

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ  
ІМЕНІ АДМІРАЛА МАКАРОВА

Кваліфікаційна наукова  
праця на правах рукопису

РОБОТЬКО СЕРГІЙ ПАВЛОВИЧ

УДК 681.513.2:004.932

**ДИСЕРТАЦІЯ**

БАГАТОРІВНЕВА АВТОМАТИЗОВАНА СИСТЕМА КЕРУВАННЯ  
РОБОТИЗОВАНОЮ ПЛАТФОРМОЮ ДЛЯ ЗАДАЧ ПРОСТОРОВОГО  
МОНІТОРИНГУ НА ОСНОВІ КОМП'ЮТЕРНОГО ЗОРУ

151 – Автоматизація та комп'ютерно-інтегровані технології  
15 – Автоматизація та приладобудування

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей,  
результатів і текстів інших авторів мають посилання на відповідне джерело

\_\_\_\_\_ /С. П. РОБОТЬКО/

Науковий керівник:  
Топалов Андрій Миколайович  
кандидат технічних наук, доцент

## АНОТАЦІЯ

Дисертаційна робота присвячена розв'язанню актуальної науково-технічної задачі підвищення ефективності та безпеки просторового моніторингу в умовах, де рівень ризиків є критичним, зокрема у сценаріях гуманітарного розмінування та ідентифікації предметів. Традиційні підходи, що ґрунтуються на ручному обстеженні, залишаються повільними й небезпечними, а тому автоматизація на базі безпілотних платформ розглядається як один із найбільш перспективних напрямів розвитку відповідних технологій. У цьому контексті комп'ютерний зір та інтелектуальні методи аналізу даних дають змогу перейти від фрагментарного збору інформації до безперервного конвеєра обробки, який може підтримувати прийняття рішень у режимі, наближеному до реального часу, та зменшувати залежність результату від людського фактору.

Метою дослідження є підвищення ефективності, швидкості та безпеки процесів виявлення небезпечних предметів за допомогою БПЛА шляхом створення багаторівневої автоматизованої системи керування, яка забезпечує мультимодальне злиття даних, семантичну верифікацію об'єктів і адаптивну корекцію траєкторії польоту під час виконання місії. Запропонований підхід передбачає, що безпілотна платформа виступає не лише носієм сенсорів, а активним агентом, який здатний змінювати поведінку залежно від контексту та узгоджених сигналів від різномірних джерел інформації. Така логіка переводить моніторинг із парадигми пасивного збору та ретрансляції даних до замкненого циклу, у якому результати аналізу безпосередньо впливають на наступні дії платформи. Формулювання мети та ключових вимог до системи узгоджено з необхідністю одночасно підвищувати надійність розпізнавання та забезпечувати оперативність реакції на події, що виникають у польових умовах.

Концептуально система будується як ієрархічна трирівнева архітектура «борт – наземна станція – хмара», де кожен рівень виконує власну роль у

наскрізному процесі перетворення необроблених сенсорних масивів у верифіковану семантичну інформацію. У межах такої організації реалізується принцип паралельної асинхронної обробки інформаційних потоків у контурі «Edge–Ground–Cloud». Запропонований підхід забезпечує динамічне оркестрування обчислювальних процесів, де критичні для безпеки польоту задачі виконуються синхронно на бортовому обчислювачі з детермінованою латентністю, тоді як ресурсомісткі задачі семантичної верифікації та глобального планування місії виконуються асинхронно на наземному та хмарному рівнях. Це дозволяє збалансувати затримку прийняття рішень і глибину аналізу, а також раціонально використовувати обмежену пропускну здатність каналів зв'язку під час польових робіт.

Методична основа роботи поєднує підходи теорії автоматичного керування, комп'ютерного зору, глибокого навчання, мультимодального аналізу та ймовірнісного прийняття рішень. У модулі комп'ютерного зору застосовано згорткові нейронні мережі сімейства YOLOv8 для детектування об'єктів у відеопотоці, а для підвищення семантичної надійності розпізнавання залучено візуально-мовні моделі, що дають змогу виконувати уточнення типу об'єкта та зменшувати частку хибних тривог у складних сценах. Фізичним каналом підтвердження виступає металодетектор, сигнали якого синхронізуються з відеоданими та телеметрією, утворюючи основу для мультисенсорного аналізу. Практична реалізація передбачає узгодження даних за часом і координатами, а також формування журналів і артефактів випробувань, що забезпечують трасованість від фінальної мітки ризику на карті до первинних вимірювань.

Ключовим науково-прикладним результатом є формалізація математичної моделі зваженої агрегації довіри, яка об'єднує нормалізовані виходи візуального детектора, семантичного аналізу та металоканалу в інтегральний показник імовірності загрози, що використовується для прийняття рішень у системі. У роботі показано, що пряме оцінювання повних умовних імовірностей для кожного сенсора в реальному часі є практично

складним, тому застосовано наближення у вигляді зваженого голосування, яке апроксимує байєсівську логіку оновлення та дозволяє гнучко налаштовувати робочу точку системи між чутливістю та специфічністю залежно від умов місії. Вагові коефіцієнти інтерпретуються як параметри довіри до каналів та можуть коригуватися з огляду на характеристики середовища, зокрема коли візуальний канал деградує через фон або рослинність, а канал металошукача зберігає інформативність. На основі такої інтеграції формується кінцевий показник, який порівнюється з порогом дії та ініціює або припиняє подальші кроки аналізу й керування.

Важливою особливістю запропонованого підходу є те, що модель злиття даних вбудована у контур керування польотом. Розвинуто метод автоматичного адаптивного керування, який організовує замкнений цикл «детекція – семантична верифікація – корекція місії» на базі протоколу MAVLink, завдяки чому система здатна автоматично ініціювати детальніший огляд підозрілої зони, змінювати режим польоту та уточнювати траєкторію для підвищення якості даних. Такий зв'язок між розпізнаванням і керуванням дозволяє перетворити платформу з пасивного реєстратора на інструмент активного збору доказів, коли додаткові маневри виконуються не за командою оператора, а за умовою, визначеною агрегованою оцінкою ризику.

Експериментальна частина дослідження спрямована на перевірку адекватності запропонованих моделей та оцінювання точності й часових характеристик системи в умовах, наближених до реальних. Для навчання та налаштування детектора сформовано спеціалізований набір даних, що містить 2500 кадрів із зображеннями вибухонебезпечних предметів визначених типів у реальних умовах експлуатації, що дало змогу забезпечити стійкість моделі до варіативності фону, масштабу та освітлення. Показано, що формалізована модель зваженої агрегації довіри підвищує надійність ідентифікації та дає змогу мінімізувати помилки першого й другого роду, забезпечуючи високу точність розпізнавання до 98% в умовах невизначеності зовнішнього середовища. Додатково продемонстровано операційний виграш



багаторівневої інтеграції, що відображається у зростанні інтегральної ймовірності виявлення при використанні кількох каналів підтвердження.

Практична цінність отриманих результатів підтверджується апробацією та впровадженням розробок на рівні діючого програмно-апаратного комплексу та впровадженням результатів у виробничій організації, що засвідчено актами. Реалізована система включає бортовий обчислювальний вузол, засоби збору відео та сигналів металодетектора, механізми синхронізації з телеметрією, а також інтеграцію з наземною станцією керування та хмарними сервісами аналізу. У сукупності це забезпечує цілісний технологічний ланцюг від збору даних до формування геоприв'язаних міток ризику та підтримки рішень щодо повторного обльоту чи деталізації обстеження.

Отже, дисертаційне дослідження пропонує інтегрований підхід до автоматизації пошукових місій БПЛА, у якому багаторівнева архітектура розподіляє обчислення між бортом, наземною станцією та хмарою. Мультимодальна модель злиття даних підвищує достовірність ідентифікації, а замкнений контур керування забезпечує активну адаптацію траєкторії й режимів польоту на основі результатів комп'ютерного зору та сенсорного підтвердження. Сукупність запропонованих рішень формує основу для практичного застосування у задачах просторового моніторингу з високими вимогами до надійності, швидкодії та безпеки.

**Ключові слова:** багаторівневий аналіз відео-зображення, візуально-мовні моделі, комп'ютерний зір, гуманітарне розмінування, безпілотний літальний апарат (БПЛА), хмарна обробка даних, система автоматичного керування, інтелектуальне управління, штучна нейронна мережа, аналіз перетворення сигналу, центр дистанційного керування, обробка зображень, аналітично-структурне моделювання, бездротова комп'ютерна мережа, одноплатний мікрокомп'ютер.

## ABSTRACT

The thesis addresses the urgent scientific and technical problem of enhancing the efficiency and safety of spatial monitoring in environments where the cost of error is critically high, particularly in humanitarian demining and unexploded ordnance (UXO) search scenarios. In modern conflicts, the contamination of territories with mines and munitions creates long-term risks for the civilian population and obstructs infrastructure recovery. Conventional approaches based on manual surveying remain slow and hazardous; therefore, automation based on unmanned platforms is considered one of the most promising directions for technology development. In this context, computer vision and intelligent data analysis methods enable a transition from fragmented information gathering to a continuous processing pipeline capable of supporting decision-making in near-real-time and reducing reliance on the human factor.

The aim of the research is to increase the efficiency, speed, and safety of explosive object detection processes using UAVs by creating a multi-level automated control system. This system ensures multi-modal data fusion, semantic object verification, and adaptive flight trajectory correction during mission execution. The proposed approach posits that the unmanned platform acts not merely as a sensor carrier but as an active agent capable of modifying its behavior based on context and reconciled signals from heterogeneous information sources. Such logic shifts monitoring from a passive «capture-and-transmit» mode to a closed-loop cycle where analysis results directly influence the platform's subsequent actions. The formulation of the goal and key system requirements is aligned with the necessity to simultaneously improve recognition reliability and ensure rapid response to events occurring in field conditions.

Conceptually, the system is built upon a hierarchical three-level «Edge–Ground–Cloud» architecture, where each level plays a distinct role in the end-to-end process of transforming raw sensor data into verified semantic information. This organization implements the principle of parallel asynchronous processing of

information flows within the «Edge–Ground–Cloud» loop. The proposed approach ensures the dynamic orchestration of computational processes: safety-critical tasks are executed synchronously on the onboard computer with deterministic latency, while resource-intensive tasks of semantic verification and global mission planning are performed asynchronously at the ground and cloud levels. This balances decision-making latency with analysis depth and rationalizes the use of limited communication bandwidth during field operations. The architectural principles reflect the modern paradigm of distributed heterogeneous computing, wherein safety-critical decisions must be made with minimal latency, while refinement and confirmation can be offloaded to more powerful computational resources.

The methodological basis of the work combines approaches from automatic control theory, machine learning [17, 19], computer vision, deep learning, multi-modal analysis, and probabilistic decision-making. The computer vision module utilizes Convolutional Neural Networks (CNNs) of the YOLOv8 family for object detection in the video stream. To enhance semantic recognition reliability, Vision-Language Models (VLMs) are employed to refine object types and reduce false positives in complex scenes. A metal detector serves as the physical confirmation channel, with its signals synchronized with video data and telemetry to form the basis for multi-sensor analysis. Practical implementation involves temporal and spatial data alignment, as well as the generation of logs and test artifacts, ensuring traceability from the final risk marker on the map back to the primary measurements.

A key scientific and applied result is the formalization of a mathematical model for weighted trust aggregation. This model combines the normalized outputs of the visual detector, semantic analysis, and the metal detector channel into an integral threat probability metric used for system decision-making. The study demonstrates that direct estimation of full conditional probabilities for each sensor in real-time is practically complex; therefore, an approximation in the form of weighted voting is applied. This approximates Bayesian update logic and allows for flexible tuning of the system's operating point between sensitivity and specificity depending on mission conditions. Weight coefficients are interpreted as trust

parameters for the channels and can be adjusted based on environmental characteristics—for instance, when the visual channel degrades due to background or vegetation while the metal channel retains informativeness. Based on this integration, a final metric is formed, which is compared against an action threshold to either initiate or terminate further analysis and control steps.

An important feature of the proposed approach is the embedding of the data fusion model into the flight control loop. A method of automatic adaptive control has been developed, organizing a closed «detection – semantic verification – mission correction» loop based on the MAVLink protocol. Consequently, the system is capable of automatically initiating a more detailed inspection of a suspicious zone, changing flight modes, and refining the trajectory to improve data quality. This link between recognition and control transforms the platform from a passive recorder into an active evidence-gathering tool, where additional maneuvers are executed not by operator command but based on conditions defined by the aggregated risk assessment.

The experimental part of the research focuses on verifying the adequacy of the proposed models and evaluating the system's accuracy and temporal characteristics in near-real-world conditions. A unique dataset containing 2,500 frames with images of specific explosive object types in real operating conditions was created for detector training and tuning, ensuring model robustness to variations in background, scale, and lighting. It is shown that the formalized weighted trust aggregation model increases identification reliability and minimizes Type I and Type II errors, achieving a recognition accuracy of up to 98% under environmental uncertainty. Additionally, the operational gain of multi-level integration is demonstrated, reflected in the increase of the integral detection probability when using multiple confirmation channels.

The practical value of the results is confirmed by the development of a functional hardware-software complex and the implementation of results in a production organization, as certified by an implementation act. The realized system includes an onboard computing node, tools for video and metal detector signal

acquisition, telemetry synchronization mechanisms, and integration with a ground control station and cloud analysis services. Collectively, this ensures a holistic technological chain from data collection to the generation of geo-referenced risk markers and decision support regarding re-flights or detailed surveys.

In conclusion, the dissertation offers an integrated approach to automating UAV search missions, wherein a multi-level architecture distributes computations between the onboard unit, ground station, and cloud. A multi-modal data fusion model enhances identification credibility; and a closed control loop ensures active adaptation of trajectory and flight modes based on computer vision results and sensor confirmation. The aggregate of the proposed solutions forms a foundation for practical application in spatial monitoring tasks with high requirements for reliability, speed, and safety.

Keywords: multi-level video image analysis, Vision language models, computer vision, humanitarian demining, unmanned aerial vehicle (UAV), cloud data processing, automatic control system, intelligent management, artificial neural network, signal conversion analysis, remote control center, image processing, analytical and structural modeling, wireless computer network, single-board microcomputer.

## СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

### Публікації в іноземних виданнях Scopus

1. Robotko S., Topalov A., Nechai I., Vasylenko R. Multimodal data fusion for security screening with weighted confidence aggregation and escalation in Edge-cloud systems, Proceedings of the Bulgarian Academy of Sciences, Vol. 79 No. 1. 2026. pp. 13–50. DOI: 10.7546/CRABS.2026.01.09

### Статті у фахових виданнях України:

1. Роботько С. П. Багаторівневий аналіз відеозображення на базі Raspberry Pi та ChatGPT-4 Vision, Методи та прилади контролю технологічних параметрів, № 1(54). 2025. с. 78-88. DOI: 10.31471/1993-9981-2025-1(54)
2. Роботько С., Луцак Д. Багаторівневий підхід виявлення та розпізнавання небезпечних предметів за допомогою БПЛА із застосуванням комп'ютерного зору, Вісник Харківського національного автомобільно-дорожнього університету, № 111 2025. с. 203-209. DOI: 10.30977/BUL.2219-5548.2025.111.0
3. Роботько С.П., Топалов А.М. Багаторівнева архітектура системи автоматичного керування БПЛА для здійснення пошукових місій за відеоаналізом та металодетекцією, Оптико-електронні інформаційно-енергетичні технології, Том 50 № 2, 2025. с. 114-124. DOI: 10.31649/1681-7893-2025-50-2
4. Robotko S., Zivenko O. Effectiveness of vision–language models (VLMs) for ground-object recognition in a multi-level Edge–cloud uav architecture, Науковий вісник Херсонської державної морської академії, Том 2 № 31, 2025. с. 19-29. DOI: 10.33815/2313-4763.2025.2.31.019-029
5. Zheng Y., Robotko S.P., Dong C., Wang J., Ravliuk V.V., Topalov A.M. Cloud-based GPS signal processing of the drone using Raspberry Pi and Pixhawk 6c flight controller. Вчені записки Таврійського національного університету імені В.І. Вернадського, т. 34 (73), № 6, 2023. с. 163-168. DOI: 10.32782/2663-5941/2023.6/24

6. Dong C., Wang J., Robotko S.P., Susak O.M., Topalov A.M., Kolomiets V.V. Features of human-machine interaction in the system of wireless control of a mobile robot. *Міжвузівський збірник «Наукові нотатки», Луцьк, № 76, 2023. с. 166-172. DOI: 10.36910/775.24153966.2023.76*

#### **Матеріали конференцій:**

1. Robotko S. AI-Driven OCR in Practice: Assessing ChatGPT-4o and Google Vision as Alternatives to Manual Optical Character Recognition, *2025 15th International Conference on Advanced Computer Information Technologies*, 2025. pp. 944-947. DOI: 10.1109/ACIT65614.2025.11185827.
2. Роботко С.П. Впровадження хмарного аналізу даних для аналогових металодетекторів з використанням мікрокомп'ютерних систем. *XV Міжнародна науково-технічна конференція «Інновації в суднобудуванні та океанотехніці»*, Миколаїв, 2024, с. 1011–1013.
3. Роботко С.П., Виявлення вибухонебезпечних предметів за допомогою БПЛА та візуально-мовних моделей, *«Могилянські читання – 2025: досвід та тенденції розвитку суспільства в Україні: глобальний, національний та регіональний аспекти» XXVIII Всеукраїнська науково-практична конференція*, 2025. сс. 132-137.
4. Robotko S., Susak O., Topalov A., Gerasin O.S., Buznyk A., Zivenko O. Enabling Cloud-based Data Analysis for Analog Metal Detectors using Microcomputer Systems. *14th International Conference on Advanced Computer Information Technologies (ACIT)*, 2024, pp. 670–673. URL: <https://ieeexplore.ieee.org/abstract/document/10712463>
5. Khoda V., Leshchuk N., Topalov A., Robotko S., Klymenko O., Nekrasov S. Computerized Lathe Control System based on Internet of Things Technology. *2024 14th International Conference on Advanced Computer Information Technologies (ACIT)*, 2024, pp. 674–670. DOI: 10.1109/ACIT62333.2024.10712548.

6. Роботько С. Багаторівнева хмарна обробка відео машинного зору. *Всеукраїнська науково-технічна конференція студентів, аспірантів та молодих вчених*, 2024.
7. Topalov A., Robotko S., Zaytsev V., Zaytsev D., Zaytsev V. Lukashova V. Computerized Monitoring System with Cloud Processing of the Hovercraft's GPS Signal, *14th International Conference Dependable Systems, Services & Technologies (DESSERT'2024)*, 2024. pp. 1-5. DOI: 10.1109/DESSERT65323.2024.11122034.
8. Роботько С.П. Інформаційний комплекс технічного зору з багаторівневою передачею даних на базі одноплатного комп'ютера. *XIII Міжнародна науково-технічна конференція «Інновації в суднобудуванні та океанотехніці»*, онлайн, 2022.
9. Robotko S., Topalov A., Nekrasov S., Zaitsev V., Zaitsev D. Machine Learning and Modeling of the Impact of Trademark Filings on GDP Growth based on Python. *MoMLeT+DS 2023: 5th International Workshop on Modern Machine Learning Technologies and Data Science*, 3 Jun, Lviv, 2023. pp. 65-76. URL: <https://ceur-ws.org/Vol-3426/paper6.pdf>
10. Топалов А.М., Поворознюк О.С., Роботько С.П. Технічна реалізація мікропроцесорної системи контролю енергоспоживання серводвигунів мобільного робота. *Сучасні проблеми автоматики та електротехніки: матеріали Всеукраїнської науково-технічної конференції*. Миколаїв: НУК, 2023, с. 20–21.
11. Robotko S., Topalov A., Nekrasov S., Zaitsev V., Zaitsev D. A Machine Learning Study on World Economic Impact of Intellectual Property Rights on GDP Growth with a Focus on Poland. *2023 13th International Conference on Advanced Computer Information Technologies (ACIT)*, Wrocław, Poland, 2023, pp. 261–264. DOI: 10.1109/ACIT58437.2023.10275475
12. Роботько С.П., Топалов А.М. Хмарна обробка GPS даних на базі Raspberry Pi з політного контролера дрона. *I Міжнародна науково-*



практична інтернет-конференція «*Achievements of 21st Century Scientific Community*», 2023, с. 384–385.

URL: <http://www.wayscience.com/wp-content/uploads/2023/09/Conference-Proceedings-September-14-15-2023.pdf>

13. Balitskyi O., Topalov A., Robotko S., Povorozniuk O. Features of the magnetorheological damper in comparison with other types of magnetic dampers. *International Multidisciplinary Scientific Internet Conference «World of scientific research. Issue 25»*, December 14–15, 2023, Ternopil – Opole, pp. 340–343. URL: <https://www.economy-confer.com.ua/full-article/5185/>

#### **Патенти:**

1. Патент України на корисну модель № 157368 «Пристрій для вимірювання товщини струмопровідної металевої плівки в процесі напилювання у вакуумі». / Тростинський М.М., Топалов А.М., Карпеченко А.А., Герасін О.С., Бобров М.М., Роботко С.П., Хо́да В.М., Не́до А.О. № u202400179; заявл. 11.01.2024; опубл. 09.10.2024.

## ЗМІСТ

|                                                                                                                                                          |    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| АНОТАЦІЯ                                                                                                                                                 | 2  |
| ПЕРЕЛІК СКОРОЧЕНЬ                                                                                                                                        | 17 |
| ВСТУП                                                                                                                                                    | 20 |
| РОЗДІЛ 1. СУЧАСНИЙ СТАН ПРОБЛЕМИ АВТОМАТИЗОВАНОГО<br>КЕРУВАННЯ РОБОТИЗОВАНИМИ ПЛАТФОРМАМИ ДЛЯ ЗАДАЧ<br>ПРОСТОРОВОГО МОНІТОРИНГУ НАВКОЛИШНЬОГО СЕРЕДОВИЩА | 27 |
| 1.1 Огляд автоматичних систем керування роботизованих платформ                                                                                           | 27 |
| 1.1.1 Архітектура систем керування та навігації                                                                                                          | 27 |
| 1.1.2 Програмно-апаратна реалізація польотних контролерів                                                                                                | 29 |
| 1.1.3 Протоколи взаємодії та маршрутизація повідомлень (MAVLink)                                                                                         | 30 |
| 1.1.4 Взаємодія бортового комп'ютера та автопілота                                                                                                       | 32 |
| 1.2 Огляд існуючих апаратних рішень для моніторингу та розмінування.                                                                                     | 34 |
| 1.3 Розвиток методів комп'ютерного зору                                                                                                                  | 37 |
| 1.4 Архітектурні рішення у комп'ютерному зорі                                                                                                            | 43 |
| 1.4.1 Збільшення глибини та складності нейронних мереж                                                                                                   | 45 |
| 1.4.2 Комбіновані підходи CNN і трансформери                                                                                                             | 46 |
| 1.4.3 Легковагі моделі для мобільних і вбудованих пристроїв                                                                                              | 48 |
| 1.4.4 Мультимодальні моделі                                                                                                                              | 49 |
| 1.5 Формалізація та порівняльний аналіз рівнів обчислень                                                                                                 | 51 |
| 1.6 Методи збору, обробки та збереження даних на бортових<br>обчислювальних платформах                                                                   | 53 |
| 1.6.1 Огляд та класифікація сучасних роботизованих платформ                                                                                              | 54 |
| 1.6.2 Застосування комп'ютерного зору для різних типів платформ                                                                                          | 56 |
| 1.6.3 Методи збору, збереження та обробки даних на Edge рівні                                                                                            | 59 |
| 1.7 Постановка задачі                                                                                                                                    | 61 |
| 1.8 Висновки за розділом 1                                                                                                                               | 62 |
| РОЗДІЛ 2. АРХІТЕКТУРА АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ<br>БАГАТОРІВНЕВОЇ РОБОТИЗОВАНОЇ ПЛАТФОРМИ ПРОСТОРОВОГО<br>МОНІТОРИНГУ                                      | 65 |
| 2.1 Функціональна організація багаторівневої системи обробки відеоданих                                                                                  | 68 |

|                                                                                                                              |            |
|------------------------------------------------------------------------------------------------------------------------------|------------|
| 2.2 Обґрунтування вибору та конструктивно-функціональна реалізація безпілотної платформи гексакоптерного типу                | 74         |
| 2.3 Динамічна модель гексакоптера та рівняння руху в системі керування                                                       | 81         |
| 2.4 Апаратна реалізація та інтеграція імпульсного металошукача Tracker-2 у сенсорний комплекс                                | 88         |
| 2.5 Одноплатний компютер Raspberry Pi 4B як бортовий обчислювальний модуль системи                                           | 93         |
| 2.6 Висновки за розділом 2                                                                                                   | 96         |
| <b>РОЗДІЛ 3. АЛГОРИТМІЧНО-ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ МОДУЛЮ КОМП'ЮТЕРНОГО ЗОРУ</b>                                               | <b>99</b>  |
| 3.1 Математична модель багаторівневої системи аналізу для виявлення небезпечних об'єктів                                     | 99         |
| 3.2 Виявлення вибухонебезпечних предметів за допомогою нейронної мережі YOLOv8                                               | 109        |
| 3.2.1 Загальні відомості про архітектуру YOLOv8                                                                              | 109        |
| 3.2.2 Роль хмарного рівня у багаторівневій архітектурі                                                                       | 113        |
| 3.2.3 Підготовка даних для навчання YOLO v8                                                                                  | 116        |
| 3.2.4 Навчання, валідація та калібрування нейромережі YOLOv8                                                                 | 119        |
| 3.2.5 Постпроцесинг даних від модулів системи, геопросторова прив'язка та інтеграція з ГІС                                   | 122        |
| 3.2.6 Експлуатаційні характеристики хмарного сервісу та обґрунтування вибору YOLOv8                                          | 123        |
| 3.3 Хмарна обробка: семантична верифікація з використанням Gemini 2.5 Flash                                                  | 125        |
| 3.4 Застосування OpenAI API для аналізу зображень                                                                            | 128        |
| 3.5 Порівняльний аналіз ChatGPT-4.1 Vision та Gemini 2.5 Flash                                                               | 133        |
| 3.6 Реалізація замкненого контуру автоматичного керування БПЛА на основі мультимодального аналізу та семантичної верифікації | 138        |
| 3.7 Висновки за розділом 3                                                                                                   | 154        |
| <b>РОЗДІЛ 4. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ІНТЕГРОВАНОЇ СИСТЕМИ КЕРУВАННЯ.</b>                                                | <b>156</b> |
| 4.1 Методика та сценарії експериментів                                                                                       | 156        |
| 4.2 Тестування модуля комп'ютерного зору                                                                                     | 166        |

|                                                                             |     |
|-----------------------------------------------------------------------------|-----|
|                                                                             | 16  |
| 4.3 Тестування модуля металошукача                                          | 177 |
| 4.4 Злиття даних: інтеграція результатів комп'ютерного зору та металошукача | 184 |
| 4.5 Висновки за розділом 4                                                  | 191 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ                                                  | 194 |
| ДОДАТОК А                                                                   | 202 |
| ДОДАТОК Б                                                                   | 206 |
| ДОДАТОК В                                                                   | 208 |
| ДОДАТОК Г                                                                   | 215 |
| ДОДАТОК Ґ                                                                   | 222 |
| ДОДАТОК Д                                                                   | 224 |
| ДОДАТОК Е                                                                   | 225 |
| ДОДАТОК Є                                                                   | 226 |
| ДОДАТОК Ж                                                                   | 227 |

