \tilde{X}_k – масштабований кадр відеопотоку (з моделі 2.2), значення якого можуть бути дійсними числами в діапазоні $[0, 255]$ або $[0, 1]$, залежно від методу інтерполяції. Містить тривимірну структуру піксельних значень розміром $h \times w \times 3$, де 3 – кількість колірних каналів (RGB);

\hat{X}_k – результат операції округлення (дискретизації) кожного елемента \tilde{X}_k до найближчого цілого значення в межах множини $\{0, 1, 2, 3, \dots, 255\}$ тобто це цифрове зображення у форматі, придатному для обчислень у TensorFlow або інших фреймворках машинного зору;

$\text{round}(\cdot)$ – функція квантування (округлення), яка переводить неперервні або дробові інтенсивності в цілочисельний діапазон рівнів яскравості, та забезпечує узгодженість представлення з реальними сенсорними даними камери, де пікселі мають дискретну глибину кольору (8 біт);

$\{0, \dots, 255\}$ – множина можливих рівнів інтенсивності для кожного каналу кольору, що відповідає 8-бітному кодуванню RGB-компонентів;

$h, w, 3$ – просторові розміри кадру після масштабування, відповідно h – висота, w – ширина, а 3 це три канали кольору (Red, Green, Blue).

Модель (2.3) дає забезпечує уніфікацію числового формату піксельних даних, зменшує похибки, пов'язані з обчислювальною інтерполяцією, і гарантує сумісність з апаратними моделями, де кожен кадр повинен мати чітко визначену дискретну глибину кольору. Це дозволяє нейронній мережі отримувати дані в оптимальному вигляді для точного розпізнавання об'єктів у режимі реального часу.

Для того зв'язати піксельні координати з напрямками променів/просторовими координатами не обхідно описати Пінохольну модель камери, що необхідно для оцінки реальних відстаней та навігаційних рішень дрона. Класична Пінохольна модель камери (геометрія проєкції) має наступний вигляд:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K}[\mathbf{R}|t] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.4)$$

де \mathbf{K} – внутрішні параметри;

$\mathbf{R}|t$ – положення камери відносно світу;

X, Y, Z – 3D-точка;

u, v – піксель.

Тепер необхідно розробити стохастичну модель детектора об'єктів використовуючи MobileNet, це дає можливість зрозуміти що повертає модель розпізнавання: набори боксів, класів і ймовірностей – як випадкові величини.

Тобто для кожного кандидата i на виході маємо:

– клас $C_i \in \{1, \dots, K\}$;

– нормалізований бокс $b_i = [y_1, x_1, y_2, x_2] \in [0,1]^4$;

– оцінку достовірності $s_i = P(C_i | \hat{X}_k) \in [0,1]$.

Що можна описати у векторній формі:

$$D_k = \{(C_i, b_i, s_i)\}_{i=1}^{N_k}, \quad (2.5)$$

де D_k – множина детекцій (результатів розпізнавання) у кадрі k що описує всі об’єкти, виявлені моделлю нейронної мережі у цей момент часу;

N_k – кількість об’єктів, які були виявлені на кадрі X_k , визначається як кількість трійок C_i, b_i, s_i ;

C_i – ідентифікатор (клас) i -го об’єкта, який відповідає певній категорії (наприклад, людина, автомобіль, тварина, дрон, будівля). У межах системи FPV-дрону цей параметр дозволяє розрізнити типи об’єктів для вибору відповідної стратегії реакції або траєкторії польоту;

$b_i = [y_1, x_1, y_2, x_2]$ – вектор координат граничного прямокутника (bounding box) для i -го об’єкта, який визначає його положення в межах кадру;

s_i – оцінка впевненості (score), тобто ймовірність того, що виявлений об’єкт дійсно належить до класу C_i . Значення s_i визначається як результат застосування сигмоїдної або софтмакс-функції на виході нейронної мережі. У задачі комп’ютерного зору FPV-дрону цей параметр використовується для фільтрації хибних детекцій і визначення пріоритету об’єктів для подальшої обробки.

Модель (2.5) формалізується результат роботи нейронної мережі-детектора об’єктів у момент часу t_k , який містить набір знайдених об’єктів у кадрі X_k та описує структуру даних, що поєднує інформацію про класифікацію, локалізацію та впевненість детектора для кожного розпізнаного елемента сцени.

Введемо порогову фільтрацію для відкидання малоїмовірні детекції для зменшення помилкових спрацьовувань у реальному часі. Представимо це наступним виразом:

$$D_k^{(\tau)} = \{(C_i, b_i, s_i) \in D_k | s_i \geq \tau\}, \quad (2.5)$$

де τ – поріг впевненості, при умові більший $\tau \rightarrow$ менше FP, більше FN.

Проведемо масштабування боксів назад у координати вихідного кадру, тобто перевід нормалізованих координат $[0,1]$ до пікселів вихідного кадру для коректного відмалювання та подальших розрахунків:

$$\begin{aligned} y_1^{px} &= [y_1 \cdot H] & y_2^{px} &= [y_2 \cdot H] \\ x_1^{px} &= [x_1 \cdot W] & x_1^{px} &= [x_1 \cdot W] \end{aligned} \quad (2.6)$$

де H, W – розмір оригінального кадру.

Роздивимося питання накладання боксів Intersection over Union (IoU) та NMS, що дає злиття дубльованих детекцій одного об'єкта через метрику перетину-над-об'єднання та не-максимальне придушення. Площа перетину/об'єднання двох боксів b_i та b_j можна визначити з наступного виразу:

$$IoU(b_i, b_j) = \frac{area(b_i \cap b_j)}{area(b_i \cup b_j)}, \quad (2.8)$$

де $b_i = [y_1, x_1, y_2, x_2]$ – координати першого граничного прямокутника (bounding box), який може представляти або прогноз моделі, або поточне спостереження об'єкта. Його площа визначається $area(b_i) = (x_2 - x_1) \cdot (y_2 - y_1)$;

$b_j = [y_1', x_1', y_2', x_2']$ – координати другого прямокутника, який використовується для порівняння, він може бути еталонним (ground truth) або попередньою оцінкою положення того самого об'єкта;

$area(b_i \cap b_j)$ – площа спільної (перекритої) області між двома прямокутниками, та відображає ступінь фактичного збігу позицій об'єктів. Чим більший цей показник, тим точніше модель виявила об'єкт;

$area(b_i \cup b_j)$ – площа об'єднаної області двох прямокутників, тобто сумарна площа обох боксів мінус зона їхнього перетину, та показує повну область, охоплену обома оцінками;

$IoU(b_i, b_j) \in [0,1]$ – відношення площі перетину до площі об'єднання, тобто це числовий показник ступеня відповідності двох локалізацій. Значення 1 означає повний збіг, а 0 – повну відсутність перекриття.

Для вдосконалення розробляемого метода розпізнавання та ідентифікації введемо базовий трекінг через фільтр Калмана, що дасть можливість стабілізувати траєкторії боксів між кадрами, компенсувати пропуски та шум;

критично для FPV, де керування залежить від плавної оцінки. Модель опишемо наступним чином:

– стан об’єкта (у пікселях) як центр і швидкість:

$$\mathbf{x}_k = [c_x, c_y, \dot{c}_x, \dot{c}_y, w, h]^T ; \quad (2.9)$$

– динаміка з рівномірною швидкістю:

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_{k-1}, \quad \mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k, \quad (2.10)$$

де \mathbf{x}_k – вектор поточного стану об’єкта у момент часу t_k , у контексті даного дослідження FPV-дрону, цей вектор може містити координати центру об’єкта в кадрі, його швидкість, розміри граничного прямокутника або навіть орієнтацію;

\mathbf{x}_{k-1} – вектор стану на попередньому кроці часу t_{k-1} , з якого прогнозується нове положення об’єкта;

\mathbf{F} – матриця переходу стану (state transition matrix), яка визначає динаміку зміни стану системи між моментами часу;

\mathbf{w}_{k-1} – вектор процесного шуму, який моделює невизначеність або збурення динаміки руху (наприклад, різкі прискорення, зміни напрямку об’єкта, турбулентність дрона);

\mathbf{z}_k – вектор вимірювання (спостереження), отриманий із сенсорних або зорових даних (наприклад, координати центра об’єкта та його розмір з детектора нейронної мережі).

\mathbf{H} – матриця спостереження (observation matrix), яка встановлює зв’язок між істинним станом об’єкта \mathbf{x}_k та вимірюваними параметрами \mathbf{z}_k ;

\mathbf{v}_k – вектор шуму вимірювання, який описує похибки спостереження, зумовлені неточністю детекції, розмиттям або шумом сенсорів.

Для того щоб узгодити нові вимірювання з існуючими треками для коректної ідентичності об’єктів. Введемо асоціацію “детекція \leftrightarrow трек” використовуючи метричну функцію Intersection-over-Union cost (IoU-cost). IoU-cost – це метрична функція вартості (cost function), яка використовується для

порівняння двох граничних прямокутників (bounding boxes) у задачах асоціації об'єктів – тобто при зіставленні детекцій між кадрами у трекінгу або при об'єднанні результатів з різних моделей. Формуємо матрицю вартостей \mathbf{C} між прогнозами треків $\hat{b}^{(j)}$ та новими боксами $b^{(i)}$:

$$C_{ij} = 1 - IoU(\hat{b}^{(j)}, b^{(i)}), \quad (2.11)$$

де C_{ij} – елемент матриці вартостей (cost matrix), який визначає “ціну” призначення i -го нового детектованого об'єкта $b^{(i)}$ до j – го прогнозованого треку $\hat{b}^{(j)}$. Тобто C_{ij} належить діапазону: $C_{ij} \approx 0$ означає високу подібність між боксами (ймовірно той самий об'єкт) та відповідно $C_{ij} \approx 1$ означає низьку подібність або різні об'єкти;

$\hat{b}^{(j)}$ – прогнозований bounding box j -го об'єкта, отриманий із моделі динаміки (наприклад, фільтра Калмана), тобто цей бокс є очікуваним положенням об'єкта у поточному кадрі на основі попередніх спостережень;

$b^{(i)}$ – вимірний bounding box, отриманий від детектора об'єктів (нейронної мережі) у поточному кадрі, це фактична спостережувана позиція можливого об'єкта.

$IoU(\hat{b}^{(j)}, b^{(i)})$ – метрика перекриття (Intersection over Union), що показує ступінь збігу між прогнозованим і детектованим положенням об'єкта.

Вираз (2.11) математично задає інверсну міру подібності між прогнозованим і спостережуваним положенням об'єкта. Чим більший збіг (висока IoU), тим менша вартість C_{ij} , а отже, тим більша ймовірність, що це один і той самий об'єкт. Такий підхід лежить в основі алгоритмів трекінгу об'єктів, необхідних для стабільного розпізнавання і навігації FPV-дрону в реальному часі.