## ПЕРЕЛІК СКОРОЧЕНЬ

|       |                                                                                                               |
|-------|---------------------------------------------------------------------------------------------------------------|
| АЦП   | – аналого-цифровий перетворювач                                                                               |
| БПЛА  | – безпілотний літальний апарат                                                                                |
| ВНП   | – вибухонебезпечний предмет                                                                                   |
| ГІС   | – геоінформаційна система                                                                                     |
| ПЗ    | – програмне забезпечення                                                                                      |
| ШИМ   | – широтно-імпульсна модуляція (те саме, що PWM)                                                               |
| AI    | – Artificial Intelligence (штучний інтелект)                                                                  |
| API   | – Application Programming Interface (програмний інтерфейс застосунку)                                         |
| ASIC  | – Application-Specific Integrated Circuit (інтегральна схема спеціального призначення)                        |
| AGL   | – Above Ground Level (висота над рівнем землі)                                                                |
| CLAHE | – Contrast Limited Adaptive Histogram Equalization (адаптивне вирівнювання гістограми з обмеженням контрасту) |
| CNN   | – Convolutional Neural Network (згортова нейронна мережа)                                                     |
| CV    | – Computer Vision (комп'ютерний зір)                                                                          |
| EKF   | – Extended Kalman Filter (розширений фільтр Калмана)                                                          |
| ELRS  | – ExpressLRS (протокол радіозв'язку з низькою затримкою)                                                      |
| ESC   | – Electronic Speed Controller (електронний регулятор швидкості)                                               |
| FAR   | – False Alarm Rate (ймовірність хибних тривог)                                                                |
| FPGA  | – Field-Programmable Gate Array (програмована користувачем вентильна матриця)                                 |
| FPS   | – Frames per Second (кадрів на секунду)                                                                       |
| GCS   | – Ground Control Station (наземна станція керування)                                                          |
| GNC   | – Guidance, Navigation, and Control (наведення, навігація та керування)                                       |
| GPS   | – Global Positioning System (глобальна система позиціювання)                                                  |
| GPU   | – Graphics Processing Unit (графічний процесор)                                                               |

|         |                                                                                            |
|---------|--------------------------------------------------------------------------------------------|
| Haar    | – Haar Cascades (каскади Хаара)                                                            |
| HOG     | – Histogram of Oriented Gradients (гістограми орієнтації градієнтів)                       |
| IMU     | – Inertial Measurement Unit (інерціальний вимірювальний модуль)                            |
| IoU     | – Intersection over Union (міра перекриття областей / коефіцієнт Жаккара)                  |
| ISP     | – Image Signal Processor (процесор обробки сигналів зображення)                            |
| LiDAR   | – Light Detection and Ranging (активний далекомір оптичного діапазону)                     |
| LLM     | – Large Language Model (велика мовна модель)                                               |
| mAP     | – mean Average Precision (середня точність – метрика оцінки якості детектування)           |
| MAVLink | – Micro Air Vehicle Link (протокол обміну даними для мікро-БПЛА)                           |
| NMS     | – Non-Maximum Suppression (алгоритм придушення немаксимумів)                               |
| NPU     | – Neural Processing Unit (нейронний процесор / прискорювач нейромереж)                     |
| OCR     | – Optical Character Recognition (оптичне розпізнавання символів)                           |
| VQA     | – Visual Question Answering (Візуальні відповіді на запитання)                             |
| ORB     | – Oriented FAST and Rotated BRIEF (орієнтований FAST та повернутий BRIEF)                  |
| PID     | – Proportional-Integral-Derivative (пропорційно-інтегрально-диференціальний регулятор)     |
| R-CNN   | – Region-based Convolutional Neural Network (згорткова нейронна мережа на основі регіонів) |
| RESTful | – Representational State Transfer (передача стану представлення)                           |
| ROI     | – Region of Interest (зона/область інтересу)                                               |
| ROS     | – Robot Operating System (операційна система для роботів)                                  |
| SLAM    | – Simultaneous Localization and Mapping (одночасна локалізація та картування)              |
| SSD     | – Single Shot Multibox Detector (одностадійний детектор)                                   |
| SURF    | – Speeded-Up Robust Features (прискорені стійкі ознаки)                                    |

|      |                                                                                        |
|------|----------------------------------------------------------------------------------------|
| SWaP | – Size, Weight, and Power (розмір, вага та енергоспоживання)                           |
| TPU  | – Tensor Processing Unit (тензорний процесор)                                          |
| UBEC | – Universal Battery Elimination Circuit (універсальний ланцюг живлення)                |
| UGV  | – Unmanned Ground Vehicle (безпілотний наземний транспортний засіб)                    |
| USV  | – Unmanned Surface Vehicle (безпілотне надводне судно)                                 |
| UUV  | – Unmanned Underwater Vehicle (безпілотний підводний апарат)                           |
| VIO  | – Visual-Inertial Odometry (візуально-інерціальна одометрія)                           |
| ViT  | – Vision Transformer (візуальний трансформер)                                          |
| VLM  | – Vision-Language Model (візуально-мовна модель)                                       |
| VTOL | – Vertical Take-Off and Landing (вертикальний зліт і посадка)                          |
| YOLO | – You Only Look Once (сімейство нейромереж для детектування об'єктів у реальному часі) |

## ВСТУП

**Актуальність теми.** Сучасний стан глобальної безпеки характеризується безпрецедентним зростанням загроз, пов'язаних із забрудненням значних територій вибухонебезпечними предметами (ВНП). Особливу гостроту ця проблема набула в контексті сучасних військових конфліктів, де щільність небезпечних предметів створюють довгострокові перешкоди для відновлення інфраструктури та безпеки цивільного населення. Традиційні методи розмінування, що спираються на ручну працю саперів та використання наземних щупів і металодетекторів, є критично повільними, ресурсомісткими та, що найважливіше, пов'язані з високим ризиком для життя персоналу. Автоматизація процесів виявлення загроз за допомогою роботизованих платформ розглядається міжнародною науковою спільнотою як найбільш перспективний напрямок розвитку технологій гуманітарного розмінування. Проте, існуючі рішення стикаються з низкою фундаментальних технічних та методологічних обмежень:

*Обмеження моносенсорних систем:* використання виключно оптичних сенсорів (камер видимого діапазону) є недостатнім через проблеми маскування (камуфляж), зміни умов освітлення, та наявність перешкод (висока трава, листя). Водночас, класичні магнітометричні методи, ефективні для виявлення великих металевих об'єктів, часто дають хибні спрацювання на безпечні металеві фрагменти або не реагують на пластикові міни з мінімальним вмістом металу (наприклад, ПФМ-1).

*Проблема хибних спрацювань (False Positives):* сучасні алгоритми комп'ютерного зору, зокрема детектори на базі CNN (Convolutional Neural Network), схильні до помилок першого роду на складному фоні, плутаючи з мінами побутове сміття, каміння або специфічні елементи рельєфу. Це призводить до надмірного навантаження на операторів, які змушені вручну перевіряти тисячі кадрів.

*Розрив у контурі керування:* більшість існуючих роботизованих систем функціонують у розімкненому циклі («open-loop»). Дані збираються пасивно,



постфактум передаються на сервер, і лише після тривалого аналізу приймається рішення про необхідність повторного вильоту. Відсутність автоматичної реакції на виявлену підозру в реальному часі значно знижує темпи обстеження територій.

*Ресурсні обмеження:* високоточні нейромережеві моделі вимагають значних обчислювальних потужностей, які проблематично розмістити на борту БПЛА через обмеження енергоспоживання та вантажопідйомності (SWaP – Size, Weight, and Power). Враховуючи вищезазначене, розробка багаторівневої автоматизованої системи, яка інтегрує обчислення на периферії (Edge), наземній станції (Ground) та у хмарі (Cloud), поєднує різноманітні сенсори (мультимодальність) та використовує новітні візуально-мовні моделі (VLM) для семантичної верифікації, є актуальним науково-технічним завданням. Така система дозволяє створити замкнений цикл керування, де робот виступає не пасивним сенсором, а активним агентом, здатним адаптувати свою місію залежно від контексту.

**Мета і завдання дослідження.** Метою дослідження є підвищення ефективності, швидкості та безпеки процесів виявлення вибухонебезпечних предметів за допомогою БПЛА шляхом створення багаторівневої автоматизованої системи керування, що забезпечує мультимодальне злиття даних, семантичну верифікацію об'єктів та адаптивну корекцію траєкторії польоту в режимі реального часу.

Для досягнення поставленої мети вирішувалися наступні завдання:

- Провести системний аналіз методів дистанційного зондування та існуючих архітектур керування роботизованими платформами для визначення їхніх недоліків та «вузьких місць» у задачах розмінування.
- Розробити та обґрунтувати багаторівневу архітектуру (Edge–Ground–Cloud), яка дозволяє збалансувати навантаження між бортовим обчислювачем, наземною станцією та хмарними сервісами, забезпечуючи мінімальну затримку критичних рішень.

- Створити математичну модель мультимодального злиття даних (Weighted Confidence Aggregation), що об'єднує виходи візуальних детекторів, семантичних моделей та фізичних сенсорів (металодетектора) в єдину інтегральну метрику ймовірності загрози.
- Розробити алгоритми та програмне забезпечення для замкненого контуру керування, яке реалізує автоматичний перехід між режимами польоту та виконання маневрів детального обстеження.
- Дослідити ефективність застосування візуально-мовних моделей (VLM) як семантичних фільтрів для зменшення кількості хибних спрацювань та порівняти їхню точність з традиційними методами.
- Здійснити практичну реалізацію прототипу системи на базі БПЛА гексакоптерного типу та провести польові експериментальні дослідження для верифікації запропонованих моделей та алгоритмів.

**Об'єктом дослідження** є процеси автоматизованого керування рухом та інформаційною взаємодією у складі роботизованої платформи під час виконання завдань просторового моніторингу та пошуку об'єктів на місцевості.

**Предметом дослідження** є методи, моделі та алгоритми багаторівневої обробки сенсорних даних, мультимодального злиття інформації (sensor fusion), семантичної верифікації образів та адаптивного керування траєкторією роботизованої платформи у вигляді БПЛА при умовах невизначеності зовнішнього середовища.

**Методи дослідження.** Для розв'язання поставлених задач у роботі застосовано комплексний методологічний підхід, що базується на поєднанні теоретичних досліджень з експериментальною верифікацією отриманих результатів. Зокрема, для декомпозиції задачі керування складною робототехнічною системою та проєктування розподіленої архітектури Edge-Cloud використано методи системного аналізу. Синтез законів керування рухом БПЛА, стабілізація висоти за даними лідара та реалізація логіки перемикання режимів польоту здійснені на основі методів теорії

автоматичного керування. Побудова та навчання згорткових нейронних мереж (YOLOv8) для детектування об'єктів у відеопотоці виконані з використанням методів комп'ютерного зору (Computer Vision) та глибокого навчання (Deep Learning). Для інтеграції візуально-мовних моделей (VLM) та забезпечення семантичного аналізу сцен («open-vocabulary recognition») застосовано методи обробки природної мови та мультимодального навчання. Розробка математичної моделі зваженої агрегації довіри, а також калібрування виходів нейромереж базуються на методах теорії ймовірностей та прийняття рішень. Перевірка адекватності запропонованих моделей, оцінка точності та часових характеристик системи проведена за допомогою експериментальних методів під час випробувань.

### **Наукова новизна отриманих результатів:**

Наукова новизна дисертаційної роботи полягає у вирішенні важливої науково-технічної задачі автоматизації пошукових місій БПЛА шляхом створення інтегрованої системи керування, що базується на нових принципах розподіленої обробки та семантичного аналізу.

Основні наукові результати, що виносяться на захист:

- *Вперше розроблено багаторівневу розподілену архітектуру системи керування роботизованою платформою просторового моніторингу, яка, на відміну від традиційних централізованих підходів, реалізує принцип паралельної асинхронної обробки інформаційних потоків у контурі «Edge–Ground–Cloud». Запропоноване рішення забезпечує динамічну оркестрацію обчислювальних процесів, де критичні для безпеки польоту задачі (навігація, уникнення перешкод, первинна детекція) виконуються синхронно на бортовому обчислювачі (Edge) з детермінованою латентністю, тоді як ресурсомісткі задачі семантичної верифікації та глобального планування місії виконуються асинхронно на наземному та хмарному рівнях. Це дозволяє усунути затримки, характерні для послідовної ескалації, забезпечуючи безперервність*

моніторингу та адаптивну поведінку платформи як активного агента в умовах обмеженої пропускної здатності каналів зв'язку.

- *Набуло подальшого розвитку* застосування методу автоматичного адаптивного керування польотом БПЛА, який передбачає організацію замкненого циклу детекції, семантичної верифікації та відповідної корекції місії з використанням протоколу MAVLink. Використання цього методу дозволяє трансформувати пасивний моніторинг у активний процес збору даних, виключити людський фактор з рутинних операцій та суттєво зменшити час реакції системи на виявлені загрози.
- *Набула подальшого розвитку* математична модель прийняття рішень в умовах невизначеності, яка базується на методі зваженої агрегації довіри (Weighted Confidence Aggregation) для мультимодального злиття гетерогенних даних (оптичний потік, семантичні дескриптори VLM, сигнали металодетектора). Відмінною рисою моделі є введення механізму адаптивної корекції вагових коефіцієнтів, що функціонує як апроксимація рекурсивного байєсівського оцінювання. Це дозволяє системі автоматично зміщувати пріоритет довіри між візуальним та фізичним каналами сприйняття залежно від оцінки якості середовища (рівень освітленості, наявність рослинності, електромагнітні завади та ін.), забезпечуючи робастність ідентифікації та мінімізацію помилок першого і другого роду (хибних тривог та пропусків цілей) у складному фоновому оточенні.

**Практичне застосування отриманих результатів.** Результати дисертаційного дослідження доведені до рівня практичної реалізації у вигляді діючого програмно-апаратного комплексу. Дисертаційна робота була використана у наукових дослідках Національного університету кораблебудування імені адмірала Макарова в рамках держбюджетної теми «Новітні системи дистанційного керування мобільними роботами для відновлення суден і портової інфраструктури» (державний реєстраційний номер 0126U001721), а також впроваджена у навчальний процес цього ж

університету для підготовки фахівців за спеціальністю 151 «Автоматизація та комп'ютерно-інтегровані технології». Крім того, розроблені багаторівневі системи керування мобільними платформами, алгоритми обробки візуальних даних та методи комплексування сенсорних даних впроваджено у науково-виробничий процес МП «ОРБІТА» при створенні систем дистанційного моніторингу, а також у практичну діяльність ТОВ «ХЛАДОТЕХНИКА», що підтверджено відповідними актами впровадження.

### **Особистий внесок здобувача.**

Основний зміст роботи, всі теоретичні та практичні результати, висновки і дослідження, що представлено до захисту, одержані автором самостійно. Особисто здобувачеві належать наступні наукові результати:

- у роботі [56] здобувачем створено систему автоматичного керування місіями гексакоптера із замкненим циклом зворотного зв'язку. Реалізовано алгоритм взаємодії через протокол MAVLink, який на основі об'єднаної оцінки загроз автоматично переводить апарат з режиму AUTO (автоматичне виконання місії) в режим GUIDED (керований політ), скеровує його для детального дообстеження локації металодетектором та самостійно відновлює перерваний маршрут;
- у роботі [45] здобувачем розроблено математичну модель мультимодального злиття даних, яка об'єднує результати візуального детектора YOLOv8, візуально-мовних моделей та металодетектора в єдину інтегральну ймовірність. Кожна модальність калібрується та агрегується через вагові коефіцієнти або логічні правила для балансування ймовірності виявлення та кількості хибних тривог;
- у роботі [35] здобувачем запропоновано багаторівневу архітектуру виявлення мін (Edge–Ground–Cloud). Створено просторово-часовий конвеєр, де швидка візуальна детекція та геоприв'язка виконуються на наземній станції, а семантична верифікація спірних випадків делегується візуально-мовним моделям у хмарі, що знижує хибні спрацьовування без суттєвого впливу на затримку.

**Апробація результатів.** Основні положення та результати дисертаційного дослідження пройшли апробацію на міжнародних та всеукраїнських науково-практичних конференціях. Зокрема, результати доповідалися та обговорювалися на конференціях, матеріали яких індексуються у наукометричній базі Scopus: *5th International Workshop on Modern Machine Learning Technologies and Data Science* з доповіддю «Machine Learning and Modeling of the Impact of Trademark Filings on GDP Growth based on Python», *International Conference on Advanced Computer Information Technologies* (ACIT 2024) з доповіддю «Enabling Cloud-based Data Analysis for Analog Metal Detectors using Microcomputer Systems» та *ACIT 2025* з доповіддю «AI-Driven OCR in Practice: Assessing ChatGPT-4.1 and Google Vision as Alternatives to Manual Optical Character Recognition». Крім того, окремі аспекти роботи оприлюднено на XV Міжнародній науково-технічній конференції «Інновації в суднобудуванні та океанотехніці» (2024 р.), XXVIII Всеукраїнській науково-практичній конференції «Могилянські читання – 2025» (тема доповіді: «Виявлення вибухонебезпечних предметів за допомогою БПЛА та візуально-мовних моделей»), Всеукраїнській науково-технічній конференції студентів, аспірантів та молодих вчених (2024 р.) за темою «Багаторівнева хмарна обробка відео машинного зору», а також на I Міжнародній науково-практичній інтернет-конференції «Achievements of 21st Century Scientific Community» (2023 р.) з доповіддю «Хмарна обробка GPS-даних на базі Raspberry Pi з польотного контролера дрона».

## **РОЗДІЛ 1. СУЧАСНИЙ СТАН ПРОБЛЕМИ АВТОМАТИЗОВАНОГО КЕРУВАННЯ РОБОТИЗОВАНИМИ ПЛАТФОРМАМИ ДЛЯ ЗАДАЧ ПРОСТОРОВОГО МОНІТОРИНГУ НАВКОЛИШНЬОГО СЕРЕДОВЩА**

### **1.1 Огляд автоматичних систем керування роботизованих платформ**

Автоматична система керування (АСК) є центральним елементом архітектури будь-якої роботизованої платформи і безпілотний апарат не є виключенням, він забезпечує перетворення високорівневих завдань місії у точні сигнали для виконавчих механізмів, гарантуючи стабільність польоту та точність навігації. Сучасні АСК є складними кіберфізичними системами, що інтегрують математичні методи теорії автоматичного керування, алгоритми інерціальної навігації та мережеві протоколи обміну даними. Нижче наведено детальний аналіз компонентів та принципів роботи таких систем.

#### **1.1.1 Архітектура систем керування та навігації**

В основі функціонування будь-якого автономного робота лежить концепція GNC (Guidance, Navigation, and Control – Наведення, Навігація та Керування), яка декомпозує задачу керування на три взаємопов'язані підсистеми [1]. Підсистема Навігації (Navigation) відповідає за оцінку поточного стану апарата у просторі та часі. Вона формує вектор стану, який зазвичай включає географічні координати (широта, довгота, висота), лінійні та кутові швидкості, а також орієнтацію апарата (кути Ейлера або кватерніони) [2]. Оскільки дані з окремих сенсорів завжди містять шуми та похибки (наприклад, дрейф гіроскопів або багатопроменеве поширення сигналу GPS), ядром підсистеми навігації є алгоритм сенсорної фузії (sensor fusion). Найпоширенішим методом є Розширений Фільтр Калмана (Extended Kalman Filter, EKF), який статистично оптимально об'єднує високочастотні дані інерціальної системи (IMU) з низькочастотними абсолютними вимірюваннями від GPS, магнітометра, барометра та інших джерел (наприклад, оптичного

поток). EKF прогнозує стан системи на основі динамічної моделі руху та коригує цей прогноз при надходженні нових вимірювань, оцінюючи також похибки (bias) самих сенсорів. Підсистема Наведення (Guidance), яку часто називають «зовнішнім контуром» (outer loop), відповідає за формування бажаної траєкторії руху для досягнення цілей місії [3]. На вхід цього блоку надходять маршрутні точки (waypoints) або команди від системи планування шляху. На основі поточного навігаційного стану та цільової точки алгоритми наведення (наприклад, L1 Navigation для літаків або P-регулятор позиції для коптерів) розраховують необхідні кінематичні параметри руху: бажаний кут крену для повороту, необхідну вертикальну швидкість для зміни висоти або цільовий вектор швидкості. У системах з нерухомим крилом часто використовується алгоритм TECS (Total Energy Control System), який координує роботу двигуна та руля висоти для керування повною енергією літака (кінетичною та потенціальною), забезпечуючи ефективний набір висоти та підтримку швидкості. Підсистема Керування (Control), або «внутрішній контур» (inner loop), забезпечує безпосередню стабілізацію апарата та відпрацювання команд (рис. 1.1), отриманих від підсистеми наведення.

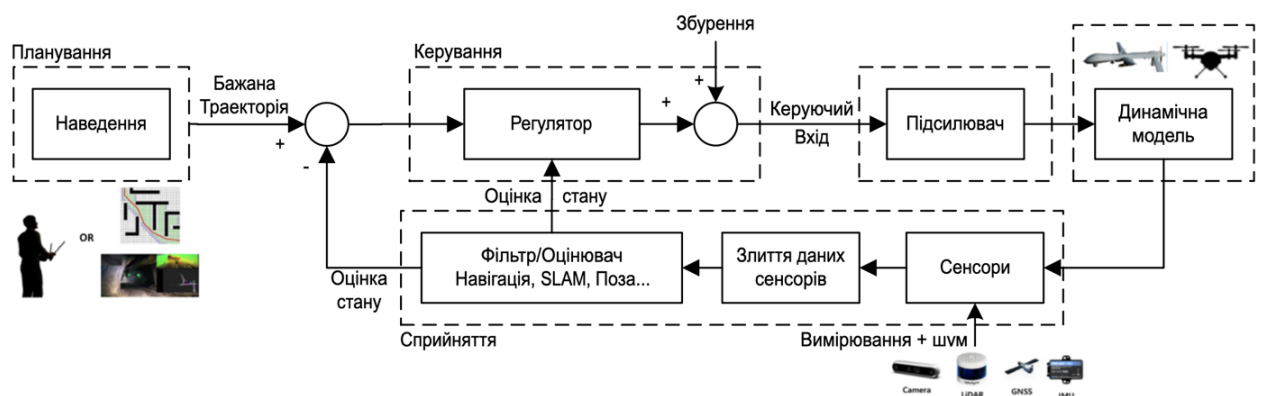


Рисунок 1.1 – Узагальнена блок-схема системи керування БПЛА

Її завданням є мінімізація помилки між бажаними (commanded) та виміряними (measured) кутовими положеннями та швидкостями. Архітектурно це часто реалізується у вигляді каскадних PID-регуляторів:



регулятор кута формує завдання для регулятора кутової швидкості (rate controller), вихід якого, у свою чергу, перетворюється мікшером (mixer) у сигнали ШІМ (PWM) для конкретних моторів та сервоприводів. Внутрішній контур працює з найвищою частотою (400 Гц – 1 кГц і вище), забезпечуючи миттєву реакцію на турбулентність та інші зовнішні збурення. Блок-схеми GNC систем наочно демонструють потік інформації: від «сирих» сенсорних даних через блок оцінки стану (State Estimation) до генератора траєкторії (Guidance) та стабілізатора (Control), замикаючись через динаміку об'єкта керування.

### **1.1.2 Програмно-апаратна реалізація польотних контролерів**

Теоретичні принципи GNC знаходять своє втілення у сучасних польотних контролерах (Flight Controllers), серед яких екосистема Pixhawk та програмне забезпечення ArduPilot є стандартом для професійних та дослідницьких застосувань. Апаратна архітектура Pixhawk (серія FMU – Flight Management Unit) базується на використанні потужних 32-бітних мікроконтролерів архітектури ARM Cortex-M4 або M7 (наприклад, STM32F4/F7/H7), які мають апаратну підтримку операцій з плаваючою комою (FPU). Ключовою особливістю дизайну Pixhawk є висока надійність та відмовостійкість. Це досягається завдяки використанню ізольованого блоку інерціальних сенсорів з механічною вібророзв'язкою та температурною стабілізацією. Сенсори часто дублюються або навіть потроюються (наприклад, три акселерометри та три гіроскопи різних виробників), що дозволяє застосовувати алгоритми голосування (voting logic) для виключення несправних датчиків у польоті [4]. Крім основного процесора, в архітектурі часто присутній окремий процесор введення-виведення (IO Processor), який відповідає за генерацію сигналів для моторів та обробку сигналів радіокерування (RC Input). У випадку програмного збою або перезавантаження основного процесора (FMU), IO-процесор переходить у

режим failsafe, дозволяючи зберегти базову керованість апаратом у ручному режимі (manual override). Програмний стек ArduPilot є складною операційною системою реального часу (RTOS), побудованою на модульній архітектурі. Фундаментом системи є рівень абстракції обладнання (Hardware Abstraction Layer – HAL), який надає уніфікований інтерфейс до апаратних ресурсів (GPIO, UART, SPI, I2C), дозволяючи переносити код ArduPilot на різні плати та операційні системи (ChibiOS, Linux) без зміни основної логіки. Вище знаходяться предметно-орієнтовані бібліотеки, що реалізують алгоритми EKF, PID-регулювання, навігації та керування периферією. Центром програмного забезпечення є планувальник (scheduler), який забезпечує детерміноване виконання задач. Критичні задачі, такі як зчитування гіроскопів та розрахунок контуру стабілізації, виконуються з найвищим пріоритетом та строгою періодичністю (наприклад, кожні 2.5 мс для 400 Гц), тоді як менш пріоритетні задачі (обробка повідомлень GCS, запис логів на SD-карту, перевірка батареї) виконуються у вільний час процесора. Така архітектура гарантує передбачуваність поведінки системи та мінімізацію Джитера (jitter) у контурах керування.

### **1.1.3 Протоколи взаємодії та маршрутизація повідомлень (MAVLink)**

Для забезпечення комунікації між польотним контролером, наземною станцією керування (Ground Control Station, GCS) та іншими бортовими пристроями використовується протокол MAVLink (Micro Air Vehicle Link) [5]. Це бінарний протокол телеметрії (рис 1.2), спеціально розроблений для систем з обмеженими обчислювальними ресурсами та каналами зв'язку з низькою пропускнуою здатністю.

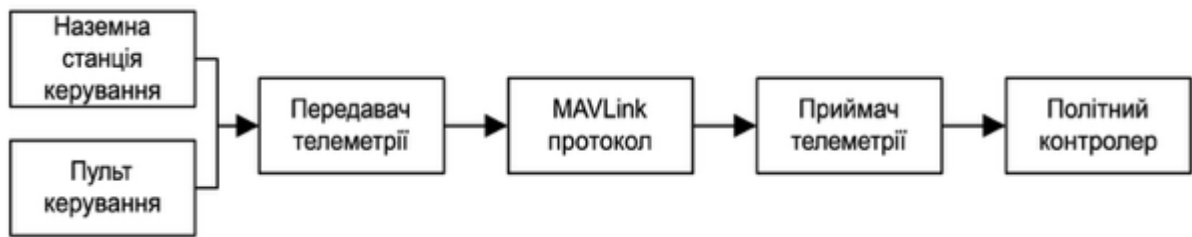


Рисунок 1.2 – Узагальнене представлення логічного взаємозв'язку компонентів архітектури керування польотом БПЛА за допомогою MAVLink

Структура пакету MAVLink максимально оптимізована: він містить компактний заголовок (header), що включає магічний байт початку пакету, довжину корисного навантаження, порядковий номер пакету (для виявлення втрат), ідентифікатори системи (System ID) та компонента (Component ID), ідентифікатор повідомлення (Message ID), власне дані (Payload) та контрольну суму (CRC) для перевірки цілісності. Протокол підтримує два основні режими передачі даних: потоковий (streaming) та надійний (reliable). Потоковий режим використовується для передачі телеметрії (положення, висота, заряд батареї), де втрата одного пакету не є критичною, оскільки наступний пакет прийде через короткий проміжок часу. Надійний режим використовується для критичних операцій, таких як завантаження місії, зміна параметрів налаштування або відправка команд керування. У цьому режимі реалізовано механізм підтвердження отримання (ACK) та автоматичної повторної відправки (retransmission) у разі втрати пакету. Мережева архітектура MAVLink базується на системі адресації, де кожен вузол (автопілот, камера, GCS, бортовий комп'ютер) має унікальну пару System ID / Component ID. Це дозволяє будувати складні топології мережі. Польотний контролер виконує функцію маршрутизатора (router): отримавши пакет, він перевіряє адресу призначення. Якщо пакет адресовано йому, він обробляється; якщо ні – контролер пересилає його на інші доступні інтерфейси (UART, USB, CAN) відповідно до внутрішньої таблиці маршрутизації. Це забезпечує прозору комунікацію («тунелювання») між наземною станцією та периферійними пристроями на борту дрона (наприклад, налаштування камери через

радіоканал автопілота). MAVLink також визначає набір стандартних мікросервісів (microservices) – протоколів вищого рівня, що описують діаграми послідовності обміну повідомленнями для складних операцій, таких як передача файлів (FTP), обмін місіями (Mission Protocol) та калібрування сенсорів.

#### 1.1.4 Взаємодія бортового комп'ютера та автопілота

Оскільки мікроконтролер автопілота оптимізований для задач жорсткого реального часу і має обмежені ресурси пам'яті та обчислювальної потужності, для реалізації інтелектуальних функцій (комп'ютерний зір, планування складних місій, SLAM) в архітектуру (рис 1.3) вводиться додатковий бортовий комп'ютер (Companion Computer).

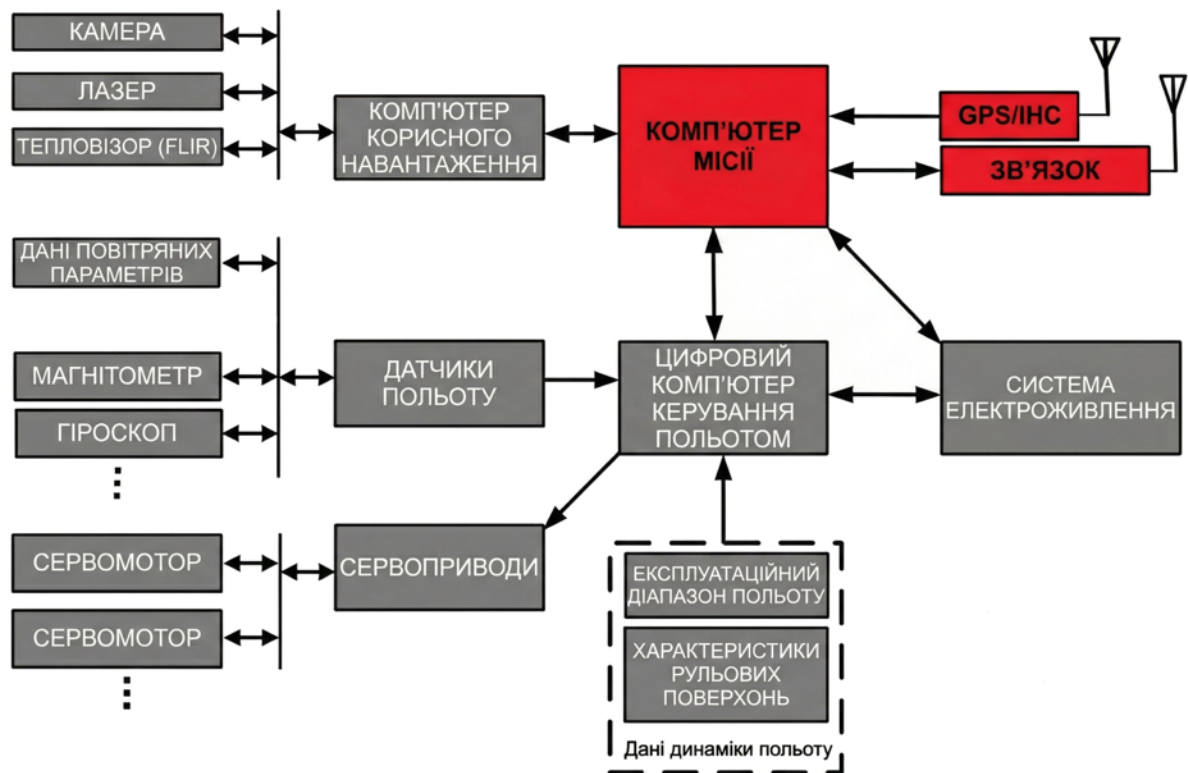


Рисунок 1.3 – Схема інтеграції додаткового бортового комп'ютера

Це зазвичай одноплатний комп'ютер під керуванням Unix-подібної операційної системи (наприклад, Raspberry Pi, NVIDIA Jetson, Odroid), який

з'єднується з польотним контролером через високошвидкісний інтерфейс UART (Telemetry Port), USB або Ethernet. Взаємодія будується за схемою «ведучий-ведений», де бортовий комп'ютер виступає як інтелектуальна надбудова над автопілотом. Він отримує від польотного контролера потік телеметрії (стан, координати, дані сенсорів) через MAVLink, виконує ресурсомістку обробку даних (наприклад, детекцію об'єктів на відео) і генерує керуючі команди. Для безпосереднього керування дроном використовуються спеціальні режими польоту: GUIDED (в ArduPilot) або OFFBOARD (в PX4). У цих режимах автопілот ігнорує команди від пульта радіокерування (або використовує їх лише як коригуючі) і очікує від бортового комп'ютера регулярного потоку цільових точок (setpoints). Це можуть бути цільові координати (Position Setpoints), вектори швидкості (Velocity Setpoints) або навіть бажані кути орієнтації та тяга (Attitude/Thrust Setpoints). Для спрощення розробки програмного забезпечення на бортовому комп'ютері використовуються бібліотеки-посередники, які абстрагують складність протоколу MAVLink. Найпопулярнішими є MAVSDK, DroneKit та MAVROS (вузол-міст для інтеграції з Robot Operating System). Блок-схеми таких систем ілюструють замкнений контур керування вищого рівня: камера фіксує сцену > алгоритм CV на бортовому комп'ютері локалізує ціль > навігаційний модуль розраховує необхідне зміщення > через API (наприклад, MAVROS) відправляється команда SET\_POSITION\_TARGET\_LOCAL\_NED > автопілот отримує команду через UART і відпрацьовує її через свої внутрішні PID-регулятори [6]. Важливим елементом надійності такої схеми є механізм застосування протоколів безпеки (watchdog): автопілот контролює частоту надходження команд від бортового комп'ютера, і якщо потік переривається (зависання комп'ютера, обрив зв'язку), автоматично перемикається в безпечний режим (наприклад, зависання на місці або повернення додому), запобігаючи втраті апарата.

## **1.2 Огляд існуючих апаратних рішень для моніторингу та розмінування.**

Аналіз сучасного стану розвитку технічних засобів гуманітарного розмінування свідчить про стійку тенденцію до заміни традиційних контактних методів дистанційними роботизованими комплексами, оснащеними різноманітними сенсорними системами. Еволюція апаратного забезпечення у цій сфері спрямована на вирішення критичної дилеми між продуктивністю обстеження територій та точністю ідентифікації вибухонебезпечних предметів (ВНП), що в умовах реальних бойових дій та повоєнного відновлення набуває особливого значення [7]. Сучасні дослідження фокусуються на розробці спеціалізованих безпілотних авіаційних (БПЛА) та наземних (UGV) платформ, які інтегрують передові методи сенсорики з алгоритмами штучного інтелекту для підвищення ймовірності виявлення загроз [8].

Вагомий внесок у розвиток аеромагнітних систем розвідки зроблено у роботах, присвячених створенню малогабаритних БПЛА-магнітометрів. Показовим прикладом такого підходу є розробка платформи «Maggy», [9] яка позиціонується як економічно ефективне рішення для швидкого сканування забруднених територій. Апаратна архітектура комплексу базується на легкій мультироторній платформі, оснащій системою високоточного утримання висоти за допомогою радарного альтиметра, що дозволяє виконувати польоти в режимі огинання рельєфу на надмалих висотах (близько 0,5 м). Ключовим елементом сенсорного навантаження виступають інтегровані феррозондові магнітометри (fluxgate sensors) типу HWT3100, які забезпечують вимірювання компонентів вектора магнітної індукції з високою частотою дискретизації. Експериментальні дослідження підтвердили високу ефективність такої конфігурації для виявлення феромагнітних об'єктів: при дотриманні оптимальних параметрів польоту (швидкість 1 м/с та висота 0,5 м) система продемонструвала точність виявлення на рівні 96% для мін із металевим корпусом. Водночас, критичною особливістю подібних рішень є їхня вузька

спеціалізація та залежність від стабільності польоту, оскільки збільшення висоти сканування навіть до 1 метра призводить до різкого падіння амплітуди корисного сигналу, що унеможливорює надійну ідентифікацію малорозмірних цілей.

Альтернативний напрям дистанційного зондування базується на використанні тепловізійних систем, які дозволяють виявляти ВНП за рахунок ефекту теплової інерції – різниці температур між корпусом міни та навколишнім ґрунтом. Сучасні дослідження, зокрема роботи [10] Neuschmid et al., демонструють перехід від використання важких георадарних комплексів (GPR) до легших БПЛА, оснащених високочутливими інфрачервоними камерами (наприклад, Zenmuse XT). Для автоматизації аналізу термограм застосовуються методи глибокого навчання, оптимізовані для роботи на бортових обчислювачах з обмеженими ресурсами. Зокрема, використання архітектури MobileNetV3-Large дозволило досягти вражаючих показників точності класифікації (Test Accuracy 96,14%) на валідаційних наборах даних, що містять зображення мін на поверхні та на невеликій глибині. Перевагою такого підходу є можливість швидкого обстеження великих площ без необхідності складного постпроцесингу, характерного для георадарів. Проте ефективність методу залишається залежною від часу доби та метеорологічних умов, а фізичні обмеження тепловізійного методу не дозволяють надійно виявляти об'єкти на значній глибині або під густою рослинністю.

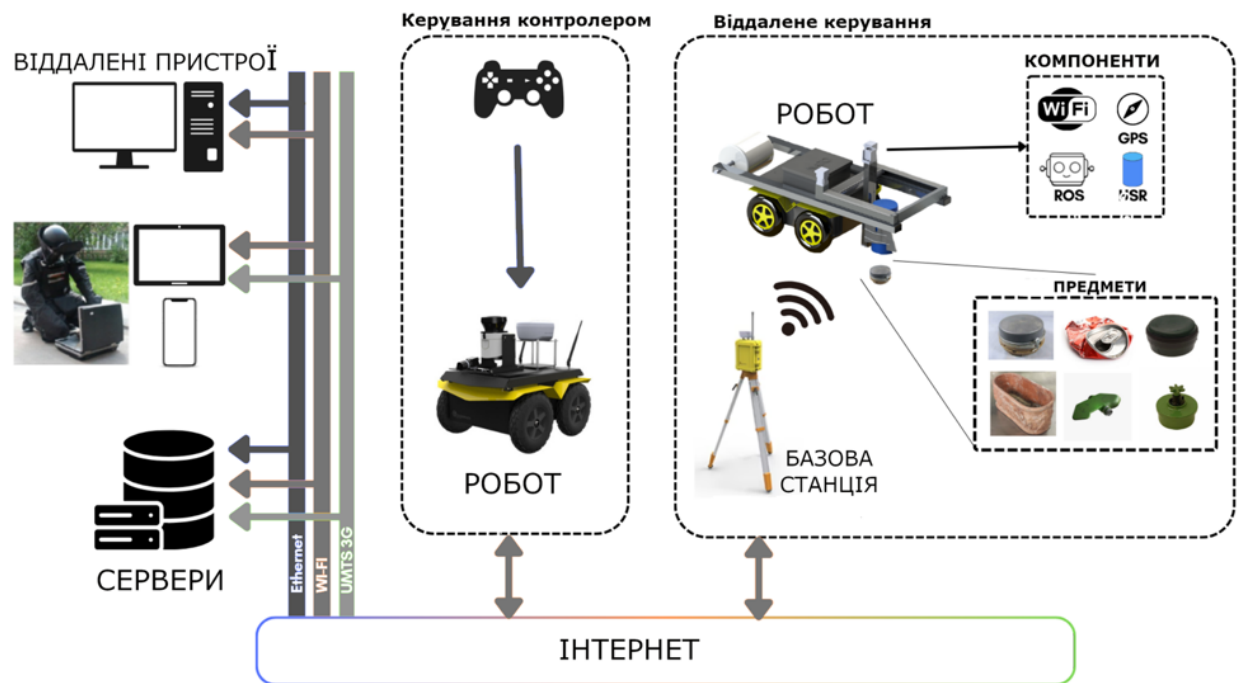


Рисунок 1.4 – Схема архітектури наземних роботизованих платформ для ідентифікації предметів

Окрім повітряних засобів, значного розвитку набули наземні роботизовані платформи, які забезпечують верифікацію загроз із близької відстані. Концепція «UGO-first», запропонована Vivoli та співавторами, реалізує парадигму Industry 4.0 у задачах розмінування [11], інтегруючи оптичні сенсори, лідари та системи точного позиціонування (RTK-GPS) у єдину мережеву структуру. Особливістю архітектури є розподілена модель обробки даних: первинний відеопотік транслюється з борту робота, а інференс нейронних мереж (зокрема моделей сімейства YOLOv8) виконується на стороні клієнтського пристрою оператора через веб-інтерфейс із використанням формату ONNX. Такий підхід дозволяє знизити енергоспоживання бортового обладнання та забезпечити гнучкість розгортання системи. В експериментах система продемонструвала високу надійність (recall) при детектуванні поверхневих мін типів ПФМ-1 та ПМА-2, проте її застосування обмежене прохідністю наземної платформи та швидкістю обробки відеопотоку на мобільних пристроях, що становить близько 2–3 кадрів на секунду.



З метою подолання обмежень окремих сенсорів, у науковій літературі, зокрема у працях [12] Stankevich & Saprykin, обґрунтовується необхідність комплексування різнорідних даних. Запропонована методологія передбачає концепцію «роздільного картування» (mapping separately), коли збір оптичної та магнітометричної інформації виконується різними носіями з відмінними польотними завданнями. Інтеграція результатів відбувається на етапі постобробки шляхом формування ймовірнісних карт місцевості, де дані від кожного сенсора об'єднуються з використанням байєсівського правила злиття ймовірностей. Такий підхід дозволяє математично коректно врахувати рівень довіри до кожного джерела інформації та підвищити результуючу ймовірність виявлення до значень понад 0,99 у складних сценаріях, де моносенсорні методи дають неоднозначні результати. Втім, асинхронність отримання даних та необхідність точного просторового суміщення карт від різних носіїв ускладнюють оперативне прийняття рішень безпосередньо під час виконання місії.

Узагальнюючи огляд існуючих апаратних рішень, можна констатувати, що сучасні системи розвиваються у напрямку спеціалізації та підвищення автономності обробки даних. Водночас, більшість розглянутих підходів реалізують або пасивний збір інформації з подальшою офлайн-обробкою, або працюють у розімкненому контурі керування, де результати детектування не впливають миттєво на поведінку платформи. Це створює передумови для розробки нових інтегрованих систем, здатних забезпечити адаптивне керування рухом носія та мультимодальне злиття даних у режимі реального часу, що є предметом дослідження даної дисертаційної роботи.

### **1.3 Розвиток методів комп'ютерного зору**

Сучасні системи комп'ютерного зору відіграють ключову роль у розвитку автоматизованого моніторингу, робототехніки та безпілотних платформ. Зростання кількості сенсорів, збільшення роздільної здатності камер і доступність безпілотних літальних апаратів привели до того, що обсяг

даних, який необхідно обробляти в реальному часі, постійно зростає. У таких умовах вирішальне значення набувають методи автоматичного розпізнавання об'єктів, здатні не лише замінювати ручний візуальний перегляд, а й інтегруватися в замкнені контури керування технічними системами. Помилка оператора або затримка в аналізі кадрів може призвести до втрати важливих подій, тому перехід від переважно ручного аналізу до автоматизованих конвеєрів комп'ютерного зору розглядається як базова передумова побудови сучасних систем просторового моніторингу.

На практиці задачі розпізнавання об'єктів охоплюють широкий спектр сценаріїв. Це виявлення вибухонебезпечних предметів і підозрілих об'єктів на місцевості, контроль цілісності критичної інфраструктури, виявлення пошкоджень дорожнього полотна або елементів інженерних споруд, пошук людей у зонах надзвичайних ситуацій. Спільною рисою таких задач є складні умови роботи. Об'єкти можуть мати малий розмір на зображенні, частково перекриватися елементами фону, маскуватися рельєфом або рослинністю, потрапляти в кадр під різними кутами і на різних відстанях. До цього додаються зміни освітлення, атмосферні завади, вібрації платформи та обмеження на обчислювальні ресурси, характерні для бортових обчислювальних модулів. У результаті просте застосування традиційних алгоритмів обробки зображень часто виявляється недостатнім для стабільної роботи системи в реальних умовах експлуатації.

Розвиток глибинного навчання радикально змінив підходи до розв'язання задач комп'ютерного зору. Поява згорткових нейронних мереж дала змогу перейти від ручного конструювання ознак до автоматичного навчання ієрархічних представлень безпосередньо з даних. Це привело до створення цілого класу архітектур для детектування об'єктів, серед яких особливу роль відіграють моделі сімейства YOLO, орієнтовані на роботу в режимі реального часу. Використання таких детекторів у складі бортових систем безпілотних літальних апаратів дало можливість реалізувати автоматичне виявлення невеликих наземних об'єктів на відеопотоці з повітря

і таким чином істотно зменшити навантаження на оператора. Практичні експерименти показали, що правильно налаштовані моделі цієї групи здатні забезпечувати прийнятний компроміс між точністю, швидкістю та вимогами до апаратного забезпечення навіть на відносно малопотужних обчислювальних платформах. Подальший розвиток систем комп'ютерного зору пов'язаний з переходом від ізольованих детекторів до багаторівневих архітектур аналізу даних. У таких архітектурах попередній аналіз і фільтрація кадрів виконуються на борту безпілотної платформи або на ближчому до сенсорів рівні, а більш ресурсомісткі операції передаються на наземні чи хмарні обчислювальні вузли. Це дає змогу поєднати переваги низької затримки та високої частоти оновлення на Edge рівні з гнучкістю складніших моделей, які можуть працювати лише з невеликою частиною найбільш інформативних даних. Для задач моніторингу небезпечних і критично важливих об'єктів такий підхід забезпечує можливість вибіркової деталізації аналізу саме в тих зонах, де первинні детектори фіксують потенційно значущі події. Окремим етапом еволюції стали системи, які використовують не лише дані зображень, а й інформацію від додаткових сенсорів. Поєднання відеопотоку з даними металодетекторів, тепловізорів, магнітометрів або інших спеціалізованих датчиків дозволяє компенсувати слабкі сторони окремих каналів і підвищити загальну надійність виявлення. У таких системах ключову роль відіграє коректне узгодження просторових і часових характеристик різних сенсорів, а також побудова інтегральних показників впевненості, які відображають спільний результат аналізу. Це переводить задачу розпізнавання об'єктів у площину мультимодальної обробки та злиття даних, що природно поєднується з вимогами до автоматизації прийняття рішень у реальному часі.

Новим напрямом розвитку комп'ютерного зору є використання візуально мовних моделей (VLM), які поєднують аналіз зображень і природної мови. Такі моделі здатні інтерпретувати сцену за текстовим запитом, пояснювати рішення та працювати з відкритими множинами класів. У контексті задач автоматизованого моніторингу їх доцільно розглядати як

високорівневі семантичні фільтри над результатами класичних детекторів. Базові моделі виявляють підозрілі області, а візуально мовні моделі уточнюють їхню інтерпретацію, перевіряють гіпотези та зменшують кількість хибних спрацювань. Попри значні обчислювальні витрати і поки що обмежену автономність таких моделей, їх інтеграція в багаторівневі архітектури створює передумови для побудови систем, здатних не лише знаходити об'єкти, а й узгоджувати ці знахідки із задачами оператора природною мовою.

Отже, сучасний стан розвитку комп'ютерного зору характеризується переходом від окремих алгоритмів обробки зображень до комплексних автоматизованих систем, у яких детекція об'єктів, мультимодальне злиття даних та високорівнева семантична інтерпретація об'єднані в єдиний технологічний ланцюг. Такі системи орієнтовані на роботу в умовах великих обсягів даних, обмежених ресурсів і високих вимог до надійності. Подальші підрозділи цього розділу присвячені детальнішому аналізу історичного розвитку методів комп'ютерного зору, сучасних архітектур детекторів і тенденцій інтеграції комп'ютерного зору в багаторівневі автоматизовані комплекси просторового моніторингу.

Комп'ютерний зір як самостійна галузь науки виник у другій половині XX століття і пройшов шлях від простих процедур обробки зображень до сучасних складних моделей глибокого навчання. Початкові дослідження у сфері комп'ютерного зору були зосереджені переважно на класичних методах аналізу зображень, що базувалися на алгоритмах виділення ознак та евристичних правилах. Великим кроком у розвитку комп'ютерного зору стало впровадження штучних нейронних мереж. Перші нейронні мережі, такі як LeNet, розроблені Яном ЛеКуном (Yann LeCun) наприкінці 80-х років, дали змогу вирішувати прості задачі класифікації зображень з використанням автоматично навчених ознак [13]. Однак значущого прориву вдалося досягти лише у 2012 році завдяки архітектурі AlexNet, що перемогла у змаганні ImageNet, продемонструвавши істотно вищу точність порівняно з класичними

алгоритмами [14]. Ця подія ознаменувала початок ери глибокого навчання у комп'ютерному зорі.

Подальший розвиток CV був пов'язаний з удосконаленням згорткових нейронних мереж (CNN). Були запропоновані нові, більш складні архітектури, такі як VGGNet, GoogLeNet (Inception), ResNet, DenseNet, що дозволили ефективніше вирішувати задачі класифікації та локалізації об'єктів [14, 15]. Водночас виникли й спеціалізовані архітектури для задач детектування об'єктів – такі, як двостадійні моделі R-CNN, Fast R-CNN, Faster R-CNN та одностадійні – SSD (Single Shot Multibox Detector), YOLO (You Only Look Once), які забезпечили високу швидкість і точність роботи в реальному часі. Згодом набули широкого поширення підходи до задач семантичної та інстансної сегментації, що ґрунтуються на використанні нейронних мереж, зокрема моделей U-Net, Mask R-CNN, DeepLabv3 та більш сучасної моделі Segment Anything Model (SAM). Ці моделі дозволили вирішувати задачі виділення контурів і точних меж об'єктів, що важливо для застосувань у галузі медицини, автомобільного транспорту, робототехніки та безпілотних систем [14, 15].

Новітнім трендом у комп'ютерному зорі стала поява трансформерів, що спочатку активно використовувалися у задачах обробки природної мови, але згодом успішно адаптовані для аналізу візуальної інформації. Vision Transformer (ViT) – один із прикладів новітніх архітектур, що демонструє високі результати у класифікації та детектування зображень за рахунок уваги (self-attention), яка дозволяє ефективно аналізувати віддалені залежності у даних [14]. Ці моделі показали перспективність застосування трансформерів у сфері комп'ютерного зору завдяки своїй здатності ефективно навчатись на великих обсягах даних і демонструвати високу точність у складних задачах.

Подальшим етапом еволюції методів комп'ютерного зору стало формування візуально мовних моделей VLM які поєднують механізми трансформерів для обробки зображень і природної мови в єдиному архітектурному рішенні. Такі моделі навчаються на великих колекціях пар

«зображення – текст» і формують спільний простір ознак у якому візуальні та мовні представлення узгоджені між собою. Це дає змогу виконувати не лише традиційні задачі класифікації або детектування об'єктів, а й більш складні операції пов'язані з генерацією описів сцен, відповідями на запитання щодо зображення, пошуком за текстовими запитами та відкритим розпізнаванням нових класів без явного перенавчання під кожну окрему задачу. Поява візуально мовних моделей означила перехід від суто піксельного або ознакового аналізу до семантичної інтерпретації сцен де результати роботи детекторів можуть безпосередньо узгоджуватися з формулюваннями завдань оператора природною мовою.

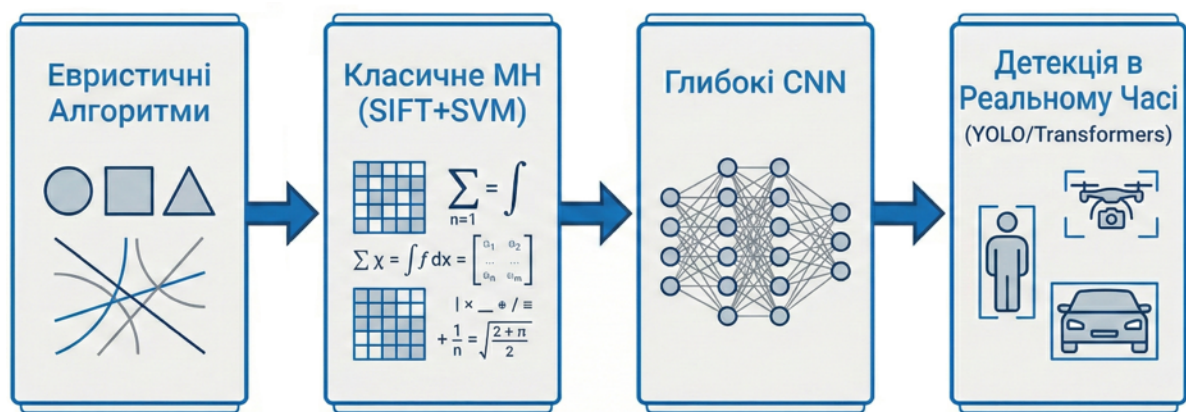


Рисунок 1.5 – Еволюція методів комп'ютерного зору

Таким чином, розвиток методів комп'ютерного зору (рис. 1.5) свідчить про постійний рух від простих евристичних алгоритмів до високопродуктивних моделей глибокого навчання, які здатні ефективно вирішувати комплексні задачі, зокрема й у такій специфічній галузі, як гуманітарне розмінування. Цей еволюційний шлях створює необхідне підґрунтя для подальших досліджень і визначає сучасні можливості та напрямки розвитку комп'ютерного зору.