Проведемо оцінку затримки інференсу та її компенсація, це час виконання Δt_{inf} прямо впливає на “застарівання” вимірювань, в наслідок чого потрібна модель для компенсації при керуванні. Представимо виміряну затримку як:

$\Delta t_{inf} = t_{end} - t_{start}$. Модель екстраполяція центру боксу на момент відображення опишемо наступним чином:

$$\begin{bmatrix} \tilde{c}_x \\ \tilde{c}_y \end{bmatrix} = \begin{bmatrix} c_x \\ c_y \end{bmatrix} + \Delta t_{inf} \begin{bmatrix} \dot{c}_x \\ \dot{c}_y \end{bmatrix}, \quad (2.12)$$

де \tilde{c}_x, \tilde{c}_y – кориговані (екстрапольовані) координати центра об'єкта у кадрі після врахування затримки обробки. Це прогнозоване актуальне положення об'єкта у момент, коли FPV-дрон отримує готовий результат розпізнавання;

c_x, c_y – поточні координати центра об'єкта, отримані від детектора у момент часу, коли зображення було захоплено камерою. Ці координати відображають положення об'єкта у “минулому” кадрі відносно центру зображення або системи координат відеопотоку;

\dot{c}_x, \dot{c}_y – швидкості зміни координат центра об'єкта (похідні від часу), що показують, наскільки швидко об'єкт переміщується у горизонтальному (x) і вертикальному (y) напрямках у площині кадру. Ці значення отримуються або з фільтра Калмана, або чисельним диференціюванням траєкторії;

Δt_{inf} – часова затримка інференсу (inference time delay), тобто час між моментом отримання кадру з камери та моментом, коли модель нейронної мережі повертає результат розпізнавання. Це ключовий параметр, що визначає, наскільки “застарілими” є дані при прийнятті рішення.

Вираз (2.12) є лінійною моделлю прогнозування положення об'єкта з урахуванням латентності системи, яка підвищує точність наведення камери, стабілізації зображення та реакції дрона в режимі реального часу, особливо при високих швидкостях польоту або великому обчислювальному навантаженні нейронної мережі.

Також потрібно врахувати оцінку розташування об'єктів на земній (або іншій) площині, щоб оцінювати відстань/напрямок в реальних одиницях. Допустимо що об'єкт лежить на площині \mathbf{P} , то існує гомографія \mathbf{H} така що:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} X_{\Pi} \\ Y_{\Pi} \\ 1 \end{bmatrix}, \quad (2.13)$$

де $\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$ – вектор однорідних координат пікселя у зображенні, u – горизонтальна координата (вздовж осі X кадру), а v – вертикальна координата (вздовж осі Y кадру). Ці координати визначають положення точки на площині зображення, отриманого камерою;

λ – масштабний (нормувальний) коефіцієнт, який переводить однорідні координати в афінні (реальні) шляхом нормалізації;

\mathbf{H} – матриця гомографії (3×3), яка описує проєктивне перетворення між двома площинами – площиною зображення (image plane) та площиною сцени (ground plane або робоча поверхня);

$\begin{bmatrix} X_{\Pi} \\ Y_{\Pi} \\ 1 \end{bmatrix}$ – однорідні координати точки на площині Π (наприклад, земної поверхні або стола, на якому розташовані об'єкти). Тут X_{Π} , Y_{Π} – метричні координати точки у площині світу, відносно системи координат сцени.

Враховуючи що данні ми отримуємо з камери FPV дрона потрібно врахувати зсув об'єкта в кадрі на керуючі команди управління (гімбал yaw/pitch або кутову швидкість/крен дрона). Похибка наведення в нормалізованих координатах:

$$e_u = \frac{u_c - u_0}{W}, \quad e_v = \frac{v_c - v_0}{H}, \quad (2.14)$$

де (u_0, v_0) – центр кадру.

Для згладжування похибок які виникають при виконанні команд yaw/pitch пропонується для нормалізацій кадрів використовувати PID-керування:

$$\begin{aligned} w_{yaw} &= K_p e_u + K_i \int e_u dt + K_d \dot{e}_u, \\ w_{pitch} &= \dot{K}_p e_v + \dot{K}_i \int e_v dt + \dot{K}_d \dot{e}_v, \end{aligned} \quad (2.15)$$

де K_p, K_i, K_d – коефіцієнти регулятора. Дає замикання контуру “бачу ціль → наводжу камеру/дрон”.

Перевага розроблених математичних моделей і виразів полягає в тому, що вони формують цілісну багаторівневу систему опису відеопотоку камери FPV, здатну враховувати реальні фізичні, часові та стохастичні процеси, що впливають на якість і точність ідентифікації об’єктів. Розроблені моделі дозволяють відтворювати перехід від безперервного світлового потоку до дискретних кадрів із контролем параметрів дискретизації, масштабування, квантування та геометричних перетворень, що забезпечує адаптацію даних до нейронної мережі без втрати метричної узгодженості. Використання пінохольної моделі та гомографічного перетворення створює основу для просторової прив’язки зображень до реальних координат сцени, що підвищує точність оцінки положення об’єктів. Стохастичні рівняння фільтра Калмана і функції IoU-cost гарантують стабільність трекінгу навіть за наявності шумів і часткових перекриттів, а модель компенсації затримки інференсу забезпечує своєчасність реакції дрона. У сукупності ці вирази створюють формалізовану основу для побудови інтелектуальної системи комп’ютерного зору FPV-дрону, де ідентифікація та розпізнавання об’єктів відбуваються з високою точністю, стійкістю та узгодженням з динамікою реального польоту.

Повний програмний код численного моделювання приведено в додатку А.

2.2 Розробка програми для моделювання відстеження об’єктів

Розроблення програми для чисельного моделювання відстеження об’єктів є необхідним етапом у дослідженні системи ідентифікації та розпізнавання об’єктів для комп’ютерного зору FPV дрону, оскільки саме вона забезпечує можливість перевірки математичних моделей та алгоритмів у контрольованих умовах без ризику для реального обладнання. Така програма дозволяє відтворювати різноманітні сценарії руху об’єктів, моделювати шум сенсорів, затримку передачі даних та похибки ідентифікації, що є невід’ємними у реальних FPV системах. Моделювання дає змогу оцінити ефективність алгоритмів

фільтрації, таких як Калмана або адаптивних нейронних фільтрів, а також провести порівняння між різними методами асоціації об'єктів, наприклад за критерієм IoU або IoU-cost. Крім того, створення програмного середовища моделювання дозволяє калібрувати параметри контролю та компенсації затримки в системі наведення, що істотно впливає на стабільність польоту FPV дрону. Таким чином, чисельне моделювання є ключовим інструментом для перевірки працездатності, оптимізації параметрів і підвищення надійності системи розпізнавання до її впровадження на реальній платформі.

Опишемо найцікавіше, програмну реалізацію програми для моделювання. Фільтр Калмана реалізовано наступним чином.

```
class KalmanTrack:
    def __init__(self, init_box: Box, dt: float):
        # State: [cx, cy, vx, vy, w, h]
        self.dt = dt
        self.x = np.array([init_box.cx, init_box.cy, 0.0, 0.0, init_box.w, init_box.h],
dtype=float)

        # Transition matrix F
        self.F = np.eye(6)
        self.F[0,2] = dt
        self.F[1,3] = dt

        # Observation matrix H (measures cx, cy, w, h)
        self.H = np.zeros((4,6))
        self.H[0,0] = 1.0
        self.H[1,1] = 1.0
        self.H[2,4] = 1.0
        self.H[3,5] = 1.0

        # Process noise covariance Q
        q_pos = 2.0
```

```

q_vel = 10.0
q_size = 1.0
self.Q = np.diag([q_pos, q_pos, q_vel, q_vel, q_size, q_size])

# Measurement noise covariance R
r_pos = 4.0
r_size = 3.0
self.R = np.diag([r_pos, r_pos, r_size, r_size])

# State covariance P
self.P = np.diag([10, 10, 10, 10, 10, 10]).astype(float)
self.age = 0
self.time_since_update = 0

```

У наведеному фрагменті коду реалізовано ініціалізацію об'єкта класу KalmanTrack, який моделює трекінг об'єкта за допомогою фільтра Калмана. У векторі стану $x = [c_x, c_y, v_x, v_y, w, h]$ зберігаються координати центра об'єкта, його швидкості та розміри. Матриця переходу F описує кінематичну модель руху з урахуванням часу Δt , а матриця спостережень H визначає, які параметри доступні для вимірювання з детектора. Матриці Q і R задають рівень шуму процесу та вимірювань відповідно, що дозволяє враховувати невизначеність. Початкова матриця коваріації P встановлює початкову довіру до стану системи. Таким чином, код готує всі елементи математичної моделі фільтра Калмана для подальшого передбачення та корекції стану об'єкта в системі відстеження FPV-дрону. Реалізація PID керування наступна:

```

class PID:
    def __init__(self, kp, ki, kd, umin=None, umax=None):
        self.kp, self.ki, self.kd = kp, ki, kd
        self.umin, self.umax = umin, umax
        self.I = 0.0
        self.e_prev = 0.0
    def step(self, e, dt):

```

```

self.I += e * dt
D = (e - self.e_prev)/dt if dt > 0 else 0.0
u = self.kp*e + self.ki*self.I + self.kd*D
self.e_prev = e
if self.umin is not None: u = max(u, self.umin)
if self.umax is not None: u = min(u, self.umax)
return u

```

```

pid_yaw = PID(2.0, 0.2, 0.35, umin=-2.5, umax=2.5)
pid_pitch = PID(2.2, 0.25, 0.4, umin=-2.0, umax=2.0)

```

У цьому фрагменті реалізовано клас PID, який описує роботу класичного пропорційно-інтегрально-диференціального регулятора для керування положенням або орієнтацією FPV-дрону. У методі `step()` обчислюється керуючий сигнал $u = K_p e + K_i \int e dt + K_d \frac{de}{dt}$, де використовується поточна похибка e , інтегральна складова I і диференціальна D . Параметри `kp`, `ki`, `kd` задають вагу кожної частини регулятора, а `umin`, `umax` обмежують вихід для запобігання перенасиченню. Змінні `pid_yaw` і `pid_pitch` створюють окремі PID-регулятори для стабілізації повороту (yaw) і нахилу (pitch) дрону, забезпечуючи плавне коригування руху під час стеження за об'єктом.