## 1.4 Архітектурні рішення у комп'ютерному зорі

Сучасний розвиток комп'ютерного зору визначається не лише зростанням точності окремих моделей, а передусім переходом до комплексних автоматизованих систем, у яких детекція об'єктів інтегрується з багаторівневим керуванням і мультимодальною обробкою даних. Зростає кількість сценаріїв, де потрібно аналізувати великі обсяги відео з безпілотних літальних апаратів, оперативно виявляти невеликі та слабо помітні об'єкти і водночас працювати в умовах обмежених обчислювальних ресурсів та нестабільного каналу зв'язку. Це змушує розробників переходити від монолітних моделей до гнучких архітектур, які поєднують кілька рівнів обробки і різні типи моделей у єдиному конвеєрі. Одним з ключових трендів є поглиблення та ускладнення нейронних мереж. Сучасні згорткові архітектури та трансформери навчаються на масштабних датасетах і здатні опрацьовувати складні сцени з багатьма об'єктами, оклюзіями та різноманітними видами фону.



Рисунок 1.6 – Зображення роботи комп'ютерного зору

На цій основі активно розвиваються спеціалізовані моделі детектування, такі як сімейство YOLO (рис. 1.6), які забезпечують роботу в режимі близькому до реального часу і тому природно вписуються в системи, що працюють на борту безпілотників або на польових обчислювальних станціях. Паралельно зростає роль трансформерних архітектур і гібридних підходів, де поєднуються переваги згорткових мереж і механізмів уваги для точнішого аналізу глобального контексту зображення.

Наступна помітна тенденція пов'язана з адаптацією комп'ютерного зору до мобільних і вбудованих платформ. Для безпілотних систем, портативних пристроїв та IoT модулів критичними є обмеження на енергоспоживання, обсяг пам'яті та продуктивність, тому активно розробляються легковагі моделі, які зберігають прийнятний рівень точності, але потребують значно менше ресурсів. На практиці це проявляється у побудові багаторівневих конвеєрів, де на Edge [16] рівні працює швидкий детектор малої розмірності, а більш потужні моделі задіюються лише для частини даних, відібраних як найбільш підозрілі або інформативні. Окремий напрям сучасних трендів пов'язаний з переходом від суто візуальних моделей до мультимодальних рішень. У системах безпілотного моніторингу активно досліджуються схеми, де оптичний канал доповнюється сигналами металодетектора, тепловізора або іншими фізичними сенсорами, а результати злиття описуються через єдиний показник ризику. На цьому тлі з'являються візуально мовні моделі, що поєднують зображення та текст у спільному просторі ознак і можуть виконувати роль семантичного фільтра над попередніми детекторами. Їх застосування в багаторівневих Edge-ground-cloud архітектурах дозволяє уточнювати результати виявлення, зменшувати кількість хибних спрацювань і надавати оператору інтерпретовані пояснення, зберігаючи при цьому прийнятну затримку та навантаження на канал зв'язку.

Спільною рисою згаданих трендів є прагнення до керованого компромісу між точністю, швидкістю, надійністю та вартістю обчислень. Сучасні архітектури розглядають детектори, мультимодальні моделі та



візуально мовні компоненти не як ізольовані алгоритми, а як елементи єдиного багаторівневого середовища, у якому можна явно налаштовувати співвідношення між ймовірністю виявлення і рівнем хибних тривог за наявних обмежень на ресурси. У подальших підрозділах цього розділу розглядаються окремі групи архітектур та рішень, що відображають зазначені тенденції і є базою для побудови автоматизованих систем комп'ютерного зору на основі безпілотних літальних апаратів.

### 1.4.1 Збільшення глибини та складності нейронних мереж

Починаючи з архітектури AlexNet, яка містила вісім навчальних шарів і стала переломним моментом у змаганні ImageNet, глибина та структурна складність нейронних мереж для комп'ютерного зору невпинно зростають. Перехід від відносно неглибоких моделей до дуже глибоких мереж дав змогу суттєво поліпшити точність класифікації та детектування, оскільки кожний додатковий шар формує дедалі абстрактніші ознаки зображення. Одним з перших великих кроків у цьому напрямі стала модель VGGNet, яка продемонструвала ефективність систематичного збільшення глибини за рахунок використання однакових за розміром фільтрів  $3 \times 3$  у великій кількості послідовних шарів [16]. Завдяки такій побудові мережа VGG з глибиною до дев'ятнадцяти шарів забезпечила помітне зростання точності на великих наборах даних, а її проста й однорідна архітектура стала базовим шаблоном для подальших досліджень та прикладних рішень.

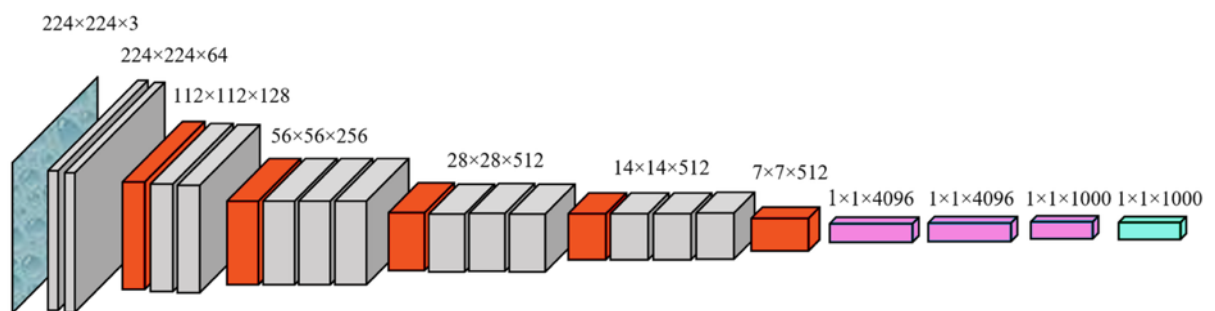


Рисунок 1.7. – Модель VGGNet

Подальший розвиток глибоких мереж пов'язаний із запровадженням залишкових з'єднань у архітектурі ResNet [18]. Основна ідея полягає в тому, що замість безпосереднього навчання відображення  $x \rightarrow y$  мережа навчає залишкову функцію  $F(x)$  так, що вихідний сигнал обчислюється як:

$$y = F(x, \{W_i\}) + x \quad (1.1)$$

де  $x$  – вхідні дані,  $F(x, \{W_i\})$  – нелінійне перетворення, реалізоване нейронною мережею, а  $y$  – вихідний результат. Залишкові з'єднання дозволяють сигналу проходити в обхід певних шарів, що сприяє кращій конвергенції та дозволяє використовувати дуже глибокі мережі (наприклад, ResNet-152, ResNet-101, ResNet-50).

Ще одним напрямом ускладнення архітектур стала поява моделей DenseNet, де впроваджено щільні з'єднання між усіма шарами всередині блоку [19]. Кожний шар отримує на вході конкатенацію виходів усіх попередніх шарів і, своєю чергою, передає власні ознаки далі. Така організація забезпечує інтенсивне повторне використання вже обчислених представлень, покращує поширення градієнтів і дає можливість зменшити кількість параметрів без втрати точності. У сукупності ці підходи демонструють загальну тенденцію розвитку комп'ютерного зору у бік дедалі глибших і більш зв'язаних мереж, здатних моделювати складні залежності у візуальних даних і слугувати базою для побудови високоточних систем розпізнавання об'єктів.

#### 1.4.2 Комбіновані підходи CNN і трансформери

Окремою тенденцією розвитку архітектур комп'ютерного зору стало поєднання згорткових мереж і трансформерів в єдиних моделях. Традиційні CNN добре вловлюють локальні просторові патерни та ефективно працюють на зображеннях високої роздільної здатності, однак їх здатність моделювати

довготривалі залежності обмежена розміром рецептивного поля. Трансформери, навпаки, природно враховують глобальний контекст сцени за рахунок механізму самоуваги, проте потребують значних обчислювальних ресурсів і великих обсягів даних для навчання. Тому логічним кроком стало створення гібридних рішень, де використовується сильні сторони обох підходів.

Архітектура Vision Transformer ViT є одним з перших прикладів успішного перенесення трансформерів у сферу аналізу зображень [20]. Вхідне зображення у ViT розбивається на невеликі патчі, які перетворюються на послідовність векторів і подаються до блоку самоуваги. Механізм самоуваги обчислює матрицю ваг на основі скалярних добутків між запитами, ключами та значеннями:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (1.2)$$

де  $Q$ ,  $K$ ,  $V$  – це матриці запитів (query), ключів (key) та значень (value), а  $d_k$  – розмірність ключів. Така операція дає змогу кожному елементу послідовності враховувати інформацію з усіх інших елементів і тим самим моделювати глобальні взаємозв'язки між ділянками зображення. У результаті модель здатна краще розрізняти об'єкти, які залежать від контексту сцени, що особливо важливо для складних задач класифікації та детектування. Практичні системи часто використовують гібридні архітектури, у яких згорткові шари відповідають за початкове виділення низькорівневих ознак, а трансформерні блоки працюють на більш компактному представленні й узагальнюють глобальний контекст. Подібний підхід реалізовано в моделі ConvNeXt, де класична згорткова мережа переосмислюється з урахуванням дизайну сучасних трансформерів [21]. У ній застосовуються великі ядра згортки, нормалізація за каналами та впорядкована побудова блоків, що робить архітектуру ближчою до Vision Transformer за структурою, але зберігає високу

ефективність згорткових операцій. Такий компроміс дозволяє досягати рівня точності, порівнюваного з чистими трансформерними моделями, і водночас утримувати прийнятну швидкодію та вимоги до обчислювальних ресурсів.

Для задач комп'ютерного зору в режимі реального часу, зокрема для безпілотних платформ, комбіновані CNN–трансформер архітектури відкривають можливість одночасно використовувати локальну чутливість згорток і глобальне узагальнення контексту, притаманне самоувазі. Це створює передумови для більш надійного виявлення складних об'єктів на насиченому фоні за умов обмежених ресурсів і робить гібридні моделі перспективною основою для побудови автоматизованих систем розпізнавання.

### **1.4.3 Легковагі моделі для мобільних і вбудованих пристроїв**

Поширення мобільних і вбудованих платформ, зокрема смартфонів, IoT-пристроїв та безпілотних літальних апаратів, висунуло вимогу виконувати задачі комп'ютерного зору в умовах жорстких обмежень щодо енергоспоживання, обсягу пам'яті та обчислювальних ресурсів. У відповідь на це з'явився клас легковагих архітектур, для яких пріоритетом є не максимальна точність, а збалансоване співвідношення точності та швидкодії за фіксованого бюджету ресурсів. До найпоширеніших рішень належать сімейства MobileNet, ShuffleNet та EfficientNet, що цілеспрямовано оптимізують структуру згорткових мереж під обмежені платформи.

Основною ідеєю таких моделей є зменшення кількості параметрів і операцій при збереженні здатності мережі виділяти інформативні ознаки. У MobileNet використовуються глибокі відокремлені згортки, які розділяють просторову фільтрацію та комбінування каналів на дві послідовні операції. Це різко зменшує обчислювальну складність порівняно з класичною згорткою того самого розміру. У ShuffleNet додатково застосовується групова згортка та операція перемішування каналів, що дозволяє ефективно використовувати

вузькі шари без втрати здатності моделі до обміну інформацією між каналами. Такі архітектурні прийоми часто доповнюються прорідженням ваг, квантуванням та іншими формами компресії, завдяки чому моделі можуть працювати на пристроях з обмеженим обсягом пам'яті та без апаратного прискорювача.

Окремим напрямом розвитку легковагих мереж стало узгоджене масштабування архітектури, запропоноване в EfficientNet [22]. Замість ручного підбору глибини, ширини та роздільної здатності для кожної версії моделі використовується єдина схема масштабування, у якій ці параметри змінюються узгоджено. Для заданого коефіцієнта масштабування  $\phi$  глибина, ширина і роздільна здатність визначаються співвідношеннями

$$\text{depth: } d = \alpha^\phi, \quad \text{width: } w = \beta^\phi, \quad \text{resolution: } r = \gamma^\phi \quad (1.3)$$

де  $\alpha$ ,  $\beta$ ,  $\gamma$  задають пропорції зміни відповідних характеристик, а  $\phi$  контролює загальний рівень масштабування моделі. Такий підхід дає змогу систематично будувати цілу лінійку мереж від найменших, придатних для мобільних пристроїв, до більших варіантів для серверного розгортання, зберігаючи збалансованість між кількістю параметрів, точністю та швидкодією. Для задач комп'ютерного зору на безпілотних платформах це особливо важливо, оскільки дозволяє підібрати конфігурацію, яка відповідає конкретним обмеженням бортового обладнання та вимогам до часу обробки.

#### 1.4.4 Мультимодальні моделі

Важливим напрямом розвитку комп'ютерного зору стали мультимодальні моделі, які одночасно працюють з кількома типами даних і формують спільний простір ознак для зображень, тексту, а інколи й аудіо чи інших сигналів. На відміну від класичних підходів, де зображення обробляється ізольовано, мультимодальні системи навчаються на парах

«зображення – опис» або на більш складних комбінаціях даних. Це дозволяє не лише розпізнавати об'єкти, а й пов'язувати їх із природномовними запитами, формулювати узагальнені описи сцен, виконувати пошук зображень за текстом і навпаки.

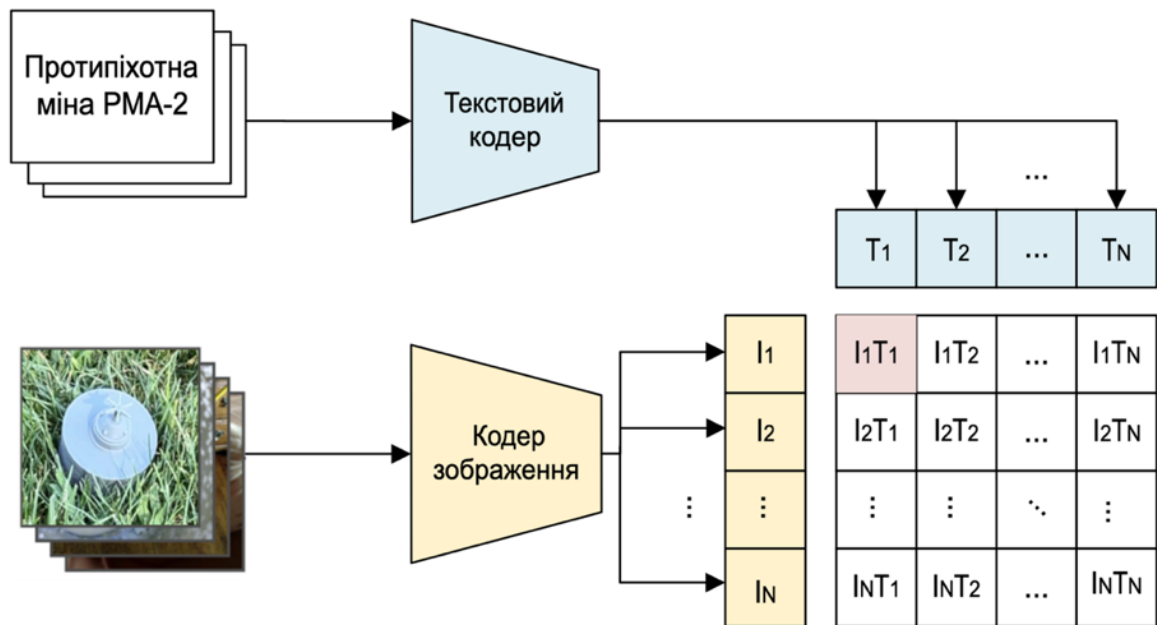


Рисунок 1.8 – Схематичне зображення процесу навчання моделі

Прикладом таких рішень є модель CLIP, у якій візуальний і текстовий енкодери (рис. 1.8) навчаються спільно на великій кількості підписаних зображень [23]. Модель оптимізує представлення таким чином, щоб відповідні пари «зображення – текст» розташовувались близько одна до одної у спільному векторному просторі, а невідповідні пари розділялися. Завдяки цьому система здатна виконувати відкриту класифікацію, коли класи задаються не фіксованим списком, а довільними текстовими описами. Інший приклад становлять генеративні моделі на кшталт DALL-E, які вчаться будувати зображення за текстовим запитом. У таких моделях текст використовується як керуючий сигнал, що визначає композицію, стиль і зміст зображення. Мультиmodalний підхід виходить за межі класичної задачі розпізнавання і переходить до семантичного розуміння сцени. Для систем автоматизованого моніторингу це означає можливість пов'язувати візуальні

спостереження з додатковими каналами інформації, наприклад з описами оператора, контекстними даними про місцевість або показаннями спеціалізованих сенсорів. Об'єднання таких джерел у спільному представленні підвищує стійкість до завад, зменшує кількість хибних спрацювань і дозволяє будувати більш гнучкі системи підтримки прийняття рішень, у яких запити та пояснення результатів можуть формулюватися природною мовою.

### 1.5 Формалізація та порівняльний аналіз рівнів обчислень

Сучасні системи просторового моніторингу, що генерують великі обсяги сенсорних даних (відеопотоки високої роздільної здатності, телеметрія), стикаються з проблемою обмеженої пропускної здатності каналів зв'язку та вимогами до мінімізації затримок (*latency*). Для формалізації вибору архітектури доцільно розглянути три парадигми: граничні обчислення (*Edge Computing*), туманні обчислення (*Fog Computing*) та хмарні обчислення (*Cloud Computing*). У контексті БПЛА, *Edge*-рівень представлений бортовим обчислювачем, *Fog*-рівень – наземною станцією керування (*GCS*) та локальними серверами, а *Cloud*-рівень – віддаленими дата-центрами. Вибір оптимальної архітектури  $A_{opt}$  можна представити як задачу мінімізації цільової функції витрат  $C$ , що включає затримку  $L$ , енергоспоживання  $E$  та використання смуги пропускання  $B$ :

$$A_{opt} = \arg \min_{A \in \{Edge, Fog, Cloud\}} (\alpha L(A) + \beta E(A) + \gamma B(A)) \quad (1.4)$$

де  $\alpha, \beta, \gamma$  – вагові коефіцієнти, що визначаються пріоритетами місії (для розмінування критичними є  $L$  та точність детектування).

Нижче наведено порівняльний аналіз цих парадигм, адаптований до умов роботи роботизованих платформ моніторингу [24].

Таблиця 1.2 – Порівняльний аналіз рівнів обчислень для задач БПЛА

| Характеристика                     | Edge Computing<br>(Борт БПЛА)                                                | Fog Computing<br>(Наземна станція)                                     | Cloud Computing<br>(Хмара)                                                  |
|------------------------------------|------------------------------------------------------------------------------|------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| <b>Локалізація ресурсів</b>        | Безпосередньо на платформі (Raspberry Pi 4B)                                 | Локальна мережа (LAN/Wi-Fi), польовий ноутбук, Edge-Gateway            | Віддалені централізовані сервери (AWS, Hetzner)                             |
| <b>Латентність (Latency)</b>       | Низька / Детермінована (< 10-20 мс).                                         | Середня (20-100 мс). Залежить від якості радіоканалу «Борт-Земля».     | Висока / Варіативна (> 100-200 мс). Включає затримки глобальної мережі.     |
| <b>Вимоги до смуги пропускання</b> | Мінімальні. Передача лише результатів (метаданих).                           | Високі для передачі сирого відео («Raw Data»), помірні для стисненого. | Критично високі. Потребує стабільного широкосмугового каналу (4G/Starlink). |
| <b>Обчислювальна потужність</b>    | Обмежена (CPU ARM, NPU, ~1-5 TOPS). Обмеження по SWaP (Size, Weight, Power). | Середня/Висока (Дискретні GPU, ~10-100 TOPS).                          | Практично необмежена (Кластери GPU, TPU).                                   |
| <b>Енергоспоживання</b>            | Критичне. Живлення від бортової батареї впливає на час польоту.              | Помірне. Живлення від наземних джерел або генераторів.                 | Не є обмеженням для оператора (винесено за межі системи).                   |
| <b>Безпека та приватність</b>      | Висока. Дані обробляються in-situ.                                           | Середня. Дані в межах контрольованого периметра.                       | Ризик перехоплення при передачі через публічні мережі.                      |
| <b>Застосування у системі</b>      | Первинна обробка, стиснення, синхронізація сенсорів, Real-time навігація.    | Детекція об'єктів (YOLO), локальне картування, інтерфейс оператора.    | Глибока аналітика (VLM), довгострокове зберігання, навчання моделей.        |



Аналіз, наведений у табл. 1.2, демонструє, що для задачі виявлення мін у реальному часі виключне використання хмарних обчислень є неприйнятним через ризик втрати зв'язку та високу латентність, що може призвести до пропуску небезпечних об'єктів. З іншого боку, ресурси рівня Edge (Raspberry Pi 4B) недостатні для виконання важких моделей (наприклад, Vision Language Models або глибоких трансформерів) з прийнятним FPS. Це обґрунтовує необхідність гібридної архітектури, де:

1. **Edge** виконує збір, часову синхронізацію та попередню фільтрацію.
2. **Ground** (наземна станція) виконує основну детекцію нейромережею YOLOv8.
3. **Cloud** залучається асинхронно для верифікації складних випадків та оновлення карт.

Така структура дозволяє балансувати між швидкістю реакції та глибиною аналізу, що є критичним для задач гуманітарного розмінювання.

## 1.6 Методи збору, обробки та збереження даних на бортових обчислювальних платформах

Еволюція робототехніки останнього десятиліття характеризується фундаментальним зміщенням парадигми обробки інформації від централізованих хмарних обчислень до розподілених архітектур [25], де ключову роль відіграють граничні обчислення (Edge AI). Ця трансформація зумовлена необхідністю забезпечення автономності роботизованих платформ в умовах обмеженого зв'язку, мінімізації затримок у контурах керування реального часу та зростанням обсягів сенсорних даних, передача яких у «сирому» вигляді стає технічно неможливою або економічно недоцільною. Сучасні безпілотні системи, оснащені мультиспектральними камерами, лідачами та високочастотними інерціальними датчиками, перетворилися на складні інструменти збору даних, які вимагають інтеграції високопродуктивних конвеєрів (pipelines) для захоплення, попередньої

обробки, інференсу нейронних мереж та ефективного збереження результатів безпосередньо на борту носія.

### **1.6.1 Огляд та класифікація сучасних роботизованих платформ**

Роботизовані платформи, що інтегрують технології штучного інтелекту, формують широкий спектр технічних засобів, специфіка яких визначається середовищем функціонування, фізичними обмеженнями та характером виконуваних місій. Розуміння архітектурних відмінностей між різними класами роботів (рис. 1.9) є критичним для проектування ефективних Edge систем оскільки вимоги до енергоспоживання, обчислювальної потужності та алгоритмів сприйняття суттєво варіюються. Безпілотні літальні апарати (БПЛА) займають домінуюче положення у сферах дистанційного зондування, моніторингу критичної інфраструктури та пошуково-рятувальних операцій завдяки своїй здатності швидко переміщуватися у тривимірному просторі та забезпечувати унікальні ракурси спостереження. У межах цього класу виділяють кілька підтипів, кожен з яких накладає свої обмеження на бортову електроніку. Мультироторні системи (квадрокоптери, гексакоптери) характеризуються високою маневреністю та здатністю до зависання, що дозволяє виконувати детальну інспекцію стаціонарних об'єктів та забезпечувати стабільну платформу для роботи алгоритмів комп'ютерного зору. Однак їх енергетична ефективність є відносно низькою, що обмежує час польоту та, відповідно, допустиму вагу бортового обчислювача. Апарати фіксованого крила (Fixed-Wing) використовують аеродинамічну підйомну силу крила, що забезпечує значно більшу тривалість та дальність польоту, роблячи їх ідеальними для моніторингу великих площ [15]. Специфікою таких платформ є необхідність постійного руху для підтримання висоти, що вимагає від систем комп'ютерного зору роботи з об'єктами, які швидко зміщуються у кадрі, та компенсації розмиття зображення (motion blur).



Рисунок 1.9 – Схема класифікації роботизованих платформ

Гібридні схеми VTOL (Vertical Take-Off and Landing) поєднують переваги вертикального зльоту мультироторів та економічності крилатих апаратів на маршових ділянках, що дозволяє інтегрувати більш потужне корисне навантаження при збереженні високої мобільності. Ключовим викликом для всіх типів БПЛА залишаються жорсткі обмеження SWaP (Size, Weight, and Power), що змушує інженерів балансувати між продуктивністю нейромережових прискорювачів та ємністю акумуляторних батарей.

Наземні роботизовані платформи (UGV) представляють собою інший клас систем, які характеризуються значно вищою вантажопідйомністю та меншими обмеженнями щодо енергозабезпечення порівняно з БПЛА [26]. Це дозволяє оснащувати UGV потужними серверними обчислювальними модулями, масивами лідарів високої роздільної здатності та складними маніпуляторами. Специфіка UGV полягає у необхідності навігації в умовах безпосереднього контакту з перешкодами та складному рельєфі, що вимагає обробки даних глибини в реальному часі та побудови детальних карт прохідності. На відміну від БПЛА, які часто мають «чистий» фон (земля або небо), UGV працюють у насиченому динамічному середовищі, де оклюзії (перекриття) об'єктів є нормою, що вимагає застосування складніших алгоритмів відстеження та передбачення поведінки оточуючих агентів [27].

Безпілотні підводні апарати (UUV) та надводні судна (USV) функціонують у середовищі, яке накладає найбільш суворі обмеження на зв'язок та навігацію. Електромагнітні хвилі радіодіапазону швидко загасають у воді, що унеможливорює використання GPS та високошвидкісну передачу відеопотоку на поверхню. Це робить автономність UUV критично важливою: всі рішення щодо класифікації об'єктів, навігації та уникнення перешкод повинні прийматися виключно на борту. Системи комп'ютерного зору стикаються зі специфічними оптичними ефектами, такими як розсіювання світла, поглинання кольорів та рефракція, що вимагає застосування спеціалізованих методів попередньої обробки зображень та адаптації нейронних мереж до умов низької контрастності.

Окремим перспективним напрямком є створення гетерогенних груп роботів, які об'єднують переваги різних платформ. Наприклад, тандем БПЛА та UGV дозволяє використовувати повітряний апарат для глобального картування та цілевказівки, тоді як наземна платформа виконує детальне обстеження або фізичну взаємодію з об'єктами, слугуючи також зарядною станцією та обчислювальним хабом для дрона [28]. Така синергія вимагає розробки уніфікованих протоколів обміну даними та розподілених алгоритмів Edge AI, здатних динамічно перерозподіляти обчислювальні задачі між агентами системи.

### **1.6.2 Застосування комп'ютерного зору для різних типів платформ**

Технології комп'ютерного зору (Computer Vision) виступають однією з критично важливих складових сенсорного забезпечення сучасних автономних систем, оскільки надають можливість вилучення семантичної інформації з візуальних даних, що є необхідним для прийняття рішень у неструктурованому середовищі. Архітектура алгоритмічного забезпечення та пріоритетність розв'язуваних задач суттєво залежать від кінематики та функціонального призначення носія. Зокрема, для безпілотних літальних

апаратів (БПЛА) домінуючими є завдання дистанційного зондування, що включають виявлення та класифікацію об'єктів інфраструктури, транспортних засобів або біологічних об'єктів на земній поверхні (рис. 1.10).