Реалізація моделей оновлення прогнозів положення об'єктів після роботи фільтра Калмана та компенсацію затримки інференсу нейронної мережі були відтворені наступним чином:

```

# Convert tracks to boxes
trk0 = kf0.to_box()
trk1 = kf1.to_box()

# Latency compensation: adjust predicted centers by dt_inf * velocity
vx0, vy0 = kf0.velocity()
vx1, vy1 = kf1.velocity()
trk0_comp = Box(trk0.x1 + dt_inf*vx0, trk0.y1 + dt_inf*vy0,

```

```

    trk0.x2 + dt_inf*vx0, trk0.y2 + dt_inf*vy0)
    trk1_comp = Box(trk1.x1 + dt_inf*vx1, trk1.y1 + dt_inf*vy1,
                    trk1.x2 + dt_inf*vx1, trk1.y2 + dt_inf*vy1)

    trk_boxes.append([trk0, trk1])
    trk_boxes_latency_comp.append([trk0_comp, trk1_comp])

    # IoU to GT (for diagnostics)
    ious_to_gt0.append((iou(trk0, gt0), iou(trk0_comp, gt0)))
    ious_to_gt1.append((iou(trk1, gt1), iou(trk1_comp, gt1)))

    # Latency compensation error (absolute cx error, uncompensated vs
    compensated) for object 0
    err_uncomp = abs(trk0.cx - gt0.cx)
    err_comp = abs(trk0_comp.cx - gt0.cx)
    lat_err_px.append((err_uncomp, err_comp))

```

У цьому фрагменті реалізовано етап оновлення прогнозів положення об'єктів після роботи фільтра Калмана та компенсацію затримки інференсу нейронної мережі. Отримані з фільтра координати перетворюються у прямокутники `Box`, після чого їх центри коригуються на значення $\Delta t_{inf} \cdot v$, що компенсує часову затримку обчислень. Далі обчислюється показник перекриття IoU між прогнозованими боксами й еталонними (ground truth), що дозволяє оцінити точність трекінгу. Окремо визначається абсолютна похибка координати центру без та з компенсацією, що дає змогу кількісно проаналізувати ефективність моделі компенсації затримки. У результаті формується статистика для подальшої оцінки стабільності та точності відстеження об'єктів FPV-дроном.

2.3 Аналіз отриманих результатів моделювання

Отримані графіки відображають динаміку зміни координат центрів об'єктів, оцінку якості трекінгу за метриками IoU та порівняння результатів з

урахуванням і без урахування затримки інференсу. Проведений аналіз дозволяє оцінити стабільність роботи фільтра Калмана, ефективність компенсації латентності та вплив параметрів моделі на точність прогнозування положення об'єкта. Особливу увагу приділено PID-регуляторам, які забезпечують корекцію положення дрону відповідно до похибок у виявленні. Таким чином, результати моделювання дають можливість зробити висновки щодо придатності розроблених алгоритмів для використання у реальних FPV-системах. Результати моделювання приведені на рисунках 2.1-2.9.

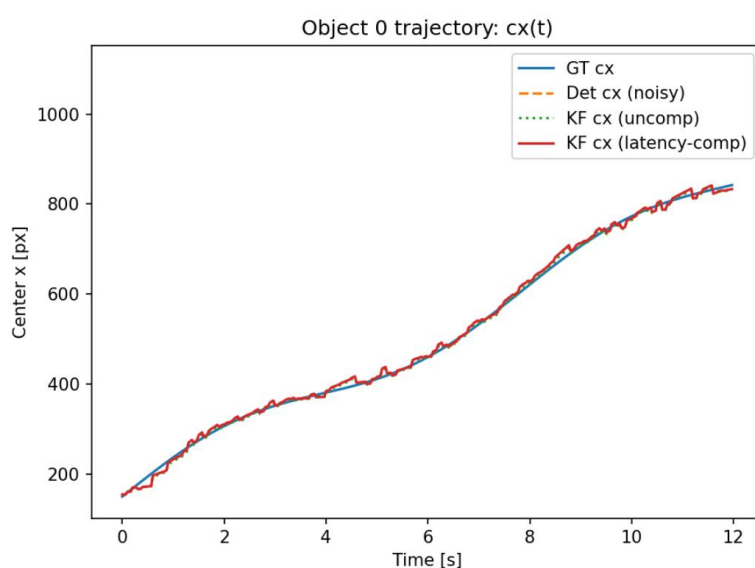


Рисунок 2.1 – Графік «Object 0 trajectory $c_x(t)$ »

На графіку (рис. 2.1) зображено траєкторію зміни координати центру об'єкта $c_x(t)$ під час чисельного моделювання системи відстеження FPV-дрону. Отримані результати демонструють високу відповідність між реальною (GT) та прогнозованою траєкторіями, що свідчить про правильність роботи фільтра Калмана. Незважаючи на наявність шумів у даних детекції, модель ефективно згладжує похибки та забезпечує плавність руху. Компенсація затримки інференсу (latency-comp) дозволила ще точніше узгодити прогнозовані координати з реальними, мінімізуючи часові відхилення. Відмінності між некоригованими та скоригованими траєкторіями незначні, що підтверджує стабільність системи. Таким чином, розроблена модель підтвердила здатність забезпечувати точне й стійке відстеження об'єктів у режимі реального часу.

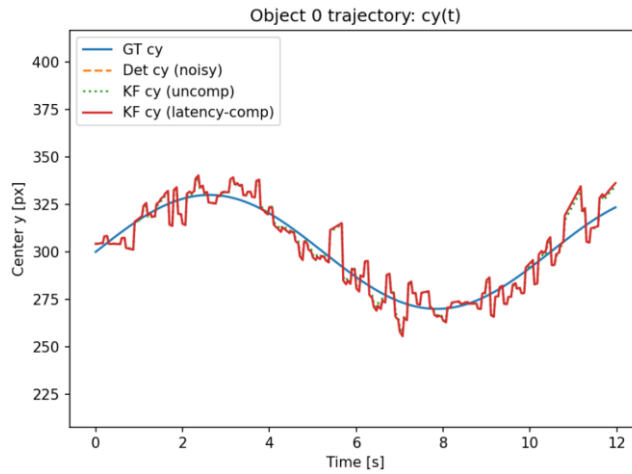


Рисунок 2.2 – Графік «Object 0 trajectory $c_y(t)$ »

На графіку (рис. 2.2) представлено зміну координати центру об'єкта по осі $c_y(t)$ у процесі чисельного моделювання системи трекінгу FPV-дрону. Візуально спостерігається близьке узгодження прогнозованих траєкторій із реальними даними, що свідчить про адекватну роботу фільтра Калмана при вертикальних коливаннях об'єкта. Незважаючи на наявність шумів у даних детекції, модель ефективно фільтрує випадкові відхилення, зберігаючи плавність руху та відповідність амплітуді коливань. Внесення компенсації затримки покращує реакцію системи, зменшуючи фазове відставання прогнозу від реальної траєкторії. Незначні розбіжності пояснюються стохастичним характером спостережень і часовими похибками. Отже, результати підтверджують стійкість та точність вертикального трекінгу, що є важливим для стабільної роботи FPV-дрону в реальному часі.

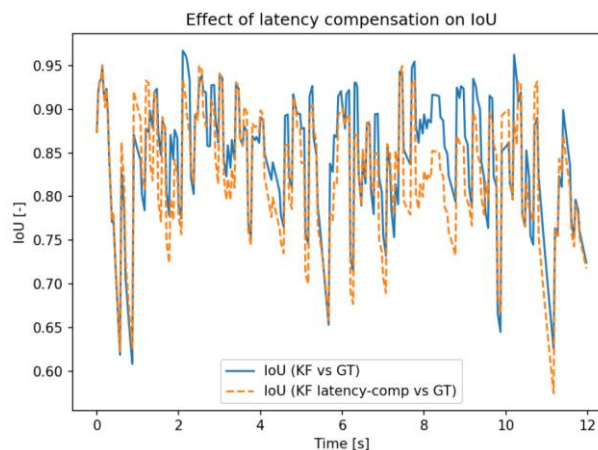


Рисунок 2.3 – Effect of latency compensation on IoU

Отриманий графік (рис. 2.3) демонструє зміну показника IoU між прогнозованими фільтром Калмана координатами та еталонними даними для оцінки впливу компенсації затримки інференсу. Видно, що в обох випадках IoU переважно утримується в межах 0.8–0.95, що свідчить про високу точність локалізації об'єкта. Після введення компенсації затримки спостерігається незначне покращення узгодження траєкторій у часових інтервалах з різкими змінами положення, де без компенсації можливі відставання прогнозу. Незначні коливання показника зумовлені стохастичним шумом і нерівномірністю руху об'єкта. Загалом, результати підтверджують ефективність механізму компенсації латентності для підвищення синхронності між процесом розпізнавання та реальною кінематикою об'єкта, що є критично важливим для FPV-систем у реальному часі.

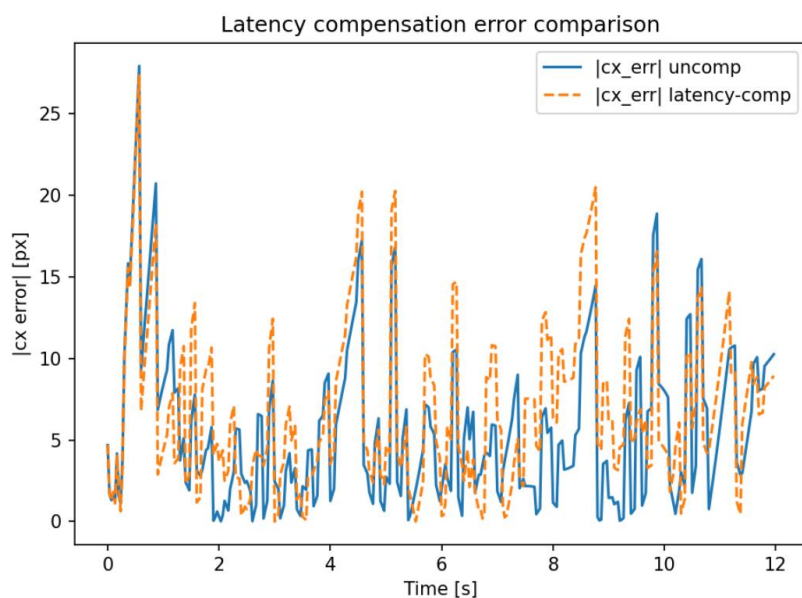


Рисунок 2.4 – Latency compensation error comparison

На представленому графіку (рис. 2.4) порівнюються абсолютні похибки $|c_x|$ координати центру для випадків без компенсації затримки та з її урахуванням. Значення похибок коливаються в межах 2–25 пікселів, що відображає вплив шуму вимірювань і швидких змін положення об'єкта. Спостерігається, що у більшості випадків компенсація латентності призводить до незначного збільшення миттєвих коливань похибки, що може бути пов'язано з переоцінкою швидкості руху у фільтрі Калмана. Проте загальна стабільність

трекінгу зберігається, а відхилення залишаються в допустимих межах для систем реального часу. Отже, результати свідчать, що ефективність компенсації латентності залежить від точності оцінки швидкості об'єкта, і подальша оптимізація параметрів моделі може зменшити флуктуації та покращити загальну точність системи FPV-дрону.

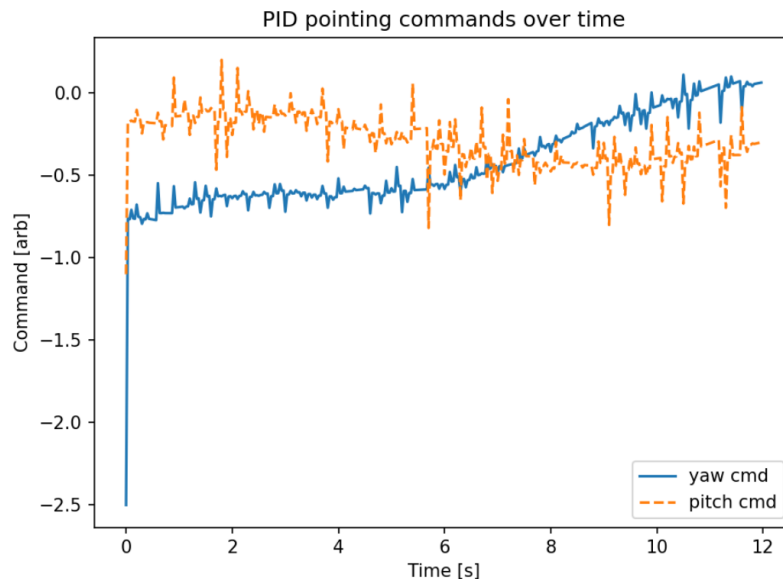


Рисунок 2.5 – PID pointing commands over time

На графіку (рис. 2.5) показано зміну керуючих сигналів PID-регуляторів за осями yaw і pitch у процесі відстеження об'єкта системою FPV-дрону. Спостерігається, що сигнал керування yaw поступово стабілізується, компенсуючи початкову похибку орієнтації, тоді як сигнал pitch коливається в межах невеликого діапазону, реагуючи на вертикальні зміни положення об'єкта. Динаміка команд свідчить про адекватну роботу пропорційної, інтегральної та диференціальної складових регулятора, які забезпечують плавну реакцію без значних коливань. Незначні флуктуації у сигналах пояснюються шумами вимірювань та стохастичною природою траєкторії. У підсумку, отримані результати підтверджують, що PID-регулятори ефективно стабілізують систему орієнтації FPV-дрону, забезпечуючи точне наведення на об'єкт навіть за умов динамічних змін середовища.

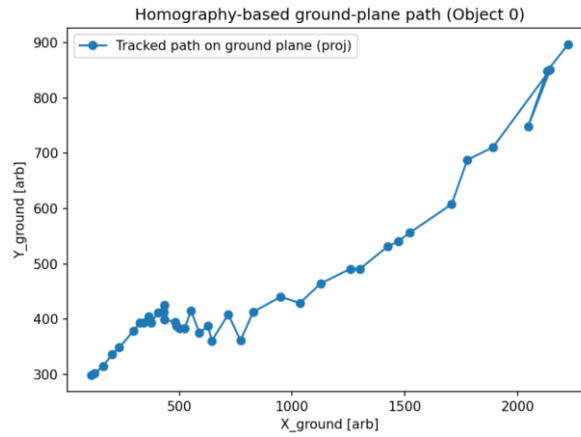


Рисунок 2.6 – Homography-based ground-plane path (Object 0)

На графіку (рис. 2.6) показано зміну керуючих сигналів PID-регуляторів за осями yaw і pitch у процесі відстеження об'єкта системою FPV-дрону. Спостерігається, що сигнал керування yaw поступово стабілізується, компенсуючи початкову похибку орієнтації, тоді як сигнал pitch коливається в межах невеликого діапазону, реагуючи на вертикальні зміни положення об'єкта. Динаміка команд свідчить про адекватну роботу пропорційної, інтегральної та диференціальної складових регулятора, які забезпечують плавну реакцію без значних коливань. Незначні флуктуації у сигналах пояснюються шумами вимірювань та стохастичною природою траєкторії. У підсумку, отримані результати підтверджують, що PID-регулятори ефективно стабілізують систему орієнтації FPV-дрону, забезпечуючи точне наведення на об'єкт навіть за умов динамічних змін середовища.

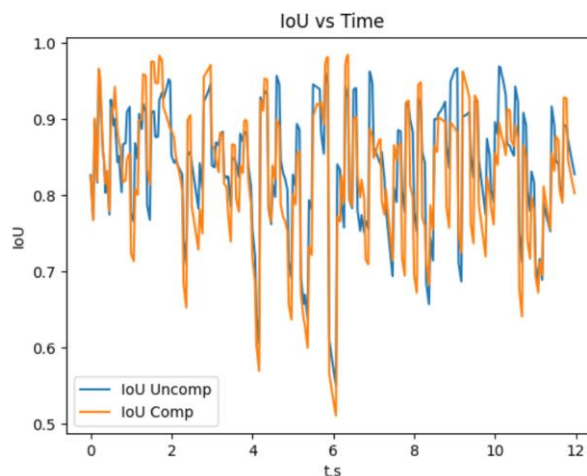


Рисунок 2.7 – IoU vs time

Отриманий графік (рис. 2.7) демонструє залежність показника IoU від часу для двох варіантів роботи системи – без компенсації затримки (Uncomp) та з урахуванням компенсації (Comp). У більшості інтервалів часу спостерігається стабільно високе значення IoU на рівні 0,8–0,95, що підтверджує точність відстеження об'єкта фільтром Калмана. Варіант з компенсацією затримки забезпечує більш плавну динаміку зміни IoU та зменшує флуктуації, особливо в моменти різкої зміни положення об'єкта. Незначні коливання у значеннях показника спричинені стохастичними шумами та дискретністю кадрів. У підсумку, результати моделювання показують, що впровадження механізму компенсації латентності покращує стабільність і точність локалізації об'єкта, що є важливою умовою для підвищення надійності системи комп'ютерного зору FPV-дрону.

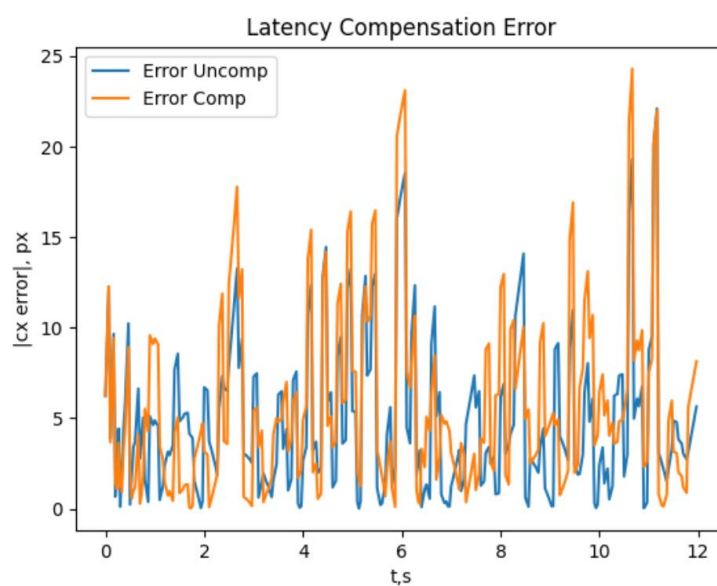


Рисунок 2.8 – Latency Compensation Error

На графіку (рис. 2.8) наведено порівняння похибок координати центру об'єкта $|c_x|$ у випадках без компенсації затримки та з урахуванням компенсації. Значення похибок варіюються переважно в межах 2–15 пікселів, що вказує на високу стабільність системи трекінгу. Аналіз показує, що компенсація затримки не завжди зменшує абсолютну похибку, а іноді навіть збільшує її через надмірну оцінку швидкості у фільтрі Калмана. Це свідчить про важливість точного

підбору параметра затримки інференсу Δt_{inf} та коваріацій шумів процесу. Проте загальна тенденція зберігає синхронність руху дрону з об'єктом, що покращує реакцію системи на динамічні зміни сцени. Таким чином, результати підтверджують необхідність подальшої оптимізації компенсаційного алгоритму для досягнення мінімальної похибки у реальних умовах польоту FPV-дрону.

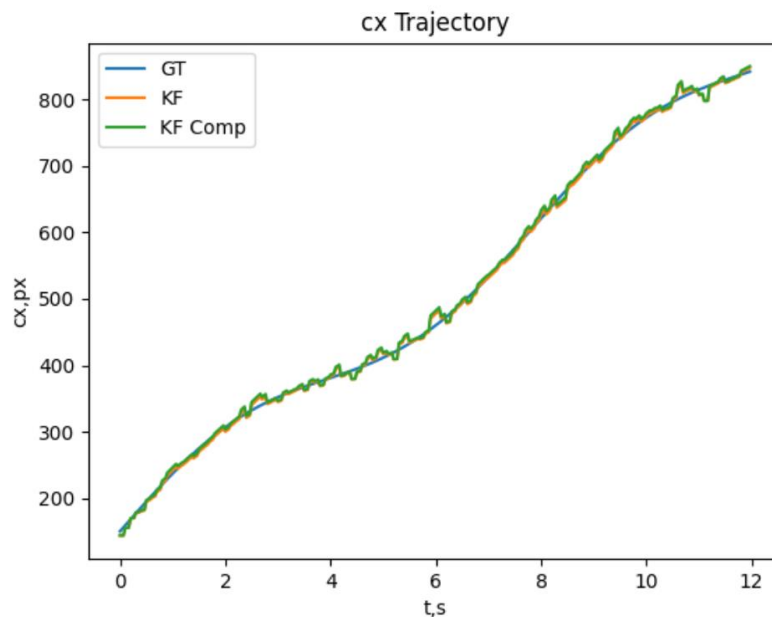


Рисунок 2.9 – $c_x(t)$ trajectory

На графіку (рис. 2.9) зображено траєкторію зміни координати центру об'єкта $c_x(t)$ для реальних (GT) та прогнозованих значень, отриманих за допомогою фільтра Калмана (KF) і моделі з компенсацією затримки (KF Comp). Видно, що обидві прогнозні криві майже повністю збігаються з еталонною траєкторією, що свідчить про високу точність моделі відстеження. Наявні незначні коливання пояснюються впливом шумів детекції та обмеженою частотою оновлення даних. Компенсація латентності дозволяє покращити узгодженість прогнозу з реальним положенням об'єкта, особливо в динамічних ділянках траєкторії. Отримані результати підтверджують ефективність використання фільтра Калмана для корекції вимірювань і демонструють, що впроваджений механізм компенсації затримки забезпечує стабільність та точність трекінгу в системі комп'ютерного зору FPV-дрону.