Рисунок 1.10 – Схематичне зображення БПЛА над об'єктами

Специфіка аерофотозйомки, яка характеризується значною варіативністю масштабу залежно від висоти польоту та зміною ракурсу від планового до перспективного, зумовлює необхідність застосування нейромережових архітектур, інваріантних до афінних перетворень, наприклад, моделей з пірамідальними ознаками (Feature Pyramid Networks). Окрім того, в умовах відсутності або зашумлення сигналів супутникової навігації, візуальні дані стають основою для систем візуально-інерційної одометрії та методів одночасної локалізації та картування (SLAM), забезпечуючи оцінку положення шляхом відстеження характерних точок ландшафту. У спеціалізованих застосуваннях, таких як точне землеробство, використання мультиспектральних сенсорів дозволяє розширити аналітичні можливості системи, забезпечуючи розрахунок вегетаційних індексів та діагностику стану рослин, що недосяжно для стандартних оптичних засобів.

На відміну від повітряних платформ, у сегменті наземних роботизованих комплексів (UGV) акцент зміщується на завдання ближньої навігації, безпосередньої взаємодії з оточенням та забезпечення безпеки руху. Ключову роль тут відіграють алгоритми семантичної сегментації, які дозволяють класифікувати підстильну поверхню для визначення зон прохідності та своєчасної ідентифікації перешкод. Для побудови детальних метричних карт навколишнього середовища, таких як воксельні моделі або карти зайнятості широко застосовується комплексування даних стереокамер та 3D-лідарів, що підвищує надійність планування траєкторії. Експлуатація наземних платформ також пов'язана з викликами, зумовленими складними умовами освітлення, наявністю тіней та впливом атмосферних опадів, що вимагає впровадження адаптивних методів попередньої обробки зображень. У сценаріях військового або рятувального призначення ефективність системи суттєво підвищується завдяки інтеграції тепловізійних каналів, що через алгоритми злиття різнорідних даних забезпечує надійне виявлення об'єктів за тепловими сигнатурами навіть в умовах обмеженої видимості [25].

Для морських та підводних систем (USV/UUV) комп'ютерний зір вирішує задачі інспекції підводної інфраструктури (трубопроводи, опори вітрогенераторів, корпуси суден), пошуку мін та моніторингу біорізноманіття. Підводне середовище створює унікальні виклики: поглинання червоного спектру призводить до спотворення кольорів, а наявність завислих частинок створює шум, схожий на «сіль і перець». Тому конвеєри обробки даних на UUV часто включають етапи відновлення кольору (color restoration) та видалення туману (dehazing) на основі фізичних моделей розповсюдження світла або GAN-мереж (Generative Adversarial Networks). На надводних апаратах (USV) комп'ютерний зір інтегрується з радаром для виявлення та класифікації інших суден відповідно до правил МПЗЗС (Міжнародні правила попередження зіткнень суден у морі), що вимагає високої дальності виявлення та точності оцінки курсу цілі.

### 1.6.3 Методи збору, збереження та обробки даних на Edge рівні

Ефективність системи Edge AI визначається не лише архітектурою нейронної мережі, але й якістю реалізації всього конвеєра даних який забезпечує безперервний потік інформації від фізичних сенсорів до алгоритмів прийняття рішень та підсистем довготривалого збереження (рис. 1.11).

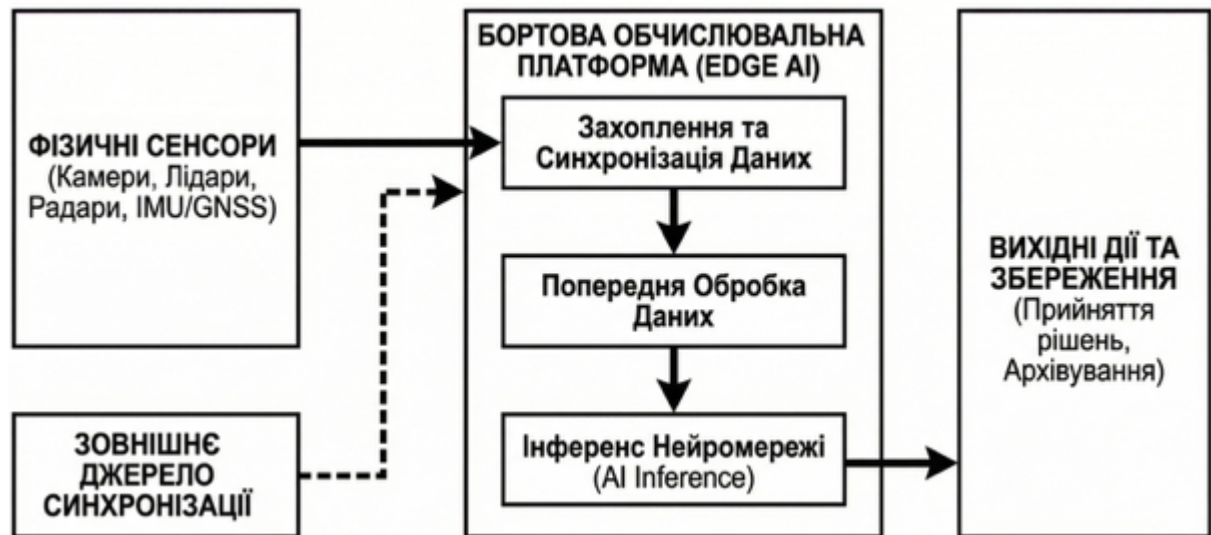


Рисунок 1.11 – Спрощена схема конвеєра даних

Цей процес охоплює етапи захоплення синхронізації, попередньої обробки, інференсу та архівування. На першому етапі дані з різномірних сенсорів надходять до бортового комп'ютера через високошвидкісні інтерфейси. Відеопотоки з камер зазвичай передаються через інтерфейс MIPI CSI-2, який забезпечує прямий доступ до пам'яті (DMA) з мінімальним завантаженням центрального процесора, або через USB 3.0/GigE Vision для периферійних камер. Дані з лідарів та радарів надходять через Ethernet, а телеметрія з IMU та GPS – через шини CAN, SPI або UART. Фундаментальною проблемою мультисенсорних систем є часова синхронізація (time synchronization). Розходження у часових мітках між кадром камери та даними гіроскопа навіть у кілька мілісекунд може призвести до значних помилок у задачах SLAM або VIO, особливо при швидкому русі платформи. Для вирішення цієї проблеми використовуються апаратні тригери (hardware

triggers), протоколи PTP (Precision Time Protocol, IEEE 1588) або PPS-сигнали від GNSS-приймачів, які дозволяють синхронізувати всі сенсори з точністю до мікросекунд [29]. «Сирі» дані часто непридатні для безпосереднього використання нейронними мережами. Відеопотоки вимагають процедур дебаєрінга (demosaicing), корекції дисторсії об'єктива, балансу білого та нормалізації яскравості. Для розвантаження CPU ці операції виконуються на спеціалізованих апаратних блоках – процесорах обробки сигналів зображення (ISP – Image Signal Processor) або на графічних ядрах (GPU) з використанням API типу CUDA або OpenCL. Для лідарних даних виконується фільтрація шумів, видалення точок, що належать корпусу робота (self-filtering), та перетворення хмари точок у структуровані формати.

Обмежений обсяг бортових накопичувачів та висока інтенсивність потоків даних (наприклад, нестиснене 4K відео) вимагають ефективних стратегій збереження. Традиційний підхід в екосистемі ROS (Robot Operating System) базується на використанні файлового формату rosbag, який записує серіалізовані повідомлення у хронологічному порядку. Однак класичний rosbag має недоліки при роботі з великими файлами та повільний пошук. Сучасною альтернативою є стандарт MCAP (Modular Container for Analytics and Performance), який оптимізований для запису гетерогенних потоків даних з високою пропускнуою здатністю, підтримує індексацію для швидкого довільного доступу (seeking) та дозволяє використовувати різні методи стиснення (LZ4, Zstd). MCAP є контейнерним форматом, що дозволяє зберігати дані різних стандартів серіалізації (ROS 2 CDR, Protobuf, JSON) в одному файлі, спрощуючи інтеграцію з інструментами візуалізації, такими як Foxglove Studio. Окрім файлового запису логів, зростає популярність використання вбудованих баз даних (embedded databases) безпосередньо на борту робота. Рішення на кшталт SQLite або DuckDB, а також спеціалізовані бази даних часових рядів (Time Series Databases), дозволяють виконувати структуровані SQL-запити до накопиченої телеметрії в реальному часі [30]. Це відкриває можливості для реалізації складних сценаріїв автономності, коли



робот може аналізувати власну історію або виконувати семантичний пошук по метаданих ("знайти координати всіх виявлених об'єктів класу «людина»»). Для економії простору часто застосовуються кільцеві буфери які зберігають повний потік даних лише за певний проміжок часу, перезаписуючи старі дані, якщо не відбулося критичної події.

Архітектура виконання інференсу на Edge передбачає використання оптимізованих середовищ виконання (runtime), таких як TensorRT, ONNX Runtime або TFLite, які адаптують модель під конкретне апаратне забезпечення, виконуючи злиття шарів (layer fusion), калібрування точності та управління пам'яттю. Важливим аспектом є керування чергами запитів та пріоритезація: задачі навігації та уникнення перешкод повинні мати найвищий пріоритет та мінімальну латентність, тоді як задачі аналітики місії можуть виконуватися у фоновому режимі. Результати інференсу (координати bounding box, маски сегментації, класи) збагачуються часовими мітками та геопросторовими даними, після чого передаються в систему керування або зберігаються як метадані в форматі JSON або Parquet для подальшого аналізу.

## **1.7 Постановка задачі**

Проведений аналіз сучасного стану технологій гуманітарного розмінування та методів дистанційного зондування виявив фундаментальну суперечність між нагальною потребою у високопродуктивному обстеженні значних площ забруднених територій та наявними технологічними обмеженнями існуючих засобів. Традиційні методи, що базуються на ручній праці саперів, забезпечують високу надійність, проте є критично повільними та пов'язані з неприпустимо високим ризиком для життя персоналу. Водночас існуючі роботизовані рішення та безпілотні авіаційні комплекси переважно функціонують у режимі пасивного збору даних, виступаючи лише носіями сенсорного навантаження без здатності до автономного прийняття рішень у реальному часі. Такий розімкнений цикл керування призводить до значної

часової затримки між виявленням підозрілого об'єкта та його верифікацією, а також породжує проблему надмірної кількості хибних спрацювань при використанні моносенсорних систем в умовах складного рельєфу та рослинності.

Критичним недоліком сучасних систем є відсутність ефективних механізмів автоматичної адаптації поведінки платформи до контексту місії та результатів поточного аналізу даних. Більшість існуючих рішень не здатні динамічно змінювати параметри польоту для деталізації огляду підозрілих зон або компенсувати деградацію одного каналу сприйняття за рахунок іншого без втручання оператора. Крім того, обмежені обчислювальні ресурси бортових систем перешкоджають впровадженню складних нейромережевих моделей глибокого навчання безпосередньо на борту, що створює дилему між автономністю платформи та глибиною аналітичної обробки.

Виходячи з цього, виникає об'єктивна необхідність у створенні принципово нової архітектури системи керування, яка б забезпечувала інтеграцію різнорідних сенсорних даних, семантичне розуміння сцени та адаптивне керування рухом у єдиному замкненому контурі. Наукова задача полягає у розробці методів та засобів, що дозволять трансформувати безпілотну платформу з пасивного інструменту реєстрації у активного інтелектуального агента, здатного виконувати пошукові місії з показниками ефективності та надійності, наближеними до роботи кваліфікованого фахівця-людини, але з мінімізацією ризиків та суттєвим підвищенням продуктивності процесу. Розв'язання цієї задачі вимагає комплексного підходу, що поєднає методи теорії автоматичного керування, комп'ютерного зору, штучного інтелекту та розподілених обчислень в межах єдиної багаторівневої системи.

## **1.8 Висновки за розділом 1**

У першому розділі дисертаційної роботи здійснено вичерпний системний аналіз сучасного стану, технологічних викликів та перспективних

напрямів розвитку автоматизованих систем керування роботизованими платформами, що застосовуються для виконання завдань просторового моніторингу навколишнього середовища, з особливим фокусом на проблематику гуманітарного розмінування. Проведене дослідження дозволило глибоко концептуалізувати проблему та сформувати науково-технічне підґрунтя для розробки інноваційної архітектури безпілотного авіаційного комплексу.

Встановлено, що сучасний стан глобальної безпеки характеризується зростанням мінної загрози, де традиційні методи ручного розмінування є критично повільними та несуть високі ризики для життя персоналу. Аналіз існуючих апаратних рішень, таких як магнітометричні комплекси на базі БПЛА (наприклад, платформа «Maggy» [9]), тепловізійні дрони та наземні роботизовані платформи (концепція «UGO-first» [11]), виявив їхню фундаментальну обмеженість. Більшість сучасних систем функціонують у розімкненому контурі керування (open-loop), де безпілотник виступає виключно в ролі пасивного носія сенсорів. Дані збираються та передаються на наземну станцію для відкладеної офлайн-обробки, що унеможливорює оперативну реакцію системи на виявлені загрози безпосередньо під час польоту. Крім того, моносенсорні системи демонструють високу вразливість до умов середовища: оптичні камери страждають від візуального маскування та складного освітлення, тоді як магнітометри є неефективними для виявлення сучасних пластикових мін (наприклад, ПФМ-1) з мінімальним вмістом металу.

Дослідження еволюції методів комп'ютерного зору засвідчило безальтернативність переходу від класичних алгоритмів виділення ознак (HOG, Naar-каскади) до методів глибокого машинного навчання. Показано, що сучасні згорткові нейронні мережі (CNN), зокрема архітектури сімейства YOLO, здатні забезпечити високу точність локалізації об'єктів у режимі, наближеному до реального часу. Однак їх застосування на борту мобільних платформ жорстко лімітується обмеженнями SWaP (Size, Weight, and Power).

Більше того, традиційні нейромережі залишаються схильними до помилок першого роду (хибних позитивних спрацювань) в умовах складного неоднорідного фону, плутаючи елементи рельєфу або побутове сміття з вибухонебезпечними предметами.

Підсумовуючи результати першого розділу, сформульовано наукову задачу дисертації, яка полягає у розробці методів, математичних моделей та алгоритмів для трансформації безпілотної платформи з пасивного реєстратора на активного інтелектуального агента. Такий агент має функціонувати в межах замкненого циклу керування, де результати мультимодального сенсорного злиття (візуальних даних та сигналів металодетектора) безпосередньо ініціюють адаптивні зміни траєкторії польоту для деталізації обстеження підозрілих зон, забезпечуючи високу надійність розпізнавання та оперативність реакції в польових умовах.

## **РОЗДІЛ 2. АРХІТЕКТУРА АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ БАГАТОРІВНЕВОЇ РОБОТИЗОВАНОЇ ПЛАТФОРМИ ПРОСТОРОВОГО МОНІТОРИНГУ**

Проектування децентралізованої комп'ютерної системи багаторівневої хмарної обробки відеозображення, що застосовується для вирішення задач комп'ютерного зору, передбачає розробку комплексної апаратної платформи (рис. 2.1). Така платформа має бути здатною ефективно збирати, первинно обробляти та передавати великі обсяги різномірних даних (відео, аудіо, GPS, телеметрія та інші супутні метадані) в умовах реального часу. Особливо актуальною ця задача є в контексті вирішення складних і критично важливих завдань, таких як моніторинг та ідентифікація небезпечних об'єктів, пошук предметів, інспектування територій та об'єктів інфраструктури в умовах складної місцевості.

Основним вимогами до такої системи є її надійність, стабільність роботи в умовах впливу зовнішніх факторів (вітер, температурні коливання, електромагнітні перешкоди), низький рівень енергоспоживання та можливість ефективної інтеграції з хмарними сервісами та локальними обчислювальними ресурсами. Враховуючи перелічені вимоги, було прийнято рішення побудувати систему на базі безпілотного літального апарата гексакоптерного типу, який забезпечує значні переваги в порівнянні з іншими видами БПЛА (квадрокоптерами, літаками-фіксованого крила), завдяки більшій вантажопідйомності, маневреності та можливості стабільного зависання в повітрі.

Запропонована апаратна архітектура (рис. 2.1) складається з ряду спеціалізованих компонентів, кожен із яких виконує свою чітко визначену роль у загальному процесі збору та обробки даних. Ядром системи є мікрокомп'ютер Raspberry Pi 4B, який обраний завдяки оптимальному співвідношенню продуктивності, енергоспоживання та можливостей підключення різних периферійних пристроїв. Цей мікрокомп'ютер відповідає за первинне збирання відеоданих із камери високої роздільної здатності

(ArduCAM 64MP), оцифровування та аналіз перетворення сигналу металодетектора за допомогою спеціалізованого звукового інтерфейсу, а також забезпечує інтеграцію і синхронізацію цих даних з високоточними координатами GPS, що надходять із контролера польоту Pixhawk 6C та GPS-модуля Holybro M9N.

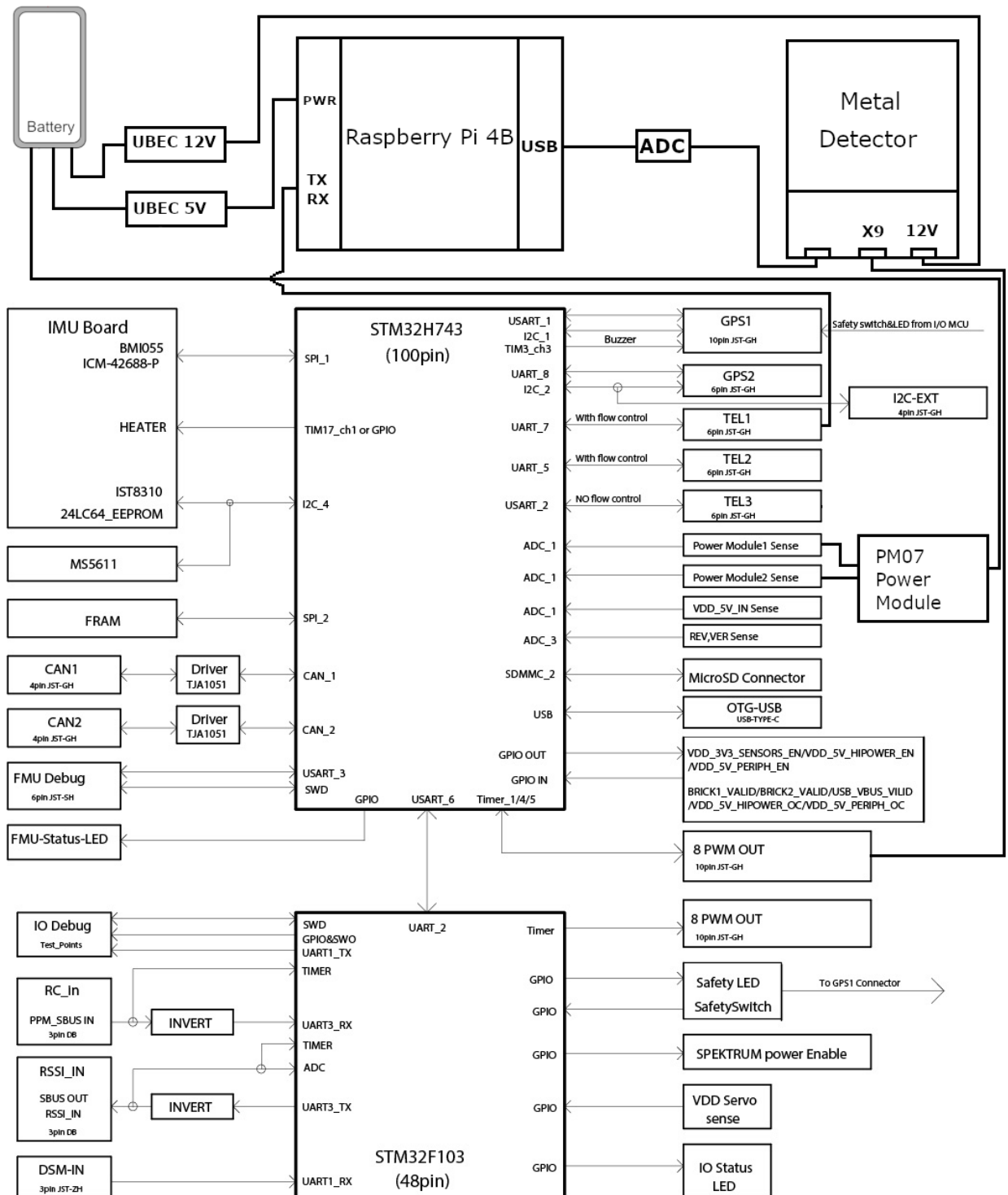


Рисунок 2.1 – Схема основних апаратних компонентів комплексу

Система живлення та розподілу енергії базується на розподільній платі Holybro PM07, яка дозволяє одночасно і безпечно жити всі компоненти, контролюючи споживання струму та напругу. Це важливо для запобігання відмов та перегріву компонентів під час тривалих польотів і забезпечує стабільну роботу усієї системи навіть при значних навантаженнях. Двигуни Tarot 4114-11 KV320, керовані цифровими регуляторами швидкості Hobbywing XRotor 40A, забезпечують потужність, необхідну для стабільного та керованого польоту БПЛА. Завдяки використанню легких та міцних карбонових пропелерів 1555, досягається оптимальне поєднання тяги, ефективності та зниженого рівня вібрацій. Для забезпечення безпеки польоту та точного контролю положення БПЛА використовується лідар Holybro ST VL53L1X, який дозволяє з високою точністю визначати висоту над землею, що критично важливо для польоту на низькій висоті і детальної зйомки поверхні. Управління польотом здійснюється за допомогою сучасного контролера польоту Pixhawk 6C, який використовує просунуті алгоритми стабілізації, автономного польоту та навігації.

Оперативне управління та контроль над БПЛА здійснюються за допомогою радіоапаратури Radiomaster TX12, яка забезпечує надійний зв'язок з апаратом за допомогою цифрового модуля Cyclone SuperD ELRS 2.4 GHz RX. Телеметрія передається через радіомодем Holybro SiK Telemetry Radio V3, який працює у діапазоні 433 МГц та забезпечує стабільну передачу даних на відстані до кількох кілометрів.

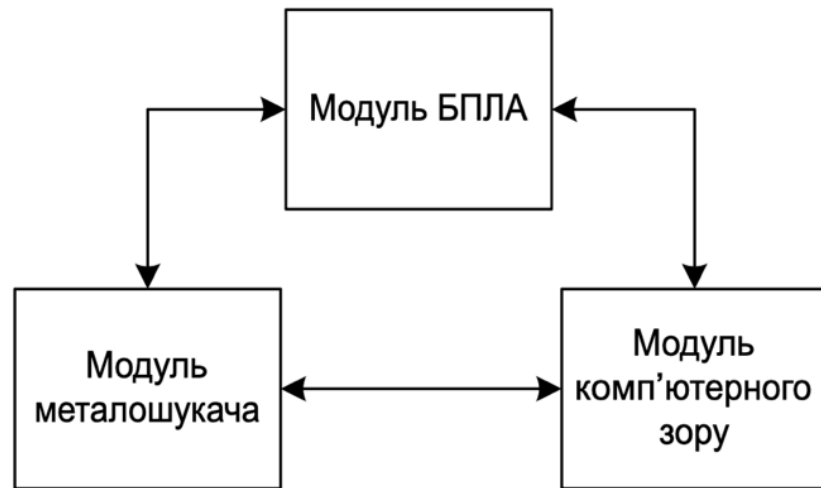


Рисунок 2.2 – Базова схема взаємозв'язку компонентів

Наступні (рис. 2.2) рівні обробки здійснюються вже на потужніших обчислювальних платформах: на локальному ПК із застосуванням неймережі YOLOv8, що забезпечує високу точність класифікації об'єктів. Найбільш складні та неоднозначні випадки або паралельно всі відео кадри аналізуються за допомогою сучасних мультимодальних великих мовних моделей (VLM), таких як OpenAI Chat GPT та Google Gemini, що дозволяють використовувати глибокий контекстуальний аналіз і надають точні експертні висновки щодо об'єктів, які можуть бути складно ідентифіковані автоматично.

Усі ці компоненти разом утворюють інтегровану, стійку до зовнішніх впливів апаратну платформу, що забезпечує якісну і стабільну роботу системи багаторівневої обробки відеозображень та супутніх даних, що дозволяє ефективно вирішувати комплексні задачі комп'ютерного зору в реальних умовах експлуатації. Такий підхід забезпечує гнучкість, надійність та високу продуктивність, необхідні для роботи в умовах оперативних завдань з високою відповідальністю.

## **2.1 Функціональна організація багаторівневої системи обробки відеоданих**

Розроблена в межах дисертаційного дослідження архітектура системи моніторингу базується на сучасній парадигмі розподілених гетерогенних



обчислень, що дозволяє оптимізувати баланс між латентністю прийняття рішень та глибиною аналізу даних. Структурно система організована за ієрархічним трирівневим принципом «борт – наземна станція – хмара», де кожен рівень виконує специфічний набір функцій у межах єдиного наскрізного конвеєра обробки інформації. Така архітектура забезпечує послідовну трансформацію сирих сенсорних даних у верифіковану семантичну інформацію шляхом проходження етапів збору, попередньої фільтрації, неймережевого детектування, мультимодальної верифікації та фінального просторово-часового злиття. Фундаментом апаратної реалізації нижнього, бортового рівня, виступає мобільна платформа на базі гексакоптера, конструктивні особливості якого забезпечують необхідну стабільність просторового положення та вантажопідйомність для розміщення комплексу корисного навантаження. Основним джерелом візуальної інформації обрано високочутливу камеру [33] ArduCAM з роздільною здатністю 64 мегапікселі. Підключення сенсора до обчислювального ядра реалізовано через спеціалізований паралельний інтерфейс MIPI CSI-2, що, на відміну від USB-з'єднань, забезпечує прямий доступ до пам'яті (DMA) та мінімізує затримки передачі кадрів, розвантажуючи центральний процесор. Для гарантування метрологічної якості відеоматеріалу, необхідної для коректної роботи алгоритмів комп'ютерного зору (рис. 2.3), параметри оптичної системи жорстко регламентовані на програмному рівні.

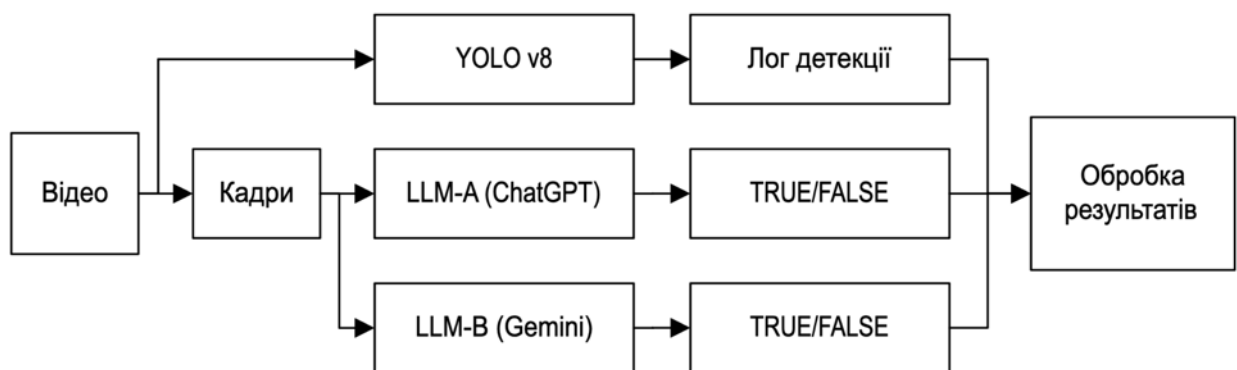


Рисунок 2.3 – Шари обробки відео зображення

Зокрема, ініціалізація камери відбувається з примусовою фіксацією позиції фокусування лінзи на значенні «6.5», що було емпірично визначено як точка гіперфокальної відстані для висоти польоту близько двох метрів. Режим автофокусування програмно переводиться у стан «manual», що дозволяє уникнути небажаних візуальних артефактів та зміни масштабу зображення під час маневрування БПЛА, забезпечуючи стабільність вхідних даних для нейромережі. Частота дискретизації відеопотоку встановлюється на рівні 30 кадрів на секунду, що є оптимальним компромісом між плавністю відеоряду для задач оптичного потоку та обчислювальним навантаженням на систему.