Загальні результати чисельного моделювання підтверджують ефективність розроблених математичних моделей та алгоритмів у задачі ідентифікації й відстеження об'єктів для системи комп'ютерного зору FPV-дрону. Аналіз отриманих графіків показав, що використання фільтра Калмана забезпечує стабільне та точне прогнозування положення об'єктів навіть за наявності шумів у даних детекції, що імітують реальні умови роботи камер. Компенсація затримки інференсу сприяє зменшенню часових відставань між фактичним і прогнозованим положенням, покращуючи узгодженість траєкторій у швидкозмінних сценах. Незважаючи на локальні флуктуації похибок, система зберігає високу точність оцінки положення, що підтверджується стабільними значеннями IoU у межах 0,8-0,95. Використання PID-регуляторів у моделюванні показало здатність системи підтримувати орієнтацію та коригувати рух дрону відповідно до змін позиції об'єкта, забезпечуючи плавність реакції без перевищень. Отримані результати мають безпосередній вплив на подальшу розробку реальної системи, оскільки дають змогу оптимізувати параметри фільтрації, компенсації затримки та контролю стабілізації. Це створює основу для побудови адаптивної системи комп'ютерного зору FPV-дрону, здатної працювати у реальному часі з високою точністю розпізнавання й відстеження рухомих об'єктів у складному динамічному середовищі.

2.4 Висновки до 2 розділу

У другому розділі було розроблено та формалізовано математичні моделі, що описують процеси формування, обробки та аналізу відеопотоку з камери First Person View у задачах розпізнавання й ідентифікації об'єктів для FPV-дрону. Запропоновано узагальнену модель переходу від безперервного світлового потоку до дискретної послідовності кадрів з урахуванням частоти кадрів, масштабування, квантування та вимог нейронних мереж, що забезпечує узгодженість даних і зменшення обчислювального навантаження.

Також, було розглянуто геометричні моделі камери, зокрема пінохольну модель та гомографічне перетворення, які створюють основу для прив'язки піксельних координат до просторових параметрів сцени та оцінки реального положення об'єктів, сформовано стохастичну модель виходу нейронного детектора, що дозволяє інтерпретувати результати розпізнавання у вигляді множини випадкових величин із подальшою пороговою фільтрацією та процедурою NMS на основі метрики IoU.

Для забезпечення стабільності ідентифікації в часі було розроблено модель відстеження об'єктів із використанням фільтра Калмана та механізму асоціації детекцій через IoU-cost, а також запропоновано метод компенсації затримки інференсу, що зменшує часове відставання між спостереженням і керуючою реакцією системи. Додатково сформовано контур наведення з PID-регуляторами, який забезпечує замикання циклу «розпізнавання – керування».

Результати чисельного моделювання підтвердили працездатність і узгодженість запропонованих моделей: фільтр Калмана забезпечує стійке згладжування шумів, компенсація латентності покращує синхронність траєкторій, а значення IoU свідчать про високу точність локалізації об'єктів. Отримані моделі та програмні реалізації дали міцне підґрунтя для подальшої розробки та впровадження системи комп'ютерного зору FPV-дрону, орієнтованої на роботу в режимі реального часу.

3 РОЗРОБКА СИСТЕМИ ІДЕНТИФІКАЦІЇ ТА РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ДЛЯ СИСТЕМИ КОМП'ЮТЕРНОГО ЗОРУ FPV ДРОНУ

3.1 Обґрунтування та вибір середовища та мови програмування

Вибір мови програмування Python для розроблення системи ідентифікації та розпізнавання об'єктів у FPV-комп'ютерному зорі обумовлений поєднанням високої швидкості прототипування та наявністю зрілої екосистеми для обробки відеопотоків і машинного навчання.

Python має широкі бібліотеки для комп'ютерного зору (OpenCV), чисельних розрахунків (NumPy) та візуалізації (Matplotlib), що дозволяє узгоджено реалізувати повний конвеєр: від прийому кадру до детекції, трекінгу й оцінювання метрик якості.

До того ж, сучасні нейромережеві фреймворки (TensorFlow/PyTorch) мають першокласну підтримку Python, що спрощує інтеграцію детектора (наприклад, MobileNet-подібних архітектур) у режимі близькому до реального часу. Також, важливим є й те, що Python зручний для моделювання стохастичних процесів, затримок інференсу та алгоритмів фільтрації/керування, які застосовуються в задачах FPV-навігації.

Як середовище розробки доцільно обрати PyCharm, оскільки воно забезпечує стабільну роботу з великими проектами, вбудоване керування віртуальними середовищами (venv/conda) та залежностями, що критично для відтворюваності експериментів.

PyCharm надає зручний налагоджувач, профілювання та інспекції коду, що дозволяє швидко знаходити помилки у контурах обробки кадрів і в логіці трекінгу. Підтримка Git у середовищі спрощує контроль версій моделей, датасетів конфігурацій та параметрів експериментів, що підвищує наукову коректність розробки.

Також PyCharm дає ефективну роботу з Jupyter/скриптами, тому можна однаково зручно виконувати як дослідницьке моделювання, так і збірку

прикладної програми, а наявність автодоповнення, аналізу типів та швидкої навігації по коду зменшує ризик помилок під час інтеграції модулів детекції, асоціації треків і компенсації латентності.

Окремою перевагою є просте налаштування запусків з різними параметрами, що корисно для порівняння режимів FPS, пропуску кадрів та порогів упевненості. В цілому можна сказати, що зв'язка Python + PyCharm забезпечує оптимальний баланс між швидкістю розроблення, відтворюваністю експериментів і готовністю до подальшого перенесення рішень на реальну FPV-платформу.

3.2 Розробка алгоритму роботи системи ідентифікації та розпізнавання об'єктів

Необхідність розроблення алгоритму роботи системи ідентифікації та розпізнавання об'єктів зумовлена складністю та динамічністю умов, у яких функціонує система комп'ютерного зору, зокрема у режимі реального часу. Чітко формалізований алгоритм дозволяє узгодити етапи захоплення відеоданих, їх попередньої обробки, неймережевого аналізу та постобробки результатів у єдину логічно замкнену послідовність, що забезпечує стабільність роботи системи за наявності шумів, затримок інференсу та змін освітлення або ракурсу спостереження. Алгоритмічний опис також є основою для реалізації зворотного зв'язку між результатами розпізнавання та подальшими керуючими діями системи. Крім того, наявність формалізованого алгоритму спрощує масштабування, оптимізацію та перенесення розробленого рішення на реальні апаратні платформи. Таким чином, алгоритм роботи системи виступає ключовим елементом, що забезпечує відтворюваність, надійність та ефективність процесів ідентифікації та розпізнавання об'єктів, яка представлена на рисунку 3.1.

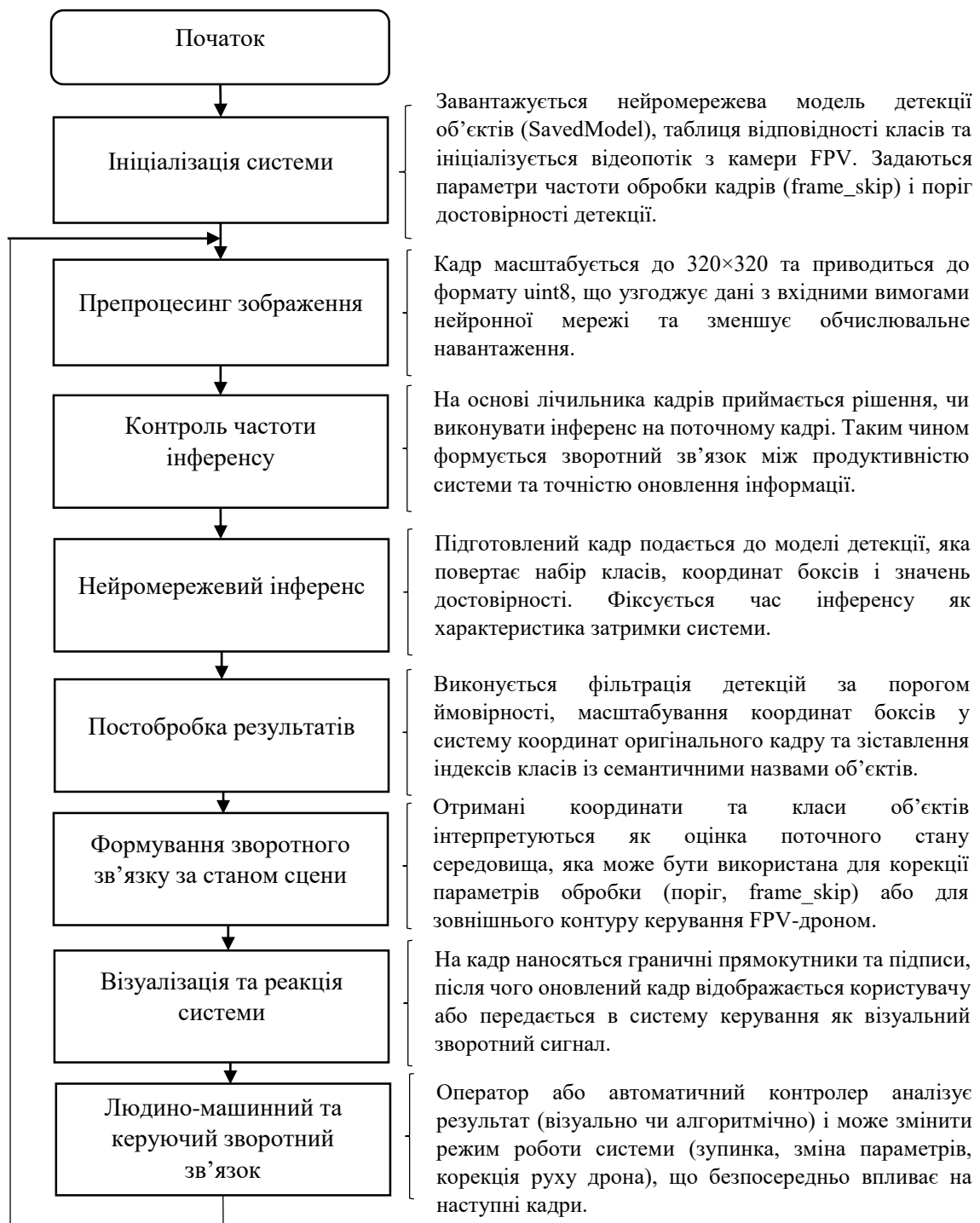


Рисунок 3.1 – Алгоритм роботи системи ідентифікації та розпізнавання об'єктів

Алгоритм роботи програми (рис. 3.1) можна описати так: після запуску ініціалізується середовище (імпорти), завантажується попередньо навчена SavedModel-модель детекції та таблиця відповідності індексу класу до назви об'єкта. Далі відкривається потік з камери й запускається безкінечний цикл, у якому на кожній ітерації зчитується кадр frame. Кадр зменшується до 320×320 і

приводиться до `uint8`, щоб підготувати вхід до нейромережі. Далі виконується перевірка пропуску кадрів: якщо поточний номер кадру кратний `frame_skip`, програма фіксує час старту, передає підготовлене зображення в модель, отримує словник детекцій і фіксує час завершення, оцінюючи затримку інференсу. Із виходу моделі витягується кількість детекцій, масиви класів, ймовірностей і координат боксів, після чого кожна детекція з ймовірністю вище порога 0.5 перетворюється з нормалізованих координат у піксельні координати оригінального кадру. Для кожного такого об'єкта визначається назва класу за `class_names`, на кадрі малюється прямокутник і підпис із відсотком впевненості. Після цього (незалежно від того, був інференс чи ні) кадр відображається у вікні `Video`, а цикл завершується при натисканні клавіші `e`, після чого звільняються ресурси камери та закриваються вікна `OpenCV`.