Роль центрального вузла агрегації та первинної обробки даних на борту виконує одноплатний мікрокомп'ютер Raspberry Pi 4B. Він функціонує у тісній взаємодії з польотним контролером Pixhawk 6C, який відповідає за навігацію та стабілізацію платформи, використовуючи дані від мультисистемного GPS-приймача моделі M9N. Унікальною особливістю запропонованої апаратної конфігурації є інтеграція в контур системи активного індукційного сенсора – рамкового металодетектора розміром  $1 \times 1$  м. Сигнал з детектора проходить через спеціалізований аналоговий тракт з каскадом попереднього підсилення та високошвидкісним аналого-цифровим перетворювачем, що дозволяє детектувати магнітні аномалії ґрунту, характерні для вибухонебезпечних предметів. Критично важливим завданням на цьому етапі є забезпечення жорсткої часової синхронізації різнорідних потоків даних: відео  $V(t)$ , телеметрії  $G(t)$  та показників металодетектора  $M(t)$ . Для вирішення цієї задачі джерелом еталонного часу виступає атомний годинник супутників GPS. Часові мітки транслуються з навігаційного модуля на Raspberry Pi за допомогою протоколів синхронізації NTP або chrony, а апаратна лінія PPS (Pulse Per Second) забезпечує мікросекундну точність прив'язки кожного кадру та виміру, що унеможливорює розсинхронізацію даних при їх подальшому злитті.

Окрему увагу приділено надійності підсистеми електроживлення (рис. 2.4), оскільки стабільність напруги є критичною для безперебійної роботи обчислювальних модулів.

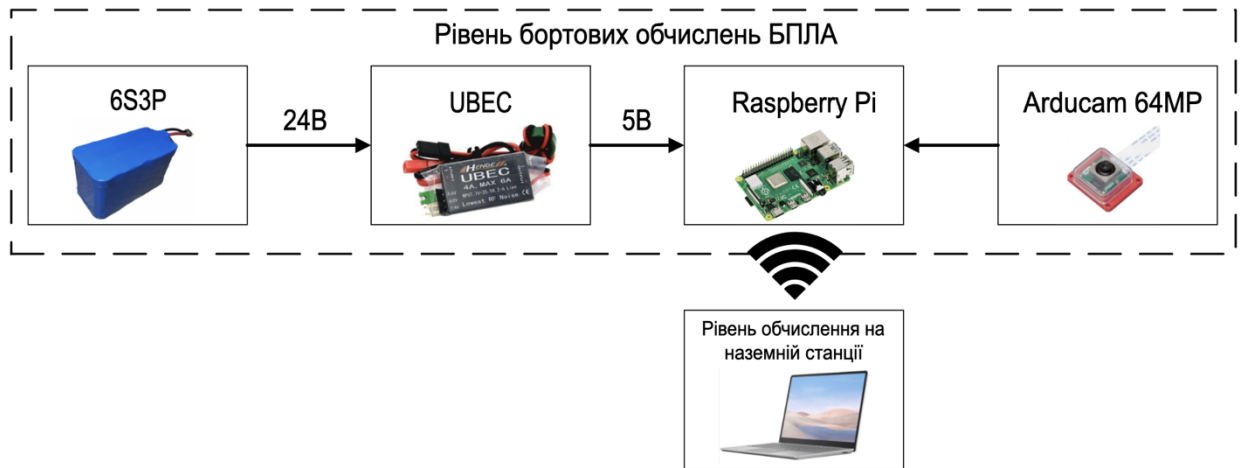


Рисунок 2.4 – Схема підключення Raspberry Pi до живлення та наземної станції

Енергетична автономність комплексу забезпечується основною силовою акумуляторною батареєю конфігурації 6S3P, яка скомпонована з високострумівих літій-іонних елементів типорозміру 21700 моделі Samsung 50S. Вибір даних елементів зумовлений їх високою питомою енергоємністю та здатністю віддавати значні струми без критичного падіння напруги. Для гальванічної розв'язки та живлення чутливої цифрової електроніки, зокрема Raspberry Pi, застосовано вискоєфективний імпульсний регулятор напруги (UBEC) моделі HENGE 4A 2-6S. Цей модуль виконує перетворення нестабільної вхідної напруги бортової мережі, яка може коливатися в межах від 7 до 25.5 В залежно від рівня заряду та навантаження на двигуни, у прецизійно стабілізовані 5 В або 6 В з вихідним струмом до 4 А. Використання окремого контуру живлення через UBEC дозволяє нівелювати вплив просідань напруги силової батареї під час різких маневрів, запобігаючи аварійним перезавантаженням комп'ютера та пошкодження файлової системи. Додатково впроваджено схеми фільтрації високочастотних перешкод, що

генеруються силовими ключами регуляторів обертів двигунів, для захисту аналогового тракту металодетектора. Наземна станція керування представляє собою другий, «туманний» (Fog Computing) рівень архітектури, реалізований на базі високопродуктивної робочої станції з дискретним графічним прискорювачем. Обмін даними з бортовим сегментом здійснюється через виділений радіоканал Wi-Fi [34] у топології «точка-точка», що забезпечує достатню ширину смуги пропускання. Саме тут розгортаються основні алгоритми машинного навчання. Як базовий детектор об'єктів використовується згортова нейронна мережа архітектури YOLOv8 у модифікації nano, яка демонструє оптимальне співвідношення швидкості інференсу та точності детектування малих об'єктів. Результатом роботи моделі є лог детектування (рис. 2.5) з покадровим описом виявлених об'єктів та впевненості розпізнавання.

```

4 Input video: testing_examples/trim_mines_1.mov
5 Output video: LANDMINE_DETECTION_trim_mines_1.mp4
6 Total detections: 852
7
8 DETECTION DETAILS:
9 -----
10 Detection #1:
11   Frame: 10
12   Time: 0.33s
13   Confidence: 77.4%
14   Location: [np.int64(280), np.int64(1028), np.int64(397), np.int64(1080)]
15
16 Detection #2:
17   Frame: 11
18   Time: 0.37s
19   Confidence: 83.9%
20   Location: [np.int64(294), np.int64(1008), np.int64(422), np.int64(1080)]
21
22 Detection #3:
23   Frame: 12
24   Time: 0.40s
25   Confidence: 88.3%
26   Location: [np.int64(312), np.int64(991), np.int64(447), np.int64(1080)]

```

Рисунок 2.5 – Лог з результатами обробки відео моделлю YOLOv8

Для мінімізації помилок першого роду (хибних спрацьовувань), спричинених візуальним шумом складного фону, впроваджено механізм семантичної верифікації за допомогою візуально-мовних моделей (VLM). Фінальним етапом обробки є сенсорне злиття (sensor fusion), де результати

оптичного детектування корелюються з даними металодетектора та просторовими координатами. Це дозволяє формувати комплексні карти загроз, де кожна виявлена аномалія підтверджується кількома фізичними принципами виявлення. Хмарний рівень системи (рис. 2.6) функціонує як середовище для паралельної обчислювальної обробки, консолідації результатів моніторингу та забезпечення оперативного зворотного зв'язку.

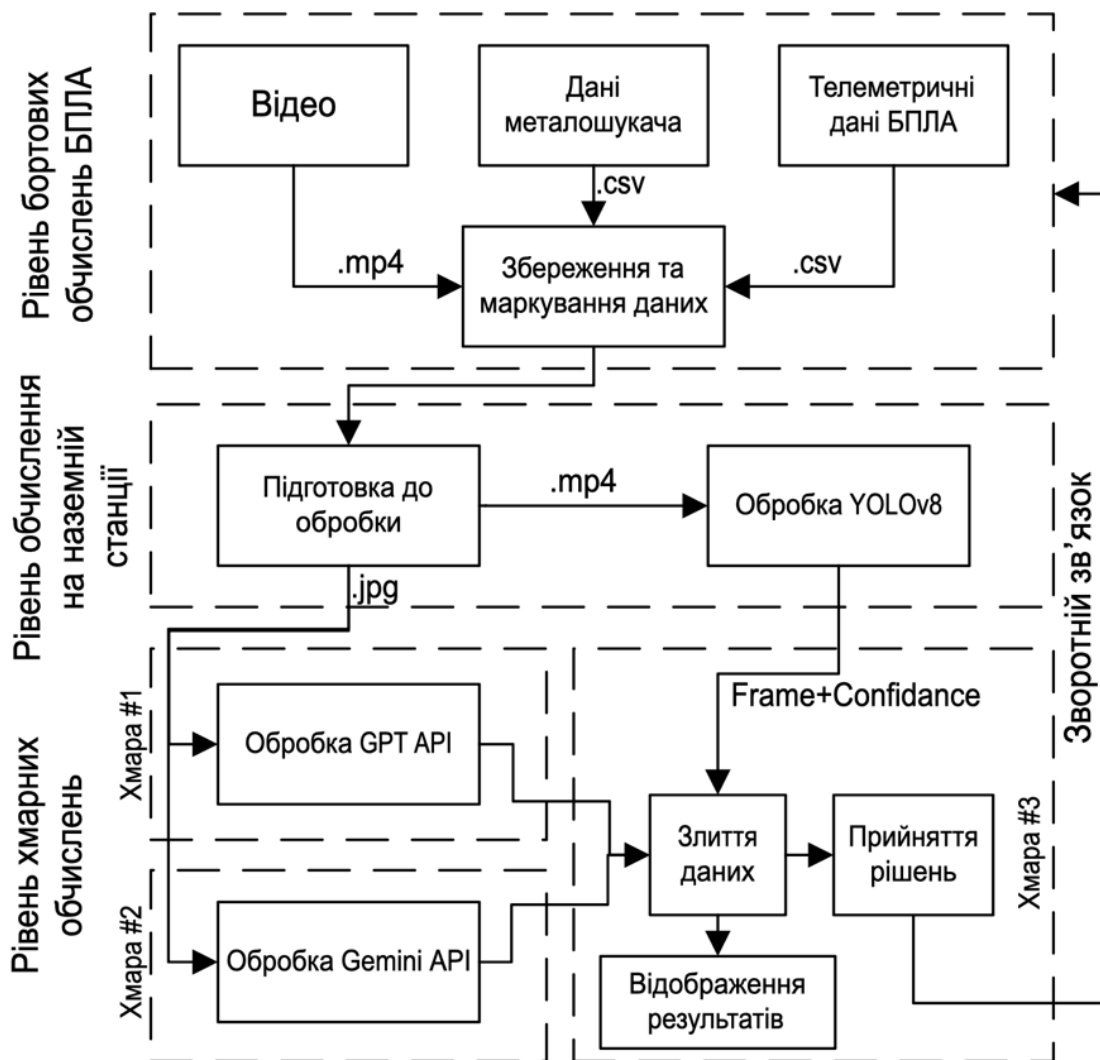


Рисунок 2.6 – Структурна схема багаторівневої обробки даних

Безпосередній аналіз відеопотоку в реальному часі здійснюється на потужностях наземної станції керування із застосуванням нейромережових алгоритмів архітектури YOLO [35], після чого попередньо оброблені масиви даних транслуються у хмарні сервіси. Такий підхід дозволяє реалізувати

розподілену архітектуру обчислень, де хмарні ресурси забезпечують поглиблений аналіз та візуалізацію результатів, а також формують керуючі впливи для динамічної корекції параметрів місії за необхідності. Комунікаційна взаємодія компонентів побудована на принципах відмовостійкості: для трансляції відеоданих застосовуються протоколи реального часу RTSP/RTP або адаптивний UDP-транспорт, обмін телеметрією здійснюється через бінарний протокол MAVLink, а інтеграція з хмарними сервісами базується на захищеному REST API. Безпека функціонування комплексу забезпечується багаторівневою системою сторожових таймерів (watchdogs) та ієрархією пріоритетів команд, що гарантує автоматичне повернення БПЛА до точки старту із збереженням даних у локальному буфері у випадку критичних порушень каналів зв'язку.

## **2.2 Обґрунтування вибору та конструктивно-функціональна реалізація безпілотної платформи гексакоптерного типу**

Розробка високоефективної системи дистанційного зондування та виявлення вибухонебезпечних предметів вимагає комплексного та системного підходу до вибору носія сенсорного обладнання, оскільки фізичні та аеродинамічні характеристики платформи безпосередньо визначають якість вхідних даних для алгоритмів комп'ютерного зору, стабільність роботи геофізичних датчиків та загальну надійність виконання місії в складних польових умовах. У контексті специфічних задач гуманітарного розмінування та інспекції об'єктів критичної інфраструктури, де необхідна висока точність просторового позиціонування, здатність до тривалого зависання над об'єктом інтересу та мінімізація вібраційного впливу на чутливі вимірювальні прилади, вибір аеродинамічної схеми мультироторного типу є найбільш обґрунтованим технічним рішенням. Серед широкого спектра існуючих мультироторних архітектур схема гексакоптера, що передбачає використання шести незалежних гвинтомоторних груп, забезпечує оптимальний баланс між

корисною вантажопідйомністю, показниками енергоефективності, габаритними розмірами та, що є критично важливим аспектом для автономних польових місій, відмовостійкістю силової установки.

*Аналіз аеродинамічної схеми та конструктивних особливостей носія.* Вибір шестироторної схеми, на відміну від класичних квадрокоптерів, зумовлений нагальною необхідністю забезпечення надлишкової тяги та збереження керованості в умовах можливих технічних відмов під час виконання відповідальних завдань. У системах квадрокоптерного типу втрата одного двигуна або критичне пошкодження пропелера неминуче призводить до миттєвої втрати стабілізації, перекидання апарата та аварії, що неприпустимо при роботі з вартісним обладнанням над потенційно небезпечними ділянками. Гексакоптер, завдяки наявності шести векторів тяги, володіє здатністю зберігати керований політ або, як мінімум, виконати безпечну аварійну посадку навіть при повному виході з ладу однієї силової установки. Така живучість досягається завдяки алгоритмам перерозподілу крутного моменту та підйомної сили між п'ятьма справними моторами, що дозволяє польотному контролеру компенсувати виниклу асиметрію тяги та утримувати горизонтальне положення платформи [36]. Для системи, що несе на борту комплексне корисне навантаження у вигляді лазерних сканерів, камер високої роздільної здатності, спеціалізованих обчислювачів та металодетекторів, такий рівень надійності є обов'язковою вимогою технічного завдання.

Базовою конструктивною платформою для побудови носія було обрано раму моделі Tarot 810, виготовлену з високоякісного вуглецевого волокна. Використання композитного карбонового матеріалу забезпечує надзвичайно високу питому міцність та жорсткість конструкції на кручення при збереженні мінімальної власної маси, що є критичним фактором для максимізації доступного польотного часу та корисної вантажопідйомності. Діагональний розмір рами, що становить 810 міліметрів по осях двигунів, дозволяє розмістити пропелери великого діаметра, що значно підвищує ефективність

рушія за рахунок зниження питомого навантаження на площу обмахуваного диска. Конструкція рами передбачає механізм складання променів, що значно спрощує логістику та транспортування комплексу безпосередньо до місця проведення робіт, дозволяючи оперативно розгортати систему в польових умовах без застосування спеціалізованого інструменту. Важливим інтегрованим елементом конструкції є система вібророзв'язки, яка механічно фільтрує високочастотні коливання, що генеруються гвинтомоторними групами, запобігаючи їх передачі на чутливі інерціальні вимірювальні модулі автопілота та оптичну систему камери. Ефективна віброізоляція безпосередньо впливає на якість отриманого відеоматеріалу, зменшуючи прояви ефекту «rolling shutter» та розмиття зображення, що, у свою чергу, спрощує роботу алгоритмів автоматичної детекції об'єктів [32] та підвищує точність класифікації.

Для забезпечення необхідного просторового кліренсу під фюзеляжем апарата, де згідно з компоновальною схемою розміщується гіростабілізований підвіс з камерою та виносна рамка індукційного металодетектора, застосовано посилене шасі типу Tarot T810/T960. Збільшена висота стійок шасі та їх тринога конфігурація з широкою базою гарантують геометричну стабільність апарата під час процедур зльоту та посадки на непідготовлених ґрунтових майданчиках, надійно захищаючи вартісне та чутливе сенсорне обладнання від механічних пошкоджень, ударів та контакту з високою рослинністю. Розміщення центру мас апарата було ретельно оптимізовано шляхом компоновання внутрішніх модулів таким чином, щоб забезпечити нейтральну стійкість та рівномірне навантаження на всі шість двигунів, що сприяє максимально ефективній роботі системи автоматичної стабілізації та зменшує енерговитрати на балансування.

*Силова установка та підсистема енергозабезпечення.* Енергетична ефективність виконання місії та загальна тривалість автономного польоту значною мірою залежать від параметрів та узгодженості компонентів силової установки. У розробленій роботизованій платформі застосовано



високоєфективні безколекторні електродвигуни моделі Tarot 4114-11 з коефіцієнтом обертів на вольт KV320. Низьке значення параметра KV вказує на те, що дані двигуни оптимізовані для створення високого крутного моменту при відносно низьких кутових швидкостях обертання, що дозволяє використовувати пропелери великого діаметра та кроку. У поєднанні з двигунами використано легкі та жорсткі карбонові пропелери типорозміру 1555, що забезпечує високу аеродинамічну ефективність рушія, оскільки значна жорсткість вуглецевого волокна запобігає небажаній деформації [37] та флаттеру лопатей під навантаженням, зберігаючи розрахунковий оптимальний кут атаки та геометрію профілю в усьому діапазоні робочих режимів.

Керування фазами електродвигунів здійснюється за допомогою спеціалізованих цифрових регуляторів швидкості (ESC) моделі Hobbywing XRotor 40A. Ці регулятори характеризуються надзвичайно швидким відгуком на керуючі сигнали автопілота, що є критично важливим для забезпечення динамічної стабілізації платформи в умовах турбулентності атмосфери та поривчастого вітру. Значний запас по робочому струму в 40 Ампер забезпечує надійну та довготривалу роботу силової системи навіть при пікових навантаженнях, що виникають під час різких маневрів або інтенсивного набору висоти, запобігаючи перегріву напівпровідникових ключів. Використання спеціалізованого мікропрограмного забезпечення регуляторів дозволяє оптимізувати таймінг комутації обмоток двигуна, підвищуючи загальний коефіцієнт корисної дії силового приводу та плавність ходу.

Енергетичне забезпечення комплексу базується на використанні сучасної літій-іонної акумуляторної батареї конфігурації 6S3P, яка зібрана з високострумівих елементів Samsung 50S типорозміру 21700. Загальна номінальна напруга батареї становить 22.2 Вольта, а сумарна ємність досягає 15000 мА/год, що забезпечує значний запас енергоємності на рівні близько 333 Вт·год. Вибір літій-іонної технології замість традиційних для авіамоделізму літій-полімерних (LiPo) акумуляторів є стратегічним рішенням, зумовленим

суттєво вищою питомою густиною енергії Li-Ion елементів, що дозволяє значно знизити загальну вагу джерела живлення при збереженні тієї ж корисної ємності. Високовольтна конфігурація 6S забезпечує підвищену напругу живлення силового контуру, що дозволяє зменшити робочі струми при тій же потужності та, відповідно, мінімізувати резистивні втрати енергії в провідниках і нагрів комутаційних елементів.

Розподіл електричної енергії між численними компонентами системи здійснюється через спеціалізовану плату розподілу живлення Holybro PM07 V2.4. Цей інтегрований модуль виконує не лише функцію фізичної комутації силових ліній від акумулятора до регуляторів обертів, але й містить прецизійні вбудовані шунтові датчики струму та дільники напруги, які в реальному часі передають телеметричні дані про енергоспоживання до польотного контролера. Наявність такої телеметрії дозволяє реалізувати на рівні автопілота складні алгоритми оцінки залишку заряду, розрахунку доступного часу польоту та впровадити багаторівневі системи захисту від глибокого розряду акумулятора, що підвищує безпеку експлуатації. Для живлення бортової цифрової електроніки, включаючи одноплатний комп'ютер Raspberry Pi та допоміжні периферійні пристрої, використовується окремий високоефективний імпульсний стабілізатор напруги (UBEC), що забезпечує надійну гальванічну розв'язку чутливих цифрових кіл та фільтрацію високочастотних завад, що генеруються силовим контуром двигунів.

*Навігаційний комплекс та архітектура системи низькорівневого керування.* Ядром системи автоматичного керування польотом виступає сучасний польотний контролер Pixhawk 6C апаратна архітектура якого побудована на базі високопродуктивного 32-бітного мікроконтролера STM32H7 з обчислювальним ядром ARM Cortex-M7 [38]. Цей процесор забезпечує необхідну обчислювальну потужність для виконання в реальному часі складних математичних алгоритмів фільтрації сенсорних даних, розрахунку навігаційного рішення та формування керуючих впливів на виконавчі механізми з частотою оновлення циклу керування до кількох

кілогерц. Архітектура Pixhawk 6C передбачає використання схеми потрібного апаратного резервування інерціальних датчиків (акселерометрів та гіроскопів), що працюють на різних фізичних принципах та мають різні діапазони вимірювань і частотні характеристики. Така надлишковість дозволяє реалізувати на програмному рівні надійні алгоритми мажоритарного голосування та перехресної перевірки даних для оперативного виявлення та програмного виключення збійних сенсорів, що значно підвищує надійність [39] та точність оцінки кутової орієнтації апарата у просторі навіть в умовах сильних вібрацій.

Для високоточного визначення глобальних географічних координат та вектора швидкості руху використовується сучасний GPS-модуль Holybro M9N, який підтримує одночасний багатоканальний прийом сигналів від чотирьох глобальних супутникових систем навігації (GPS, GLONASS, Galileo, BeiDou). Мультисистемність приймача дозволяє кардинально збільшити кількість одночасно видимих супутників, покращуючи геометричний фактор точності (PDOP) та забезпечуючи надійність позиціонування в складних умовах часткового затінення небосхилу елементами рельєфу, будівлями або густою рослинністю. Модуль також містить інтегрований цифровий магнітний компас, що забезпечує незалежне від GPS визначення курсу апарата відносно магнітного полюса Землі.

Критично важливим елементом навігаційної системи, особливо в контексті задач низьковисотного моніторингу та сканування поверхні, є лазерний далекомір (лідар) ST VL53L1X, що працює за фізичним принципом вимірювання часу прольоту світлового імпульсу (Time-of-Flight, ToF). Цей оптичний сенсор забезпечує прецизійне вимірювання істинної висоти польоту над фізичною поверхнею рельєфу (AGL – Above Ground Level) у діапазоні до 4 метрів з міліметровою точністю, незалежно від атмосферного тиску. Інтеграція лідара в контур керування дозволяє реалізувати високоефективний режим автоматичного огинання рельєфу (terrain following), утримуючи стабільну відстань до ґрунту, що є абсолютно необхідною умовою для

коректної роботи індукційного металодетектора, чутливість якого критично залежить від висоти, та отримання оптичних зображень з постійним масштабом і роздільною здатністю. Дані від лідара безпосередньо інтегруються в загальний розширений фільтр Калмана (EKF) автопілота, динамічно коригуючи дрейф барометричного висотоміра.

Організація каналів зв'язку з наземною станцією керування та оператором реалізована через кілька незалежних радіоінтерфейсів для забезпечення надійності та функціонального розділення потоків даних. Для передачі команд безпосереднього ручного керування та прийому базової телеметрії використовується сучасна система дальнього зв'язку ExpressLRS (ELRS) на частоті 2.4 ГГц, що характеризується наднизькою затримкою передачі пакетів та високою завадостійкістю завдяки використанню технології LoRa. Додатковий виділений канал двосторонньої цифрової телеметрії та керування місією реалізовано на базі радіомодемів Holybro SiK, що працюють у частотному діапазоні 433 МГц та забезпечують стабільний обмін даними високорівневого протоколу MAVLink на відстанях до кількох кілометрів. Візуальний контроль польоту від першої особи здійснюється через окрему аналогову систему відеозв'язку (FPV) на частоті 5.8 ГГц, що надає оператору зображення з курсової камери з мінімальною апаратною затримкою, дозволяючи адекватно оцінювати обстановку в реальному часі незалежно від стану цифрових каналів передачі даних та роботи системи комп'ютерного зору [40].

Таким чином, обґрунтована та реалізована апаратна конфігурація гексакоптера являє собою комплексну та збалансовану інженерну систему, де кожен окремий компонент підібрано з урахуванням суворих вимог надійності, енергетичної ефективності та функціональної достатності для виконання складних місій автономного пошуку. Гармонійне поєднання стабільної аеродинамічної платформи, енергоємного джерела живлення та високоточної навігаційної авіоніки створює необхідний технологічний фундамент для

подальшої інтеграції інтелектуальних систем обробки даних та алгоритмів автоматичного адаптивного керування місією.

### 2.3 Динамічна модель гексакоптера та рівняння руху в системі керування

Розробка системи автоматичного керування для задач просторового моніторингу вимагає побудови математичної моделі, яка враховує кінематичні та динамічні особливості гексакоптера. Гексакоптер розглядається як тверде тіло симетричної структури з шістьма обертовими двигунами. Для опису руху використовуються дві основні системи координат:

1. Інерціальна система координат (Earth-Fixed Inertial Frame,  $F_E$ ): Пов'язана з поверхнею Землі. Використовується навігаційна система NED (North-East-Down), де вісь  $X_E$  спрямована на Північ,  $Y_E$  – на Схід, а  $Z_E$  – вертикально вниз (до центру Землі). Положення центру мас гексакоптера в цій системі описується вектором  $\xi = [x, y, z]^T$  [41].
2. Зв'язана система координат (Body-Fixed Frame,  $F_B$ ): Жорстко зв'язана з корпусом апарата, початок координат  $O_B$  знаходиться в центрі мас (CoG). Вісь  $X_B$  спрямована вперед (у напрямку руху),  $Y_B$  – праворуч,  $Z_B$  – вниз (перпендикулярно площині роторів). Лінійні швидкості в цій системі позначимо як  $v_B = [u, v, w]^T$ , а кутові швидкості – як  $\omega_B = [p, q, r]^T$  [42].

Для математичного опису просторової орієнтації корпусу гексакоптера відносно інерціальної системи координат застосовано формалізм кутів Ейлера: крен ( $\phi$ ), тангаж ( $\theta$ ) та рискання ( $\psi$ ). Матриця повороту  $R$  від зв'язаної до інерціальної системи координат для послідовності поворотів  $Z \rightarrow Y \rightarrow X$  має вигляд [43]:

$$R = \begin{bmatrix} c_\theta c_\psi & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ c_\theta s_\psi & c_\psi s_\theta s_\phi + s_\psi c_\phi & c_\psi s_\theta c_\phi - s_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \quad (2.1)$$

де  $c_x = \cos(x)$ ,  $s_x = \sin(x)$ .