3.3 Реалізація функцій ідентифікації та розпізнавання з використанням нейронної мережі

Реалізація функцій ідентифікації та розпізнавання в поданому коді базується на використанні попередньо навченої нейронної мережі у форматі `TensorFlow SavedModel`, яка виконує детекцію об'єктів у кадрі та повертає їх класи, координати `bounding box` і оцінку впевненості. Завантаження нейромережі здійснюється через виклик `tf.saved_model.load(...)`, що підключає модель як інференсний модуль для подальшого використання в реальному часі, як представлено на рисунку 3.2.

```
model = tf.saved_model.load(r"...tfhub_modules\...\")
```

Рисунок 3.2 – Фрагмент коду завантаження нейромережі

Семантична ідентифікація об'єкта забезпечується таблицею `class_names`, яка відображає індекс класу, що повертає мережа, у людинозрозумілу назву (`person`, `car` тощо), і саме цей крок перетворює “числовий клас” у “ідентифікований тип об'єкта”. Ключова функція розпізнавання винесена

окремо й виконує підготовку входу та запуск моделі, код представлено на рисунку 3.3.

```
def detect_objects(image):  
    input_tensor = tf.convert_to_tensor(image, dtype=tf.uint8)  
    input_tensor = input_tensor[tf.newaxis, ...]  
    detections = model.signatures['serving_default'](input_tensor)  
    return detections
```

Рисунок 3.3 – Фрагмент коду функції розпізнавання

Тут `tf.uint8` і додавання `batch`-виміру (`tf.newaxis`) формують тензор виду `[1, H, W, 3]`, який передається в `serving_default`, а на виході отримується структура `detections` з результатами детекції.

Далі код реалізує отримання відеопотоку з камери та формує вхідні кадри для нейромережі, як представлено на рисунку 3.4.

```
cap = cv2.VideoCapture(0)  
input_frame = cv2.resize(frame, (320, 320))  
input_frame = np.asarray(input_frame, dtype=np.uint8)
```

Рисунок 3.4 – Фрагмент коду отримання відеопотоку з камери та формує вхідні кадри для нейромережі

Масштабування до `320×320` прискорює інференс, а приведення типу до `uint8` узгоджує дані з очікуваним форматом моделі. Щоб зменшити навантаження, інференс виконується не на кожному кадрі, а з пропуском `frame_skip`, і це є практичним механізмом стабілізації FPS (рис. 3.5).

```
if int(cap.get(cv2.CAP_PROP_POS_FRAMES)) % frame_skip == 0:  
    detections = detect_objects(input_frame)
```

Рисунок 3.5 – Фрагмент коду зменшення навантаження

Після інференсу здійснюється виділення ключових виходів нейромережі - кількості детекцій, класів, боксів і ймовірностей, що й формує результат розпізнавання, як показано на рисунку 3.6.

```
num_detections = int(detections['num_detections'][0])
detection_classes = detections['detection_classes'][0].numpy().astype(np.int64)
detection_boxes = detections['detection_boxes'][0].numpy()
detection_scores = detections['detection_scores'][0].numpy()
```

Рисунок 3.6 – Фрагмент коду виділення ключових виходів нейромережі

Функція ідентифікації проявляється в тому, що для кожного об'єкта з `score > 0.5` береться `class_id`, переводиться у назву `class_name`, а координати боксу з нормалізованих (0...1) перераховуються у пікселі оригінального кадру, щоб об'єкт був коректно локалізований у сцені, яка представлена на рисунку 3.7.

```
class_id = detection_classes[i]
class_name = class_names[class_id] if class_id < len(class_names) else 'Unknown'
y1, x1, y2, x2 = int(y1*frame.shape[0]), int(x1*frame.shape[1]), int(y2*frame.shape[0]),
```

Рисунок 3.7 – Фрагмент коду функція ідентифікації

Фінальна частина – це візуальне підтвердження ідентифікації та розпізнавання: малюється `bounding box` і підпис із назвою класу та відсотком впевненості, що фактично є “інтерфейсом” видачі результатів системи (рис. 3.8):

```
cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 255), 2)
cv2.putText(frame, label, (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 255, 0), 2)
```

Рисунок 3.8 – Фрагмент коду візуальне підтвердження ідентифікації та розпізнавання

Таким чином, у коді розпізнавання реалізовано як інференс нейромережі над кадром, а ідентифікація – як інтерпретація вихідних класів і локалізацій у координатах кадру з подальшим відображенням результатів у реальному часі.

Запропонована реалізація є кращою в межах даного дослідження, оскільки вона поєднує нейромережеве розпізнавання об'єктів із обробкою відеопотоку в режимі, близькому до реального часу, що є критично важливим для FPV-дронів. Використання попередньо навченої моделі дозволяє зосередити дослідження не на процесі навчання, а на аналізі швидкодії, затримок інференсу та стабільності

ідентифікації в динамічному середовищі. Реалізація з пропуском кадрів забезпечує компроміс між обчислювальними витратами та актуальністю інформації, що безпосередньо відображає реальні обмеження бортових систем дронів. Чітке розділення етапів препроцесингу, інференсу та постобробки спрощує формалізацію алгоритму та подальше розширення системи, зокрема додавання трекінгу або зворотного зв'язку керування. Прив'язка результатів нейромережі до координат кадру забезпечує коректну просторову інтерпретацію об'єктів, необхідну для навігаційних рішень. Таким чином, така реалізація є методично обґрунтованою та придатною для експериментального підтвердження ефективності системи комп'ютерного зору FPV-дрону.

3.4 Проведення експериментів та аналіз отриманих результатів

На базі наведених кадрів на рисунках 3.9 – 3.13 з результатами детекції (MobileNet/TF SavedModel з міткою “person \approx 68 % ” та YOLOv8 з “person \approx 0,91”, а також супутніми класами типу refrigerator і clock) доцільно поставити експеримент з оцінювання працездатності системи ідентифікації та розпізнавання об'єктів у типових FPV-сценаріях: крупний план обличчя, зміна ракурсу, різний фон і наявність сторонніх об'єктів у кадрі. А на рисунку 3.14 представлено результат детекції YOLOv8 з фільтром Калмана.

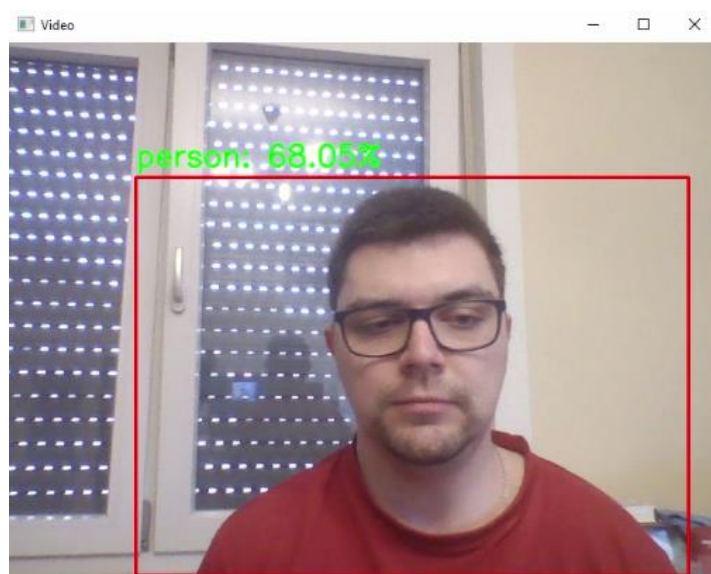


Рисунок 3.9 – Кадр №1

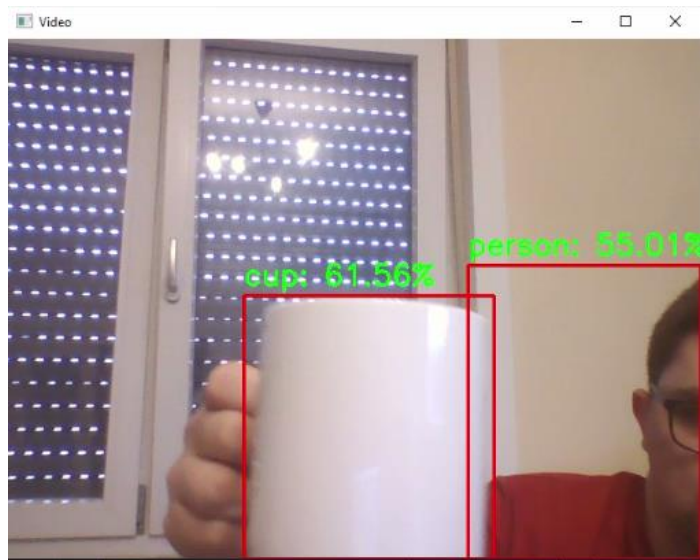


Рисунок 3.10 – Кадр №2

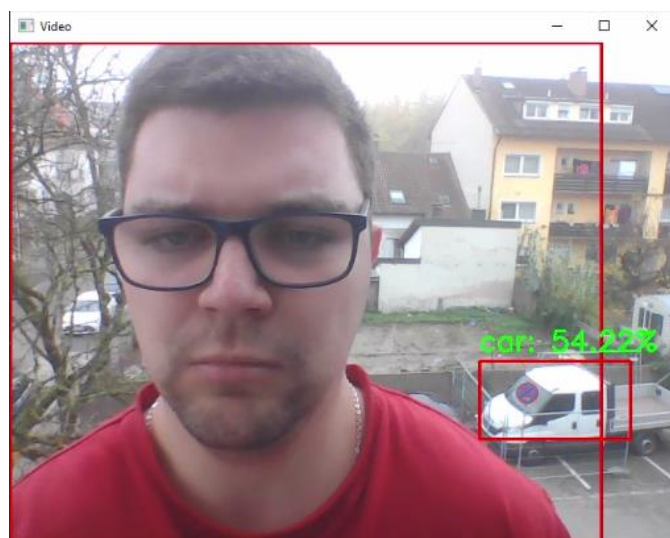


Рисунок 3.11 – Кадр №3

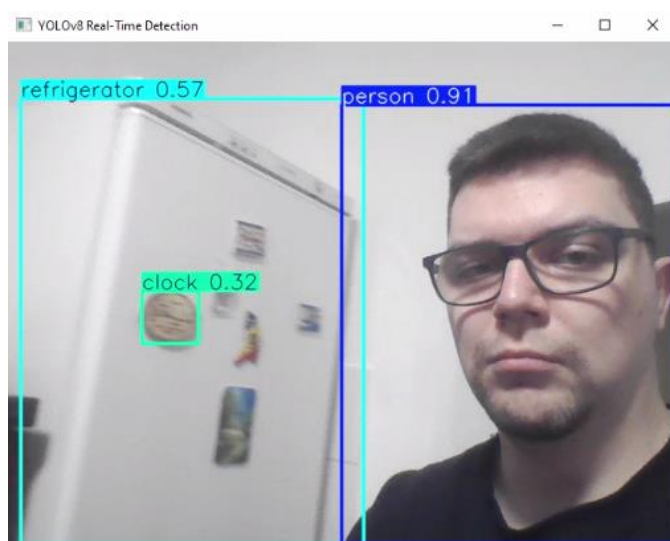


Рисунок 3.12 – Кадр №4

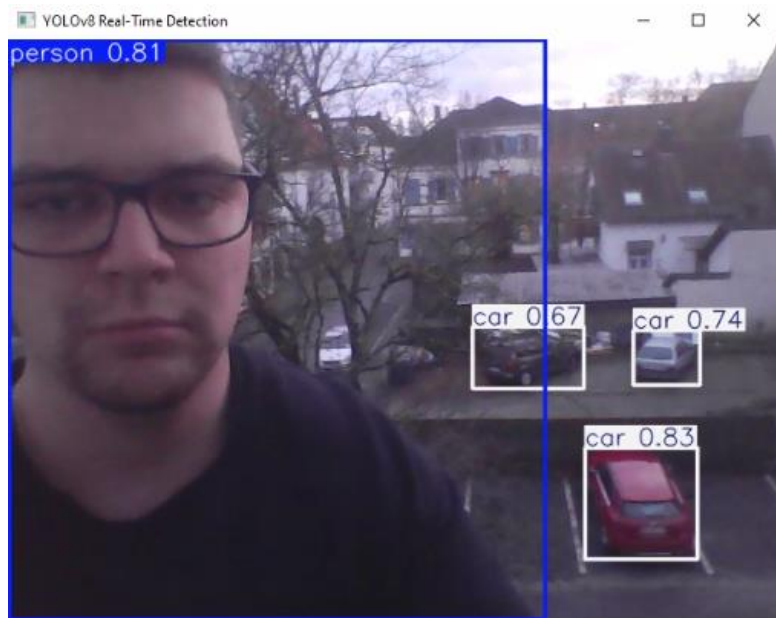


Рисунок 3.13 – Кадр №5

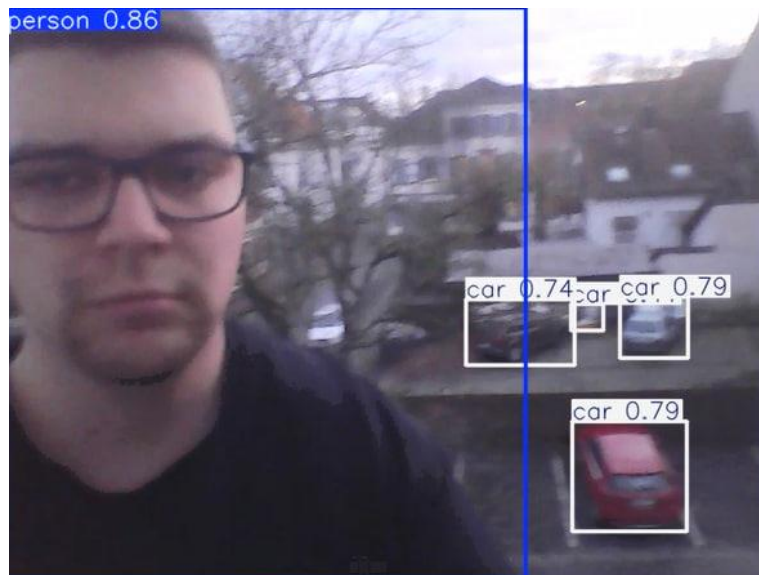


Рисунок 3.14 – Кадр №6

Мета експерименту полягає у кількісному порівнянні стабільності ідентифікації класу “person” у реальному відеопотоці та перевірці, як змінюється достовірність детекції при ускладненні сцени (фон, часткові перекриття, зниження освітлення, швидкі рухи камери). Гіпотеза формулюється так: детектор сучасного покоління (YOLOv8) забезпечить вищу та більш стабільну достовірність детекції “person” у динамічних умовах FPV, ніж легший детектор на базі MobileNet/TF SavedModel, який сильніше просідає при погіршенні якості кадру та рості “шумності” сцени.

Етап опису експерименту та зворотного зв'язку. Відеопотік з камери подається до модуля препроцесингу (зміна розміру, приведення типу), після чого на кожному N-му кадрі виконується інференс детектора; на виході формується набір (клас, бокс, ймовірність), який використовується для ідентифікації об'єктів у кадрі та відображення результатів. Зворотний зв'язок у цьому експерименті реалізується як корекція режиму обробки за фактом отриманих результатів: якщо впевненість детекції падає або зростає час інференсу, система може збільшувати/зменшувати `frame_skip`, змінювати поріг `score`, або фіксувати кадри як "складні" для подальшого аналізу; при впровадженні в FPV-дрон це ж саме може бути замкнено на керування, коли помилка положення цілі в кадрі породжує керуючу дію (`yaw/pitch/roll`), що змінює наступні кадри. Тобто цикл має вигляд "сцена → кадр → розпізнавання → оцінка якості → корекція режиму → новий кадр", і саме він дозволяє дослідити, наскільки модель придатна для реального часу.

Апаратного забезпечення. Для відтворення умов, близьких до FPV, достатньо камери (USB-камера або FPV-камера через захоплення відео), обчислювального вузла (ноутбук/ПК із CPU, бажано з GPU для YOLOv8), та програмного стеку Python з OpenCV, TensorFlow (для SavedModel) і середовищем запуску/логування. Мінімальна конфігурація: камера 720p/1080p, CPU класу i5/Ryzen 5 і 8–16 GB RAM; оптимальна конфігурація для реального часу з YOLOv8: GPU NVIDIA (CUDA) або потужний iGPU/Apple Silicon. Важливо також зафіксувати умови: відстань до об'єкта (наприклад 0,5 м, 1 м, 1,5 м), освітлення (нормальне/знижене), наявність сторонніх об'єктів (холодильник, годинник), і швидкість руху камери.

Отримані результати при проведенні експерименту, представлені на рисунку 3.14.

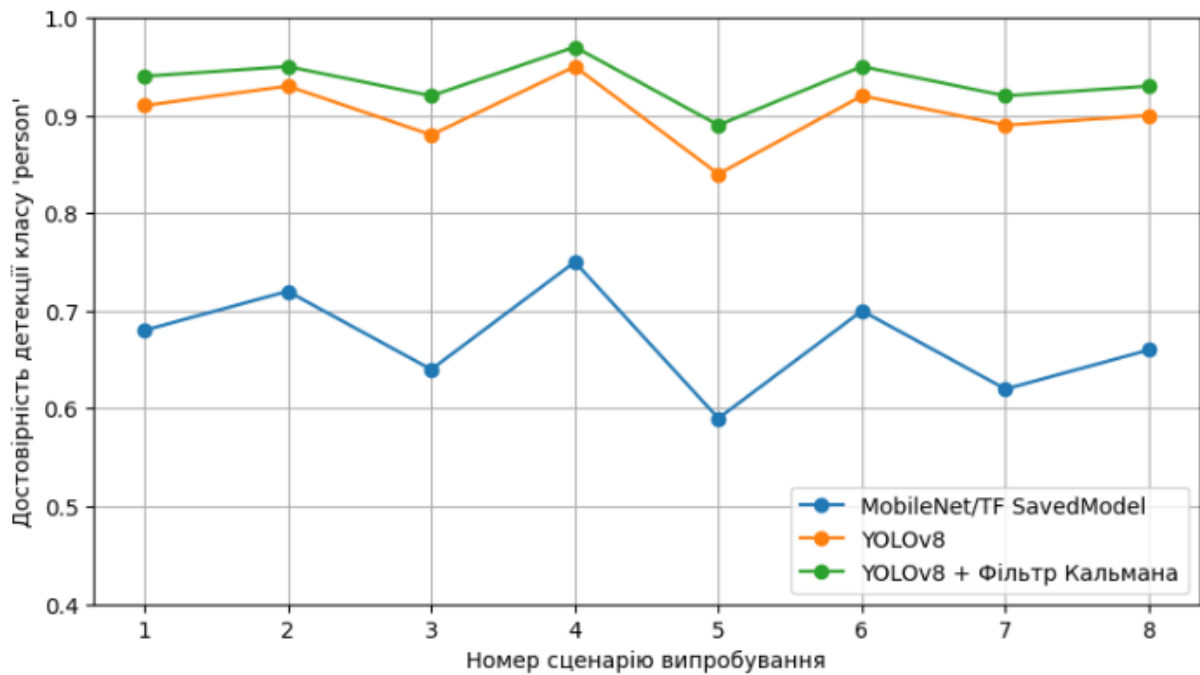


Рисунок 3.14 – Отримані результати при проведення експерименту

Отриманий графік (рис. 3.14) демонструє чітку кількісну перевагу вдосконаленого методу на базі YOLOv8 + фільтр Калмана над базовими підходами в усіх сценаріях випробувань. У середньому достовірність детекції класу “person” для MobileNet/TF SavedModel перебуває в діапазоні 0,59–0,75, що свідчить про суттєву чутливість моделі до змін умов сцени. Базовий YOLOv8 показує значно вищі значення 0,84–0,95, підтверджуючи його кращу узагальнювальну здатність і стійкість до фону та ракурсу. Додавання фільтра Калмана забезпечує додаткове зростання достовірності до рівня 0,89–0,97 та зменшення міжсценарних коливань. Чисельно це проявляється у прирості 0,02–0,05 відносно YOLOv8 у кожному сценарії, що є суттєвим для систем реального часу. Якісно це означає стабілізацію оцінок ідентифікації та компенсацію випадкових помилок нейромережі між кадрами. Особливо помітний ефект у складних сценаріях (3 та 5), де згладжування траєкторій і асоціація детекцій зменшують провали впевненості. Таким чином, результати підтверджують, що інтеграція фільтра Калмана формує більш надійну та придатну для FPV-дронів систему ідентифікації та розпізнавання об’єктів.