Зв'язок між кутовими швидкостями тіла  $\omega_B$  та швидкістю зміни кутів Ейлера  $\dot{\eta} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$  описується рівнянням  $\omega_B = W \dot{\eta}$ . Матриця  $W$  стає виродженою при  $\theta = \pm 90^\circ$  (коли  $\cos \theta = 0$ ), що призводить до явища «складання рамок» (gimbal lock) і робить обчислення орієнтації некоректним під час вертикальних маневрів [43]. Щоб уникнути цієї проблеми, у сучасних польотних контролерах (зокрема Pixhawk/ArduPilot) орієнтацію задають за допомогою кватерніонів які забезпечують лінійність формулювання, обчислювальну ефективність та відсутність сингулярностей. Нижче наведено виведення динамічної моделі гексакоптера з використанням формалізму Ньютона-Ейлера та кватерніонів.

Сили, що діють на гексакоптер, створюються в системі координат, зв'язаній з корпусом (body frame), тоді як прискорення зазвичай розглядаються в інерціальній системі координат. Рівняння поступального руху [44] має вигляд:

$$m\dot{V}_B + v \times (mV_B) = R^T G + T \quad (2.2)$$

де:

$m$  – маса гексакоптера;

$\dot{V}_B$  – вектор лінійної швидкості у зв'язаній системі координат;

$R^T G$  – вектор сили тяжіння, трансформований у зв'язану систему координат;

$T$  – вектор сумарної тяги роторів (спрямований вздовж осі  $Z$  зв'язаної системи).

В інерціальній системі координат рівняння руху набуває вигляду:

$$m\ddot{\xi} = G + R^T T \quad (2.3)$$

де  $\ddot{\xi} = [\ddot{x}, \ddot{y}, \ddot{z}]^T$  – вектор лінійного прискорення. При використанні класичних кутів Ейлера матриця повороту  $R$  призводить до громіздких тригонометричних виразів (2.3). Натомість, використання одиничного кватерніона  $q = [q_0, q_1, q_2, q_3]^T$  дозволяє записати рівняння поступального руху у компактній алгебраїчній формі без тригонометричних функцій.

Згідно з відповідними роботами, диференціальні рівняння поступальної динаміки гексакоптера у кватерніонному поданні записуються як:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = -g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \frac{T}{m} \begin{bmatrix} 2(q_0 q_2 + q_1 q_3) \\ 2(q_2 q_3 - q_0 q_1) \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (2.4)$$

Ця система рівнянь описує зміну координат центру мас апарата під дією гравітації та вектора тяги, орієнтація якого визначається поточним кватерніоном.

Динаміка обертального руху описується рівнянням Ейлера для твердого тіла. У зв'язаній системі координат зовнішній момент  $\tau_B = [\tau_\phi, \tau_\theta, \tau_\psi]^T$  врівноважує моменти від кутового прискорення інерції  $J\dot{v}$ , відцентрових сил  $v \times (Jv)$  та гіроскопічних ефектів  $J_{gyro}$ :

$$J_b \dot{v} + v \times (J_b v) + J_{gyro} = \tau_B \quad (2.5)$$

де:

$v = [p, q, r]^T$  – вектор кутової швидкості у зв'язаній системі координат;

$J_b = \text{diag}(J_{xx}, J_{yy}, J_{zz})$  – тензор інерції гексакоптера;

$J_{gyro}$  – гіроскопічний момент від обертання пропелерів.

Розв'язуючи це рівняння відносно кутового прискорення  $\dot{v}$ , отримуємо:

$$\dot{v} = J_b^{-1} \left( - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} J_{xx}p \\ J_{yy}q \\ J_{zz}r \end{bmatrix} - J_r \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega_r + \tau_B \right) \quad (2.6)$$

де  $\omega_r$  – залишкова кутова швидкість роторів.

У розгорнутому вигляді для компонент кутового прискорення  $[\dot{p}, \dot{q}, \dot{r}]^T$  це рівняння набуває вигляду:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{(J_{yy}-J_{zz})qr}{J_{xx}} \\ \frac{(J_{zz}-J_{xx})pr}{J_{yy}} \\ \frac{(J_{xx}-J_{yy})pq}{J_{zz}} \end{bmatrix} - J_r \begin{bmatrix} \frac{q}{J_{xx}} \\ -\frac{p}{J_{yy}} \\ 0 \end{bmatrix} \bar{\omega} + \begin{bmatrix} \frac{\tau_\phi}{J_{xx}} \\ \frac{\tau_\theta}{J_{yy}} \\ \frac{\tau_\psi}{J_{zz}} \end{bmatrix} \quad (2.7)$$

Для замикання системи рівнянь необхідно пов'язати похідну кватерніона  $\dot{q}$  з вектором кутової швидкості  $v$  (або  $S_v$  у позначеннях відповідних джерел). Це співвідношення є лінійним:

$$\dot{q}_u = \frac{d}{dt}(S_v) \quad (2.8)$$

Об'єднуючи рівняння поступального руху (2.8) та кінематику обертання, отримуємо повну систему диференціальних рівнянь, що описує динаміку гексакоптера:

$$\begin{cases} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = -g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \frac{T}{m} \begin{bmatrix} 2(q_0q_2 + q_1q_3) \\ 2(q_2q_3 - q_0q_1) \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \\ \ddot{q}_u = \frac{d}{dt}(S_v) \end{cases} \quad (2.9)$$

Використання цієї моделі (2.9) у системі керування дозволяє уникнути проблем із сингулярністю при виконанні складних маневрів (наприклад,



різких розворотів або польотів з великими кутами нахилу), забезпечуючи стабільну роботу автопілота у всьому польотному діапазоні. Це є критично важливим для задач моніторингу на малих висотах, де стабільність платформи безпосередньо впливає на якість даних сенсорів.

У цій роботі розглядаються пропелери типу 1555 (15") та мотори 320KV. Квадратична залежність є важливою для формування матриці мікшування, оскільки на практиці керування двигунами здійснюється через величини, пропорційні  $\Omega_i^2$  (або через ШІМ-сигнали, що наближено відповідають цьому закону). Матриця мікшування (Mixing Matrix) є математичним ядром, яке перетворює віртуальні команди керування (тяга, крен, тангаж, ристання), згенеровані PID-регуляторами автопілота, у фізичні команди для шести двигунів. Оскільки гексакоптер має 6 виконавчих механізмів, а керується лише по 4 ступенях вільності (висота + 3 кути), система є пере-визначеною (over-actuated). Це дає можливість оптимізувати розподіл навантаження та забезпечує стійкість до відмови двигуна. Згідно з документацією ArduPilot та описом платформи, для гексакоптерів найчастіше використовується конфігурація Неха X. У цій схемі перший двигун розташований не на поздовжній осі (як у «плюс» конфігурації), а зміщений на кут. Для симетричного гексакоптера кутова відстань між променями становить  $60^\circ$  [45]. У ArduPilot (та Pixhawk 6C) стандартний порядок нумерації моторів та напрямок їх обертання для Неха X є наступним (погляд зверху):

- **Motor 1:** Спереду-справа (Front-Right), обертання проти годинникової стрілки (CCW). Кут  $\gamma_1 = -30^\circ$  (або  $330^\circ$ ).
- **Motor 2:** Ззаду-справа (Rear-Right), обертання за годинниковою стрілкою (CW). Кут  $\gamma_2 = -90^\circ$  (або  $270^\circ$ ).
- **Motor 3:** Ззаду-зліва (Rear-Left), обертання проти годинникової стрілки (CCW). Кут  $\gamma_3 = -150^\circ$  (або  $210^\circ$ ).
- **Motor 4:** Спереду-зліва (Front-Left), обертання за годинниковою стрілкою (CW). Кут  $\gamma_4 = 30^\circ$ .

- **Motor 5:** Зліва (Left), обертання проти годинникової стрілки (CCW).  
Кут  $\gamma_5 = 90^\circ$ .
- **Motor 6:** Справа (Right), обертання за годинниковою стрілкою (CW).  
Кут  $\gamma_6 = -90^\circ$ .

Проте, класична схема ArduPilot для Неха Х визначає розташування моторів як вказано на зображенні 2.7:

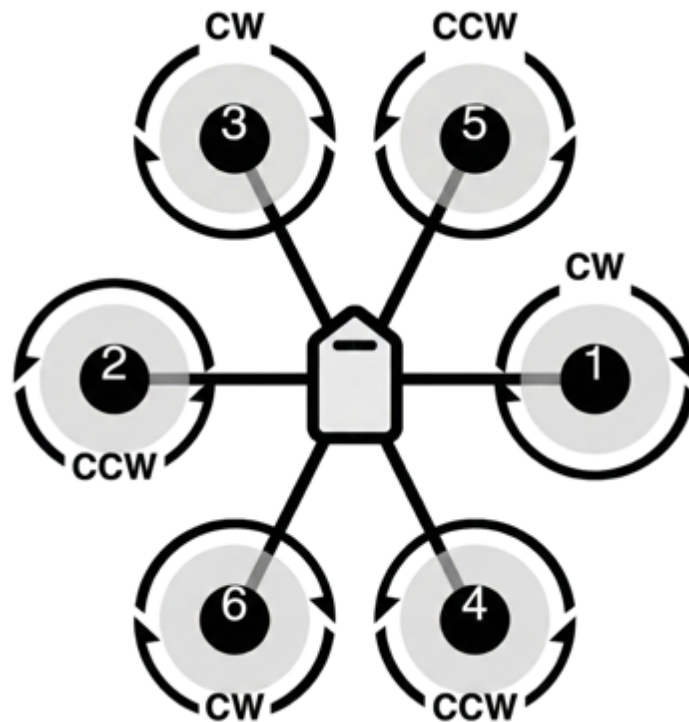


Рисунок 2.7 – Діаграма розташування моторів БПЛА

Звернемося до вихідного коду бібліотеки AP\_MotorsMatrix.cpp, де визначаються фактичні коефіцієнти впливу (Roll, Pitch, Yaw factors).

Загальне рівняння мікшування для двигуна  $i$  має вигляд:

$$Out_i = Thrust + R_i \cdot Roll + P_i \cdot Pitch + Y_i \cdot Yaw \quad (2.10)$$

де  $R_i, P_i, Y_i$  – коефіцієнти матриці.

Моменти, що діють на корпус, визначаються геометрією. Нехай  $L$  – довжина променя. Крен (Roll,  $\tau_\phi$ ): залежить від проекції плеча сили на вісь  $Y_B$  (відстань від осі  $X_B$ ):

$$\tau_\phi = L \sum T_i \sin(\gamma_i) \quad (2.11)$$

(або  $\cos$ , залежно від відліку кута). У ArduPilot прийнято нормовані фактори. Для симетричного Неха Х мотори 1 і 4 (спереду) мають плече  $\approx L \sin(30^\circ) = 0.5L$ . Мотори 5 і 6 (бокові) мають плече  $\approx L \sin(90^\circ) = 1.0L$ .

Тангаж (Pitch,  $\tau_\theta$ ): залежить від проекції на вісь  $X_B$ :

$$\tau_\theta = L \sum T_i \cos(\gamma_i) \quad (2.12)$$

Мотори 1 і 4 мають плече  $\approx L \cos(30^\circ) \approx 0.866L$ . Мотори 5 і 6 знаходяться на осі  $Y$  (або  $X$ , залежно від визначення), тому їх внесок у тангаж може бути нульовим або максимальним.

Таблиця 2.1 – Стандартна таблиця коефіцієнтів для ArduPilot Неха Х

| Motor # | Позиція<br>(приблизно) | Roll Factor<br>( $R_i$ ) | Pitch Factor<br>( $P_i$ ) | Yaw Factor<br>( $Y_i$ ) | Напрямок |
|---------|------------------------|--------------------------|---------------------------|-------------------------|----------|
| 1       | Front Right            | -0.5                     | 0.866                     | -1                      | CCW      |
| 2       | Rear Left              | 0.5                      | -0.866                    | -1                      | CW       |
| 3       | Front Left             | 0.5                      | 0.866                     | 1                       | CW       |
| 4       | Rear Right             | -0.5                     | -0.866                    | 1                       | CCW      |
| 5       | Right                  | -1                       | 0                         | 1                       | CW       |
| 6       | Left                   | 1                        | 0                         | -1                      | CCW      |

Знаки Roll/Pitch залежать від системи координат (NED vs ENU). У ArduPilot позитивний Roll – нахил вправо (опускання правого борту), позитивний Pitch – ніс вгору. Yaw Factor залежить від напрямку обертання: CW двигуни створюють CCW реактивний момент, тому для повороту за годинниковою стрілкою ( $Yaw > 0$ ) їх потрібно сповільнити, а CCW

прискорити, або навпаки. Матриця мікшування  $M_{mix}$  (обернена до фізичної матриці ефективності) фактично реалізована як набір ваг, на які множаться виходи PID-регуляторів:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} = \begin{bmatrix} 1 & -0.5 & 0.866 & -1 \\ 1 & 0.5 & -0.866 & -1 \\ 1 & 0.5 & 0.866 & 1 \\ 1 & -0.5 & -0.866 & 1 \\ 1 & -1.0 & 0 & 1 \\ 1 & 1.0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \delta_{thr} \\ \delta_{roll} \\ \delta_{pitch} \\ \delta_{yaw} \end{bmatrix} \quad (2.13)$$

Оскільки діапазон керування двигунами обмежений ( $u_{min}$ ,  $u_{max}$ ), просте лінійне перетворення може призвести до виходу за межі (насичення). У системі ArduPilot реалізовано алгоритм пріоритезації осей. Якщо сума необхідних керуючих впливів перевищує можливості моторів, система зменшує впливи в такому порядку:

- Thrust (Тяга): жертвується першою, якщо потрібно зберегти орієнтацію (щоб дрон не перекинувся).
- Yaw (Рискання): жертвується наступним, оскільки тимчасова втрата курсу менш критична за втрату горизонту.
- Roll/Pitch (Крен/Тангаж): мають найвищий пріоритет для забезпечення стабільності платформи.

Це реалізується через динамічне масштабування вихідних сигналів перед подачею на ESC.

## 2.4 Апаратна реалізація та інтеграція імпульсного металошукача Tracker-2 у сенсорний комплекс

Імпульсний металошукач Tracker-2 використовується як додатковий сенсорний канал, що доповнює візуальне детектування та підвищує достовірність локалізації небезпечних об'єктів. Це глибинний PI-детектор

українського виробництва, розрахований на виявлення крупних і середніх металевих цілей на значних глибинах (до  $\sim 4$  м залежно від розміру, орієнтації та провідності цілі). Принцип дії – класичний для імпульсних систем: управляючий каскад формує короткі потужні струмові імпульси в пошуковій котушці рамкового типу ( $\approx 1 \times 1$  м). Після вимкнення збудження котушка переходить у режим «самозагасання», а в присутності металевих об'єктів у зоні дії з'являються вихрові струми, що змінюють часовий профіль згасання магнітного поля. Електроніка детектора відсікає «сліпу ділянку» одразу після імпульсу (для гасіння власних перехідних процесів у котушці), а далі вимірює амплітуду/площу затухаючого сигналу в одній або декількох часових «вікнах». Ґрунтові ефекти (мінералізація) компенсуються схемами балансування ґрунту, в Tracker-2 вони реалізовані апаратно-програмно, що дозволяє стабілізувати базову лінію та тримати чутливість на складних ґрунтах. Детектор працює без дискримінатора (реагує на всі метали), що з точки зору гуманітарного розмінування є прийнятним і навіть бажаним – критично важлива чутливість, а відсікання «не цілей» виконується вже на рівні просторово-часового злиття з відео.

Апаратно Tracker-2 [46] побудований на двосторонній металізованій друкованій платі габаритами  $\sim 82 \times 52$  мм з сучасною елементною базою (силовий ключ, драйвер котушки, захисти, малошумні підсилювачі/вибіркі-зберігачі). Логіка керування – на мікроконтролері ATTINY2313A-PU (рис 2.8) з прошивкою, оптимізованою під глибинний пошук (типово V4.3 для великих рамкових котушок). Передбачено кілька програм/профілів (до 36), що коригують тривалість імпульсу, частоту повторення, вікна відбору та коефіцієнти підсилення – це дає змогу адаптуватися до ґрунтів і цільових розмірів. Існують два виконання: LED-версія (візуальна індикація спрацювань) і версія з індикатором (у лінійці Tracker-3), що розширює відображення станів. Живлення – від стабільного 12 В джерела, що добре узгоджується з живленням корисного навантаження БПЛА, що забезпечується підключенням напряду до батареї дрону через регулятор напруги (UBEC)

HENGE 4A. Споживання імпульсного каналу носить пульсуючий характер і залежить від налаштувань частоти/тривалості імпульсів та індуктивності котушки.

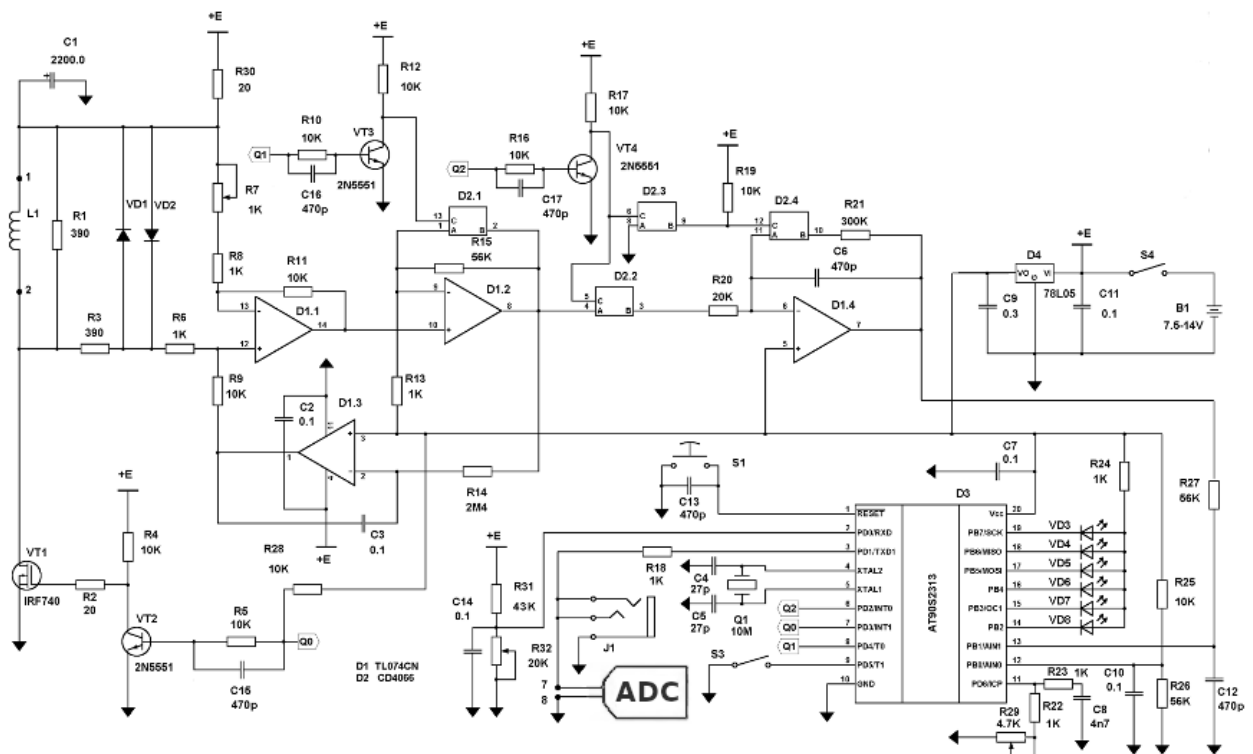


Рисунок 2.8 – Схема електрична принципова металошукача Tracker-2

На дроні рамкову котушку ( $\sim 1 \times 1$  м) змонтовано під [47] днищем на віброізовованій підвісці з жорстким геометричним позиціонуванням відносно центра мас. Кріплення забезпечує стабільний кліренс і повторювану висоту відносно ґрунту, а також розв'язку від вібрацій гвинтомоторних груп. У типовому профілі польоту для металодетектора висота AGL підтримується в коридорі  $\sim 0.3-0.7$  м (Regime «low-alt» з обмеженими швидкостями та лагідною кінематикою), що балансує безпеку польоту й співвідношення сигнал/шум. Для зйомки RGB (візуального каналу) застосовується окремий профіль на більшій висоті ( $\approx 2-3$  м) з більшим покриттям, потім результати двох проходів зводяться на рівні наземної станції під час злиття даних. Силові кабелі котушки та лінії датчиків екранізовано, траси рознесено від ESC і силових

шин, земля – «зіркою», на входах стоять ферити/RC-ланцюги, що різко зменшує ЕМ-завади від силової частини мультикоптера.

Інтеграція з бортовою обчислювальною системою побудована з пріоритетом на синхронізацію часу та варіативність форматів сигналів. Tracker-2 може надавати: *аналоговий вимірювальний* сигнал (після детекторного/вибіркового каскаду) – для його оцифрування використовується зовнішній АЦП із високою розрядністю і малим власним шумом (ADS1115/ADS1015 по І<sup>2</sup>C або MCP3008 по SPI) у парі з Raspberry Pi 4B; *цифрові події/стани* – зчитування з TTL-выводів індикації (LED/«pinpoint»/«threshold») на GPIO Pi через рівнетвір (5V→3.3V) або оптоізоляцію; *послідовний потік* (за наявності сервісного/програмного UART від ATTINY на платі) – у такій конфігурації подається телеметрія станів/профілів і числові ознаки (сума у вікні, лічильники спрацювань тощо). Усі три варіанти підтримані у конвеєрі: при аналоговому вводі Raspberry Pi ініціює згортання даних у короткі часові вікна, що відкриваються в заданому зсуві від імпульсу, з подальшою фільтрацією/нормалізацією. При «подійній» інтеграції використовується переривання GPIO і часова мітка високої роздільності, а при UART-режимі – парсинг пакетів і синхронізація через GPS-час. Синхронізацію з відео/телеметрією забезпечує загальний часовий контур GPS-час із модуля M9N через Pixhawk роздається на Raspberry Pi (NTP/chrony/PPS), тож кожен семпл металодетектора має UTC-мітку і може бути точно зіставлений з кадром і позицією/орієнтацією БПЛА.

З точки зору експлуатації важливими є режимні налаштування та калібрування. По-перше, здійснюється балансування ґрунту (авто/ручне), після чого фіксується базова лінія й обирається програма з 36 доступних під поточний фон і тип очікуваних цілей. По-друге, визначається висота польоту для металодетектора (чим нижче – тим вищий SNR, але з урахуванням безпечної кінематики), швидкість проходу та ширина смуги огляду ( $\approx$  ширина рамки + запас). По-третє, налаштовуються інтерфейси: тип виводу (аналог/події/UART), рівні сигналів, полярність, тактова частота опитування

АЦП, пороги тривоги. На практиці корисно застосовувати подвійну ознаку: безперервна аналогова «інтенсивність» у вікні + бінарна подія «перевищення порогу» – це спрощує як первинну фільтрацію, так і подальше злиття з відеодетекціями. Масово-інженерний підхід – логувати «сирі» вікна сигналів і потім виконувати офлайнову нормалізацію/дрейф-корекцію, у той час як у польових місіях із обмеженням каналу використовується передача лише зведених ознак та подій, а «сирі» семпли буферизуються на Raspberry Pi.

Tracker-2 демонструє високу чутливість до об'ємних цілей із гарною провідністю/магнітною проникністю [48], як наслідок – зменшення розміру цілі і зростання висоти польоту SNR спадає. На працездатність впливають завади (лінії електропередач, арматура, транспорт), вологість і мінералізація ґрунту. Тому консервативне планування передбачає двоступеневий обхід: «широкий» прохід для карти інтенсивності та деталізаційні локальні проходи з меншою висотою/швидкістю у «гарячих» зонах. Подібні імпульсні сенсори широко застосовуються у дуальних системах розмінування – як у ручних комплексах, так і в безпілотних зв'язках «дрон + датчик», що підтверджує доцільність обраної конфігурації для попереднього дистанційного скринінгу.

Для зручності наведемо зведені параметри, релевантні інтеграції з БПЛА та Raspberry Pi (Таб. 2.2).

Таблиця 2.2 – Параметри інтеграції з БПЛА

| Параметр          | Значення / варіанти                                                      |
|-------------------|--------------------------------------------------------------------------|
| Тип               | Імпульсний глибинний металодетектор (Raspberry Pi)                       |
| Глибина виявлення | До ~4 м (залежить від розміру/типу цілі та ґрунту)                       |
| Програми          | До 36 профілів (тривалість/частота імпульсів, вікна відбору, підсилення) |
| Баланс ґрунту     | Авто/ручний, компенсація мінералізації                                   |
| Дискримінація     | Немає (реакція на всі метали)                                            |
| Живлення          | 12 В пост. струм; пульсуюче споживання залежно від налаштувань           |



|                      |                                                                                                      |
|----------------------|------------------------------------------------------------------------------------------------------|
| Плата                | 2-шарова, ~82×52 мм, металізовані отвори; LED або індикаторна версія                                 |
| МК/ПЗ                | ATTINY2313A-PU; прошивка для глибинного пошуку (версія V2.4)                                         |
| Інтерфейси на виході | Аналоговий вимірювальний сигнал; подієві TTL-лінії; (за наявності) UART сервісний                    |
| Інтеграція з Pi      | ADC (ADS1115/ADS1015 I <sup>2</sup> C або MCP3008 SPI); GPIO з перериваннями; UART TTL з рівнетвором |
| Монтаж на БПЛА       | Рамка 1×1 м під днищем на демпферах; екрановані кабелі; «зіркова» земля                              |
| Робоча висота        | Типово ~0.3-0.7 м AGL для металодетекції; зйомка RGB окремо на ~2-3 м                                |

У підсумку, Tracker-2 у складі бортового комплексу забезпечує інформативний, фізично ортогональний канал. Його сигнали легко піддати часовій прив'язці (GPS-UTC → Raspberry Pi) та звести з відеодетекціями і телеметрією. Монтаж у вигляді рамкової котушки під днищем гексакоптера, правильно організоване екранування та вибудовані інтерфейси (ADC/GPIO/UART) дають стабільну, відтворювану якість даних, придатну для просторово-часового злиття даних та побудови теплових карт ризику в рамках запропонованої багаторівневої архітектури.

## **2.5 Одноплатний комп'ютер Raspberry Pi 4B як бортовий обчислювальний модуль системи**

У запропонованому комплексі Raspberry Pi 4 Model B [49] виконує роль бортового фільтра та комутатора даних: він приймає сирий відеопотік із камери по CSI-2, синхронізує його з телеметрією автопілота та сигналами металопукача, робить мінімально необхідну попередню обробку/стиснення і, залежно від стану каналу, відправляє кадри (або лише ROI та журнали) на наземну станцію чи в хмару. Таким чином саме на рівні Pi закладається низька

затримка й економія смуги: легкі операції виконуються локально, важкі – делегуються вище каскадом Edge→Local→Cloud.