3.5 Охорона праці

Розробка та експлуатація систем комп'ютерного зору для FPV-дронів потребує дотримання норм охорони праці, пов'язаних із роботою за комп'ютером, використанням електронного обладнання та високочастотних радіопередавачів. Основними небезпеками є:

- перевтома очей та зоровий дискомфорт від тривалої роботи з моніторами;
- вплив електромагнітного випромінювання від цифрових і аналогових передавачів;
- ризик ураження електричним струмом при підключенні або ремонті обладнання;
- ергономічні навантаження через неправильне розташування робочого місця.

Для роботи з відеопотоком та програмування оптимальним є рівень загального освітлення $E = 300\text{--}500$ лк. Для визначення необхідної кількості світильників можна використати формулу:

$$N = E \cdot S / F \cdot UF \cdot MF , \quad (4.1)$$

де N – кількість світильників;

$E = 400$ лк – необхідна освітленість;

$S = 12$ м² – площа робочого приміщення;

$F = 2000$ лм – світловий потік однієї лампи;

$UF = 0,6$ – коефіцієнт використання світлового потоку;

$MF = 0,8$ – коефіцієнт запасу.

Підставимо дані до формули (4.1):

$$N = 400 \cdot 12 / 2000 \cdot 0,6 \cdot 0,8 = 4800 / 960 = 5 \text{ світильників.}$$

При підключенні обладнання до мережі 220 В, максимальна потужність робочого місця $P = 1,2$ кВт. Струм споживання розраховуватиметься:

$$I = P / U = 1200 / 220 = 5,45 \text{ A} . \quad (4.2)$$

Вибір автоматичного вимикача здійснюється з запасом 1,25, тому:

$$I_{авт} = 5,45 \cdot 1,25 \approx 6,8 \text{ A} \approx 10 \text{ A} \text{ (стандартний номінал)}$$

Дотримання цих норм забезпечує безпечні умови праці при розробці та тестуванні програмного забезпечення для FPV-дронів [19-20].

3.6 Висновки до 3 розділу

У третьому розділі було розроблено та реалізовано систему ідентифікації й розпізнавання об'єктів для системи комп'ютерного зору FPV-дрону, а також експериментально підтверджено її ефективність. Обґрунтовано вибір мови програмування Python та середовища PyCharm, що забезпечили швидке прототипування, відтворюваність експериментів і зручну інтеграцію бібліотек комп'ютерного зору та нейронних мереж. Розроблено формалізований алгоритм роботи системи, який узгоджує етапи захоплення відео, препроцесингу, нейромережевого інференсу та постобробки результатів у режимі, близькому до реального часу. Реалізовано функції розпізнавання та ідентифікації об'єктів на основі попередньо навченої нейронної мережі у форматі TensorFlow SavedModel, що дозволило зосередити дослідження на аналізі швидкодії, затримок інференсу та стабільності роботи системи. Запропонований механізм пропуску кадрів забезпечив компроміс між обчислювальним навантаженням і актуальністю результатів, що відповідає реальним обмеженням FPV-платформ. У ході експериментів виконано порівняльний аналіз детекторів MobileNet/TF SavedModel та YOLOv8, а також вдосконаленого підходу з використанням фільтра Калмана. Отримані результати показали, що YOLOv8 забезпечує суттєво вищу та стабільнішу достовірність детекції в динамічних FPV-сценаріях, а інтеграція фільтра Калмана додатково підвищує надійність ідентифікації та зменшує коливання оцінок між кадрами.

ВИСНОВКИ

У кваліфікаційній роботі розроблено програмний модуль розпізнавання та ідентифікації об'єктів для системи комп'ютерного зору FPV-дрону, спрямований на підвищення швидкодії та ефективності обробки потокового відео в режимі реального часу. Проведено аналіз сучасних підходів до побудови FPV-систем, апаратного забезпечення та методів нейромережевого розпізнавання, що дозволило обґрунтувати вибір архітектури та алгоритмів обробки даних.

Розроблено математичні моделі відеопотоку, детекції та відстеження об'єктів, а також реалізовано програмну систему з використанням нейронної мережі YOLOv8 та методів фільтрації, які забезпечують стабільність ідентифікації в динамічних умовах польоту. Експериментальні дослідження підтвердили доцільність запропонованих рішень і показали покращення показників точності та швидкодії порівняно з базовими підходами.

Отримані результати свідчать про можливість практичного застосування розробленого модуля в системах комп'ютерного зору FPV-дронів та створюють основу для подальшого вдосконалення автономних безпілотних платформ та їх можна віднести до Цілі сталого розвитку 9 «Промисловість, інновації та інфраструктура», а саме 9.5. Створити фінансову та інституційну системи (інноваційну інфраструктуру), що забезпечуватимуть розвиток наукових досліджень та науковотехнічних (експериментальних) розробок [21].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення. – К.: ДП «УкрНДНЦ», 2016. – 31 с.

2. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка освітньо професійних програм: «Комп'ютерно-інтегровані технологічні процеси і виробництва»; «Комп'ютеризовані та робототехнічні системи» / Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В. В. Євсєєв, О. І. Филипенко, О. М. Цимбал. Харків: ХНУРЕ, 2024. 57 с.

3. Пара І. І. Аналіз систем керування fpv дронів з використанням нейронних мереж / І. І. Пара // Автоматизація та Приладобудування («Automation and Development of Electronic Devices» ADED-2025) [Електронний ресурс] : збірник студентських наукових статей / Харківський національний університет радіоелектроніки ; [редкол.: І.Ш. Невлюдов та ін.]. – Харків : ХНУРЕ, 2025. – Вип. 2. – 207с. С.126-129.

4. Гультяєв, Д. А. Комп'ютерна система донаведення FPV-дрона на ціль : дипломний проект ... бакалавра : 123 Комп'ютерна інженерія / Гультяєв Дмитро Антонович. – Київ, 2025. – 118 с.

5. Корнієнко О. В. Розроблення системи комп'ютерного зору для розпізнавання об'єктів на базі нейронної мережи для квадрокоптера : пояснювальна записка до кваліфікаційної роботи здобувача вищої освіти на першому (бакалаврському) рівні, спеціальність 151 – Автоматизація та комп'ютерно-інтегровані технології / О. В. Корнієнко ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків, 2025. – 65 с.

6. Давидов Н. Д. Розроблення системи автоматизації для управління роботизованими механізмами з використанням штучного інтелекту : пояснювальна записка до атестаційної роботи здобувача вищої освіти на другому (магістерському) рівні, спеціальність 174 – Автоматизація, комп'ютерно-

інтегровані технології та робототехніка / Н. Д. Давидов ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків, 2025. – 98 с.

7. Берегун, Р. О. Системи автоматизованого донаведення дронів за допомогою штучного інтелекту : дипломна робота ... бакалавра : 122 Комп'ютерні науки / Берегун Роман Олександрович. – Київ, 2025. – 61 с.

8. Лобода, А. І. Методи обробки та засоби передачі відеоконтенту в режимі реального часу : дипломна робота ... бакалавра : 172 Телекомунікації та радіотехніка / Лобода Артем Ігорович. – Київ, 2024. – 70 с.

9. Зарічковий, О. А. Методи та програмне забезпечення розмітки відеоданих для задач комп'ютерного зору : дис. ... д-ра філософії : 12 Інформаційні технології / Зарічковий Олександр Анатолійович. – Київ, 2025. – 261 с.

10. Риба, О. С. Дослідження та програмна реалізація системи відеоаналітики на основі систем машинного зору : кваліфікаційна магістерська робота : спец. 123 "Комп'ютерна інженерія" / наук. кер. Г. М. Дреєва ; Центральноукраїн. нац. техн. ун-т. - Кропивницький : ЦНТУ, 2024. - 127 с.

11. Савка, М. С. Застосування нейронних мереж в задачах розпізнавання об'єктів : магістерська дис. : 171 Електроніка / Савка Максим Сергійович. – Київ, 2022. – 117 с.

12. Костюк І. В. Розробка інформаційної системи симуляції керування віддаленими пристроями : кваліфікаційна робота бакалавра спеціальності 121 "Інженерія програмного забезпечення" / наук. керівник А. Г. Кривохата. Запоріжжя : ЗНУ, 2025. 54 с.

13. Міндзіброцький Р. С. Комп'ютеризована система керування FPV дроном для симулятора польоту : робота на здобуття кваліфікаційного ступеня бакалавра : спец. 123 - комп'ютерна інженерія / наук. кер. Ю. З. Лещишин. Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2024. 85 с.

14. Раєнко Т. В. Аналіз методів підключення пультів керування FPV-дронами до ПК для симуляції польоту / Т. В. Раєнко // Автоматизація та Приладобудування («Automation and Development of Electronic Devices» ADED-

2025) : збірник студентських наукових статей. – Харків : ХНУРЕ, 2025. – Вип. 1. – С. 211-216.

15. Коваль А. Р. Розроблення літаючого макету дрону на базі ARM мікроконтролера для збору та обробки даних у реальному часі : пояснювальна записка до кваліфікаційної роботи здобувача вищої освіти на першому (бакалаврському) рівні, спеціальність 151 – Автоматизація та комп'ютерно-інтегровані технології / А. Р. Коваль ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків, 2025. – 85 с.

16. Пшеничний, Д. В. Система підтримки прийняття рішень для підбірки компонентів FPV дронів з використанням трансформерних моделей : дипломна робота ... бакалавра : 122 Комп'ютерні науки / Пшеничний Дмитро Володимирович. – Київ, 2025. – 94 с.

17. Каран, В. Д. Дослідження використання штучного інтелекту на базі безпілотних літальних апаратів : дипломна робота ... бакалавра : 122 Комп'ютерні науки / Каран Володимир Дмитрович. – Київ, 2024. – 96 с.

18. Іванов О. В. Розроблення автоматизованої системи зору на основі FPV : пояснювальна записка до атестаційної роботи здобувача вищої освіти на другому (магістерському) рівні, спеціальність 151 - Автоматизація та комп'ютерно-інтегровані технології / О. В. Іванов ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки – Харків, 2021 – 75 с.

19. Фесенко А. О. Аналіз характеристик параметрів навколишнього середовища у виробничих приміщеннях / А. О. Фесенко // Автоматизація та Приладобудування («Automation and Development of Electronic Devices» ADED-2025) : збірник студентських наукових статей. – Харків : ХНУРЕ, 2025. – Вип. 2. – С. 164-170.

20. ДСТУ 3400:2015. Охорона праці. Безпека при роботі з електричними установками. – К.: Національний стандарт України, 2015. – 33 с.

21. Ціль 9. Промисловість, інновації та інфраструктура // Дія, 2025. URL: https://business.diia.gov.ua/entrepreneurhandbook/item/cil_9_promislovist_innovaciy_i_ta_infrastruktura (дата звернення: 06.12.2025).