Апаратною основою Raspberry Pi 4B слугує система-на-кристалі (SoC) Broadcom BCM2711. Вона об'єднує чотириядерний 64-бітний процесор ARM Cortex-A72 (типова частота 1.5 ГГц, з можливістю «turbo-режиму» до ~1.8 ГГц на деяких ревізіях) та графічне ядро VideoCore VI. Плата використовує пам'ять типу LPDDR4 і доступна в конфігураціях з об'ємом 1, 2, 4 або 8 ГБ ОЗП. Для завдань комп'ютерного зору (CV) практично доцільними є версії на 4-8 ГБ, оскільки такий обсяг необхідний для ефективної буферизації кадрів, організації zero-сору конвеєрів обробки та підтримки паралельних потоків. Комунікаційні можливості представлені бездротовими модулями 802.11ac Wi-Fi (1×1, 2.4/5 ГГц) і Bluetooth 5.0/LE. Важливою перевагою є повноцінний дротовий інтерфейс Gigabit Ethernet, який, на відміну від попередніх поколінь, має окремий MAC/PHY і не обмежується пропускною здатністю шини USB 2.0. Для підключення периферії передбачено два швидкісні порти USB 3.0 (придатні для SSD-накопичувачів, зовнішніх акселераторів або 4G/LTE-модемів) та два порти USB 2.0. Для взаємодії з датчиками та виконавчими пристроями слугує 40-контактний роз'єм GPIO, що підтримує стандартні протоколи (SPI, I<sup>2</sup>C, UART, CAN), ШІМ (PWM) та механізм GPIO-переривань. Окремо виведені порти MIPI: 2-смуговий CSI-2 для підключення камер (зокрема Arducam 64MP) та DSI для сервісного дисплея. Стандартне живлення плати забезпечується джерелом 5В (3А) через роз'єм USB-C. Однак у складі БПЛА застосовується спеціалізоване рішення: стабілізований DC-DC перетворювач із фільтрацією високочастотних перешкод від регуляторів обертів (ESC).

У нашому конвеєрі обробки ключова роль відведена інтегрованому графічному процесору (GPU) та процесору обробки зображень (ISP) VideoCore VI. Вони не лише апаратно обслуговують камерний стек (libcamera/ISP), але й дозволяють через OpenGL-ES 3.x та V4L2 зняти з центрального процесора (CPU) типові «важкі» операції, такі як масштабування

кадрів, кольорові перетворення, легке згладжування, виділення контурів (фільтри Собеля/Шарра), проста компенсація фону (ковзне середнє/медіана) та формування масок руху. Ці операції, реалізовані у вигляді фрагментних шейдерів, функціонують у режимі, близькому до нульового копіювання (DMABUF/zero-copy), завдяки чому навіть при 720p обробка вкладається у кілька мілісекунд на кадр і не конкурує за ресурси з потоками телеметрії. Для стиснення відео задіяний апаратний кодек H.264 (V4L2), що забезпечує стабільне кодування 1080p@30. Додатково, SIMD-розширення NEON (через OpenCV/TFLite) використовуються для дрібних CNN-задач, зокрема для фільтрації «порожніх» кадрів, що різко зменшує трафік та дозволяє попередньо ранжувати дані перед відправкою. Таким чином, на борту залишається лише «легка» логіка, стабільна в межах енергобюджету, тоді як повноцінні детектори (YOLOv8n) виконуються на наземній станції. Інтеграція з сенсорикою побудована безпосередньо на GPIO: сигнали від імпульсного металошукача можуть надходити як аналогові (через зовнішній АЦП по I<sup>2</sup>C або SPI з гнучким налаштуванням опитування), як цифрові події (через GPIO-переривання з міткою часу високої роздільності), або по UART (вимагаючи узгодження рівнів або оптоізоляції). Камера Arducam 64MP [50] підключена по MIPI CSI-2 і керується через libcamera з апаратною стабілізацією ISP. Зв'язок назовні забезпечується через 5 ГГц Wi-Fi (802.11ac) або USB-4G/LTE-модем, що в умовах поганого зв'язку плата надсилає лише батчі ROI та журнали, зберігаючи повні кадри у кільцевому буфері на носії для післяпольотного аналізу. Критично важлива часова узгодженість усіх даних забезпечується від GPS (через Pixhawk 6C/M9N), а сигнали PPS та NTP синхронізують системний годинник Raspberry Pi з UTC, гарантуючи, що кожен кадр і кожна подія від металодетектора мають точну часову мітку для подальшого коректного злиття.

Практичні обмеження та експлуатаційні режими були враховані як на рівні конструкції, так і в програмному забезпеченні. Ключовим аспектом є теплова продуктивність: оскільки Raspberry Pi 4B починає тротлити

(пропускати такти) при досягненні 80°C, у корпусі БПЛА плата оснащена ребристим радіатором. У поєднанні зі зустрічним потоком повітря від пропелерів, достатньо для утримання температури в робочому діапазоні 50-65 °C; для спекотного середовища додатково передбачено малий вентилятор (30×30 мм) із PWM-керуванням. Другим фактором є енергоспоживання, що становить 3-5 Вт у режимі очікування, 5-7 Вт під час обробки 720p-відео та H.264-кодування, і сягає пікових ~8 Вт при активному GPU-шейдингу. Тому лінія живлення спроектована зі значним запасом по струму, оснащена локальними конденсаторами та TVS-захистом. Щодо реальної продуктивності – система стабільно утримує 720p@30 кадрів/с із повним циклом обробки «ISP → шейдери → H.264 → RTP» при затримці ~40-80 мс, а режим 1080p@30 також досяжний за умови помірного використання шейдерів. При цьому задачі повної детекції (на кшталт YOLOv8x) принципово винесені на наземну станцію, тоді як на борту виконуються лише операції фільтрації «порожніх» сцен, генерації ROI та синхронізації потоків.

Підсумовуючи, Raspberry Pi 4B є оптимальним бортовим вузлом першого рівня, що забезпечує достатню продуктивність для попередньої візуальної обробки (GPU/ISP), апаратне H.264-стиснення та гнучку роботу з сенсорами (GPIO/CSI) при низькій затримці. У поєднанні з часовою прив'язкою від GPS та адаптивною політикою передачі даних (повні кадри або ROI/журнали), це створює стабільну візуальну клієнтську частину (Front-End) для подальшої високоточної обробки на наземній станції.

## 2.6 Висновки за розділом 2

Другий розділ дисертаційної роботи присвячено проектуванню, обґрунтуванню вибору елементної бази та конструктивно-функціональній реалізації бортового апаратного комплексу багаторівневої роботизованої платформи просторового моніторингу. На основі проведених інженерних розрахунків та теоретичних досліджень синтезовано кіберфізичну систему,

здатну ефективно функціонувати в умовах невизначеності зовнішнього середовища. В якості базового носія сенсорного обладнання було науково обґрунтовано вибір мультироторної аеродинамічної схеми типу гексакоптер (на базі рами Tarot 810 з карбонового волокна). На відміну від класичних квадрокоптерів, використання шести незалежних векторів тяги забезпечує критично необхідний рівень апаратної надлишковості та відмовостійкості. У разі виходу з ладу однієї гвинтомоторної групи система автоматичної стабілізації здатна перерозподілити крутний момент між справними двигунами, гарантуючи збереження керованості апарата або безпечне виконання аварійної посадки, що є обов'язковою вимогою при роботі з вартісним обладнанням над замінованими територіями.

Для забезпечення коректного функціонування алгоритмів низькорівневого керування була розроблена та уточнена динамічна математична модель гексакоптера. Обертальний та поступальний рух апарата формалізовано з використанням апарату кватерніонів, що дозволило подолати проблему сингулярності (явище «складання рамок» або gimbal lock), характерну для класичних кутів Ейлера при виконанні складних просторових маневрів. Математична модель обертального руху у векторно-матричній формі рівняння Ейлера була розширена додаванням вектора моментів зовнішніх збурень, що враховує вплив приземної турбулентності, поривів вітру та аеродинамічної інтерференції між роторами. Це створило теоретичне підґрунтя для налаштування каскадних PID-регуляторів польотного контролера Pixhawk 6C. Критичним досягненням апаратної реалізації стала інтеграція в єдиний комплекс гетерогенних сенсорних каналів та забезпечення їхньої суворої просторово-часової синхронізації. Центральним обчислювальним вузлом на борту (Edge-рівень) виступає одноплатний мікрокомп'ютер Raspberry Pi 4B, який здійснює захоплення нестисненого відеопотоку з 64-мегапіксельної камери ArduCAM через паралельний інтерфейс MIPI CSI-2. Таке рішення мінімізує навантаження на центральний процесор за рахунок прямого доступу до пам'яті (DMA). Одночасно

мікрокомп'ютер отримує телеметричні дані від автопілота через високошвидкісний UART-інтерфейс за протоколом MAVLink та оцифровує сигнали від індукційного імпульсного металодетектора Tracker-2. Вирішення проблеми синхронізації цих різнорідних потоків досягнуто шляхом використання єдиного еталонного джерела часу від GPS-модуля Holybro M9N (протоколи NTP/chrony та апаратна лінія PPS), що дозволяє прив'язувати кожен відеокادر та вимір магнітного поля до точних просторово-часових координат з мікросекундною точністю.

Таким чином, розроблена апаратна архітектура являє собою збалансовану кіберфізичну платформу. Вона поєднує стабільну аеродинаміку, енергоефективність та високу обчислювальну гнучкість, створюючи надійний фундамент для імплементації інтелектуальних програмних алгоритмів мультимодального аналізу та адаптивного керування в режимі реального часу.

## РОЗДІЛ 3. АЛГОРИТМІЧНО-ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ МОДУЛЮ КОМП'ЮТЕРНОГО ЗОРУ

### 3.1 Математична модель багаторівневої системи аналізу для виявлення небезпечних об'єктів

Багаторівнева система аналізу відеозображення призначена для автоматизованого виявлення мін на основі поєднання кількох методів комп'ютерного зору та датчиків. Вхідні дані системи – це послідовність відеокадрів місцевості (відеозображення) та сигнал від металодетектора, що сканує ту саму ділянку. Вихідні дані – оцінка ймовірності наявності вибухонебезпечного об'єкта (міни) у поточному кадрі або локації. Система поєднує результати кількох рівнів аналізу для підвищення достовірності виявлення мін. Зокрема, у дисертації розглядаються такі рівні:

- **Глибока модель YOLOv8** (версії  $m$  або  $n$ ): алгоритм детектування об'єктів [51], що аналізує зображення і знаходить підозрілі об'єкти (можливі міни) на ньому.
- **Модель ChatGPT-4.1 Vision**: мультимодальна модель (велика мовна модель з підтримкою зору), здатна інтерпретувати зображення [52]. Вона використовується для семантичного аналізу вмісту кадру та підтвердження чи спростування того, що знайдений об'єкт є міною, на основі свого знання та контексту.
- **Модель Gemini 2.5 Vision**: відкрита мультимодальна модель, розроблена Google, яка може виконувати різні завдання комп'ютерного зору (OCR (Optical Character Recognition), VQA (Visual Question Answering), підрахунок об'єктів тощо). У нашій системі Gemini 2.5 використовується як додатковий канал аналізу зображення [52], аналогічно до ChatGPT-4, для підвищення впевненості у класифікації об'єкта як міни.
- **Металодетектор**: наземний сенсор, що реагує на наявність металевих компонентів у ґрунті. Він надає додатковий сигнал, корисний для

виявлення мін із металевими частинами (навіть якщо візуально їх не видно).

*Принцип роботи багаторівневої системи.* Спочатку камера та металодетектор збирають дані з однієї і тієї ж області. Потім, на першому рівні, модель YOLOv8 обробляє зображення кадр за кадром, щоб знайти всі підозрілі об'єкти на фоні (ROI, Region of Interest) – тобто потенційні міни, відмінні від фону місцевості [41]. На наступних рівнях відео кадри додатково аналізуються моделями високого рівня, такими як ChatGPT-4.1 Vision та Gemini 2.5. Ці моделі виконують семантичний аналіз, наприклад, вони можуть отримати фрагмент зображення з виділеним об'єктом або весь кадр і відповісти на питання «*Чи є цей об'єкт міною (вибухонебезпечним предметом) чи ні?*». Таким чином, відбувається класифікація: визначення, чи підозрілий об'єкт є саме міною (або іншим вибухонебезпечним предметом), чи це хибна тривога (наприклад, камінь, уламок металу, сміття тощо).

Паралельно, сигнал металодетектора служить додатковим підтвердженням. Якщо в місці розташування підозрілого об'єкта присутній метал, то це ознака *штучного предмета*. Наземні міни зазвичай містять метал (корпус або деталі), хоча й у різних кількостях. Водночас металодетектор є неспецифічним сенсором – він реагує на будь-який метал, тому може спрацьовувати і на сторонні металеві об'єкти (уламки, гільзи тощо). Саме тому вкрай важливо виконати [41] *злиття даних* – інтегрувати результати від відеоаналізу (YOLO, ChatGPT, Gemini) та від металодетектора таким чином, щоб підвищити ймовірність правильного виявлення міни і одночасно понизити ймовірність хибних тривог. Таким чином, багаторівнева система покликана компенсувати недоліки окремих методів за рахунок їх поєднання. Кожен рівень аналізу додає інформацію: YOLOv8 швидко знаходить всі можливі цілі на зображенні (обмежена натренованою вибіркою), а ChatGPT-4.1 Vision та Gemini 2.5 «подивляться» на ці цілі уважніше, використовуючи накопичені знання, щоб відфільтрувати природні об'єкти чи безпечні предмети, а металодетектор дає незалежний фізичний сигнал про присутність



металевого об'єкта в ґрунті. Очікуваний результат – якщо хоча б один з цих компонентів зможе розпізнати міну, система в цілому згенерує тривогу. Такий підхід схожий на комплекс детекторів, де фінальний сигнал спрацьовує, якщо хоча б один із детекторів спрацював, що збільшує ймовірність виявлення цілі.

Етапи обробки: YOLOv8n, ChatGPT-4.1 Vision, Gemini 2.5 Flash, металодетектор. Нижче детальніше описані основні етапи (рівні) аналізу в нашій системі і їх виходи:

- Детекція YOLOv8: YOLO (*You Only Look Once*) – це нейронна мережа для детектування об'єктів, яка в єдиному проході по зображенню визначає координати bounding box'ів потенційних об'єктів та ймовірності належності цих об'єктів до певних класів. Зокрема, YOLOv8 розбиває зображення на сітку і для кожної комірки прогнозує декілька рамок та *confidence score* для кожної з них. Цей коефіцієнт впевненості відображає, наскільки модель впевнена, що в рамці міститься об'єкт певного класу (наприклад, міна). На виході першого етапу маємо набір знайдених регіонів, кожен з яких позначений як *потенційна міна* з деякою довірчою ймовірністю  $s_y$  (наприклад,  $s_y=0.75$  означає 75% впевненості моделі YOLO, що цей об'єкт – міна). Якщо на кадрі жодного підозрілого об'єкта не виявлено, YOLOv8 видає порожній результат (відсутність детектування) для цього кадру.
- Семантичний аналіз зображення (ChatGPT-4.1 Vision) – наступний рівень, що залучає потужну мультимодальну модель GPT-4.1. «gpt-4.1» з модальністю зору отримує або весь кадр, або вирізаний фрагмент з знайденим об'єктом, та відповідає на заданий запит. Приміром, система може сформулювати запит: *«Подивись на зображення. Чи бачиш ти протипіхотну міну чи інший вибухонебезпечний предмет?»*. Модель ChatGPT проаналізує візуальні ознаки (форму об'єкта, колір, оточення) та може видати відповідь на кшталт: *«Об'єкт виглядає як круглий металевий диск, схожий на міну»* або *«Ні, на зображенні лише каміння*

*та ґрунт, міни не бачу».* Для математичного та аналітично-структурного моделювання цього етапу введемо параметр  $s_c$  – оцінка впевненості від GPT-4 Vision, що об’єкт є міною. Хоча ChatGPT видає текстову відповідь, ми можемо інтерпретувати її зміст у ймовірнісний показник. Наприклад, якщо у відповіді чітко стверджується, що це міна, можна умовно поставити  $s_c \approx 0.9-1.0$ ; якщо ж GPT-4.1 Vision вагатиметься («не впевнений»),  $s_c$  може бути середнім ( $\sim 0.5$ ), а якщо заперечує наявність міни – близьким до 0. Такий підхід дозволяє включити вихід ChatGPT-4.1 у формальну модель як ще один сигнал/детектор. Роль цього етапу є використати знання великої моделі для зменшення кількості хибних тривог. GPT-4.1 може розпізнати контекст або деталі, які важко врахувати звичайному детектору. Наприклад, якщо YOLO помилково виділив камінь як «потенційну міну», ChatGPT-4.1 Vision з великою ймовірністю впізнає, що це природний об’єкт, і знизить впевненість, що це міна. Таким чином, штучний інтелект з рівнем розуміння сцени доповнює алгоритм YOLO. (Аналогічний підхід успішно застосовано для розпізнавання тварин: YOLO виділяє область з твариною, а модель GPT-4 Vision ідентифікує вид – те, чого один YOLO зробити не може. Комбінація дає набагато багатшу та точнішу інформацію, недосяжну одним методом).

- Додатковий аналіз моделлю Gemini 2.5 Flash. У багаторівневому конвеєрі як другий «візуально-мовний» експерт використовується Gemini 2.5 Flash – швидка малолатентна мультимодальна модель від Google з підтримкою аналізу зображень і тексту. Flash варіант оптимізований під низьку затримку та помірну вартість запитів, тому його доцільно запускати паралельно з GPT-4.1 Vision (для консенсусу) або каскадно (для верифікації спрацьовувань у режимі triage). Модель отримує той самий ROI/кадр і уніфікований запит на наявність вибухонебезпечного об’єкта, повертаючи текстове обґрунтування та ступінь впевненості (далі позначаємо як  $s_G$ ).

Комбінація двох різних VLM підвищує стійкість до помилок окремої моделі та дає ефект багатомодельної згоди: якщо GPT-4.1 Vision і Gemini 2.5 Flash незалежно підтверджують «YES» у межах спільного ROI, агрегована постеріорна ймовірність  $P_{\text{final}}$  зростає (відповідно до правил фузії, див. розд. 3). У зонах розбіжностей спрацювання Flash використовується як сигнал до повторної перевірки (додатковий прогін YOLO/повторний обліт/металоканал), а його довіри проходять калібрування (Platt/temperature) перед зваженим включенням у карту ризику. Практично це знижує залежність від одного вендора, забезпечує швидку семантичну верифікацію поверх класичного детектора та покращує баланс recall/precision у складних сценах (часткові оклюзії, неоднорідний фон), не збільшуючи критично загальну латентність системи.

- Сигнал металодетектора: Останній компонент – *фізичний датчик металу*. Він забезпечує сигнал  $Z$  (наприклад, напруженість електромагнітного поля або просто бінарний індикатор спрацювання), що відображає наявність металевих об'єктів під датчиком. В математичній моделі можна ввести бінарну змінну  $D_M$ :  $D_M=1$ , якщо металодетектор спрацював (виявив метал), і  $D_M=0$  – якщо ні. Також корисно знати ймовірність спрацювання металодетектора на міну:  $P_M=P(D_M=1|\text{міна присутня})$ . Ця ймовірність залежить від типу міни (для металевих мін близька до 1, для пластикових – низька). Важливим є і *ймовірність хибного спрацювання*  $P(D_M=1|\text{міна немає})$  – металодетектор може реагувати на будь-який металевий сміттєвий об'єкт у ґрунті. В нашій системі сигнал металодетектора використовується для підвищення впевненості, а саме якщо в регіоні, де YOLO позначив підозрілий об'єкт, також є сигнал від металодетектора, це вагомий аргумент, що об'єкт штучного походження (металевий) і може бути міною. Якщо ж металодетектор мовчить, а візуальна система бачить підозрілий предмет, можливо, це пластикова міна (тоді

відсутність металу не спростовує небезпеку) або ж таки хибна тривога (камінь). Отже, металодетектор слід використовувати обережно, бо його *негативний результат* не гарантує відсутності міни, але *позитивний результат* є сильним підтвердженням, що знайдене об'єкт є небезпечним.

Після проходження кадру через YOLOv8 ми отримуємо список кандидатів на міни з їхніми  $s_Y$ . Для кожного такого кандидата збираються додаткові ознаки: відповіді моделей GPT-4.1 ( $s_C$ ) та Gemini 2.5 ( $s_G$ ), а також сигнал металодетектора  $D_M$ . Тепер постає завдання інтегрувати всю цю інформацію та отримати фінальну оцінку ймовірності того, що кандидат справді є вибухонебезпечним об'єктом. Ця інтеграція – і є ключовою частиною математичної моделі багаторівневої системи.

*Математична модель інтеграції результатів.* Для побудови математичної моделі визначимо формально змінні та функції кожного блоку системи:

- Позначимо через  $E$  подію, що в розглянутій області присутня міна (ціль, яку треба виявити).  $E=1$  означає «міна є», а  $E=0$  – «міни немає».
- Вхідні дані:  $I$  – ознакове представлення відеозображення (наприклад, піксельні інтенсивності кадру або інші ознаки, які використовує нейронна мережа);  $Z$  – показники металодетектора (може бути безперервним сигналом або дискретним 0/1 після порогу).
- YOLOv8 як функція: позначимо через  $f_Y(I)$  операцію детектування YOLO на зображенні. Її результат – набір об'єктів  $\{O_i\}$  з координатами та оцінками впевненості. Формально для кожного знайденого об'єкта  $O_i$  модель дає *score*  $s_{Y,i}=P(\text{міна}|O_{i,I})$  – оцінку ймовірності, що об'єкт належить до класу «міна» (або «вибухонебезпечний предмет»). Якщо  $s_{Y,i}$  перевищує певний поріг детектування, об'єкт вважається виявленим як підозрілий. Надалі розглянемо один такий об'єкт (для спрощення позначень опустимо індекс  $i$ ).
- GPT-4.1 Vision як функція: нехай  $f_C(I, \text{prompt})$  – це оператор, що повертає текстову відповідь на запит (*prompt*) щодо зображення  $I$ . Ми

інтерпретуємо відповідь моделі як числовий бал  $s_C = g_C(f_C(I, \text{prompt}))$ , де  $g_C(\cdot)$ - деяка функція інтерпретації (наприклад, парсер, який перетворює відповідь «Так, це міна» у  $s_C=0$ , а «Ні, не схоже на міну» у  $s_C=0$ , проміжні невпевнені відповіді – у  $s_C$  між 0 і 1). Таким чином,  $s_C$  наближено можна трактувати як апостеріорну ймовірність міни за версією ChatGPT-4.1, тобто  $s_C \approx P(E=1|I)$  за оцінкою цієї моделі.

- Gemini 2.5 Vision як функція: аналогічно,  $f_G(I, \text{prompt})$ - відповідає на запит щодо зображення, і ми отримаємо оцінку  $s_G = g_G(f_G(\cdot))$ , яка інтерпретується як ймовірність міни за версією Gemini 2.5.
- Металодетектор як функція:  $f_M(Z)$  – функція обробки сигналу металодетектора. Найпростіше – це порогова функція, яка видає  $D_M=1$  якщо сигнал перевищує поріг, і  $D_M=0$  інакше. Також можна визначити величину  $s_M$ , що відображає ступінь «металевості» об'єкта (наприклад, нормалізований індикатор сили сигналу, де 0 – металеві елементи відсутні, 1 – сильний металевий відгук). Для бінарної моделі достатньо  $D_M$ .

Тепер, коли визначено виходи всіх компонентів ( $s_Y, s_C, s_G, D_M$ ), треба побудувати функцію, що об'єднує їх у кінцеву ймовірність наявності міни  $P(E=1|I, Z)$ .

### 3.1.1. Логічне об'єднання (правило «OR» або «AND»)

Найпростіша ідея – трактувати кожен канал як незалежний детектор міни, що видає сигнал тривоги. Тоді можна задати логічне правило прийняття рішення (рис.3.1). Наприклад:

$$D_{\text{final}} = D_Y \vee D_C \vee D_G \vee D_M \quad (3.1)$$

де  $D_Y$  – подія «YOLO детектувала міну» (тобто  $s_Y$  вище порогу), аналогічно  $D_C$ ,  $D_G$ , а  $D_M$  – спрацювання металодетектора. Ймовірність виявлення міни при такому правилі є максимізованою. Якщо позначити через:

$$P_Y = P(D_Y=1|E=1), P_C = P(D_C=1|E=1) \quad (3.2)$$

ймовірності виявлення [45] міни окремими каналами (за умови, що міна справді є), то для правила OR за припущення незалежності каналів маємо:

$$P(D_{\text{final}}=1|E=1) = 1 - (1-P_Y)(1-P_C)(1-P_G)(1-P_M) \quad (3.3)$$

Правило «AND» (логічне і), при якому сигнал тривоги генерується, лише якщо всі або більшість каналів підтверджують наявність міни. Це протилежна стратегія, яка знижує кількість хибних тривог, але й може пропустити більше мін. Наприклад, вимога, щоб і візуальні методи, і металодетектор погодилися щодо міни, суттєво підвищує довіру до тривоги (менше помилок), проте якщо хоча б один з каналів не спрацює (скажімо, міна пластикова – металодетектор мовчить), то загальна система її проігнорує (рис. 3.1). В реальних системах протимінної безпеки часто використовують комбінацію, в якій для попереднього виявлення – чутливе правило OR (щоб не пропустити міну), а для підтвердження – правило AND/більшість (щоб відфільтрувати хибні спрацювання).

У дисертації було використано підхід OR для підвищення ймовірності виявлення та орієнтування на максимально можливий  $P_{\text{detect}}$ , а в майбутньому планується корекція моделі для відсіювання хибних тривог.

### 3.1.2. Імовірнісне об'єднання (зважування або Баєсів підхід)

Альтернативою жорстким правилам є розрахунок кількісної ймовірності  $P(E=1|I, Z)$  на основі виходів моделей. Можна використати

формулу Байєса або просто навчити якусь функцію від  $s_Y, s_C, s_G, s_M$ . Наприклад, якщо припустити статистичну незалежність каналів (умовно незалежних за наявності міни і за відсутності), то згідно [53] з Байєсом:

$$P(E = 1 | I, Z) \propto P(E = 1) P(I, Z | E = 1) = P(E = 1) P(I | E = 1) P(Z | E = 1) \quad (3.4)$$

Але обчислювати  $P(I|E=1)$  безпосередньо складно. Натомість можна скористатись оцінками моделей як наближеннями до відповідних правдоподібностей. Нехай  $s_Y$  вже є наближенням  $P(E=1|I)$  за даними YOLO,  $s_C \approx P(E=1|I)$  за GPT-4.1,  $s_G$  – за Gemini, а  $s_M \approx P(E=1|Z)$ . Тоді можна агрегувати їх, наприклад, через зважене середнє:

$$P_{\text{final}} = \frac{\alpha s_Y + \beta s_C + \gamma s_G + \delta s_M}{\alpha + \beta + \gamma + \delta} \quad (3.5)$$

де  $\alpha, \beta, \gamma, \delta$  – вагові коефіцієнти, що відображають довіру до кожного джерела. Їх можна підібрати експериментально (або навіть оцінити через максимізацію правдоподібності на валідаційних даних). Наприклад, якщо YOLO дає багато помилкових детекцій, можна зменшити вагу  $\alpha$  і більше довіряти семантичним аналізаторам ( $\beta, \gamma$ ) та металодетектору ( $\delta$ ). Навпаки, якщо GPT-4.1 іноді помиляється або генерує нестабільні відповіді, його вагу можна трохи знизити. Результат  $P_{\text{final}}$  і є шуканою інтегрованою ймовірністю міни. Її можна порівняти з порогом для прийняття рішення: наприклад, якщо  $P_{\text{final}} > 0.8$  – сигналізувати про знахідку, якщо менше – вважати, що міни немає.

Зважене об'єднання плавно враховує всі джерела, якщо, скажімо, і YOLO, і всі моделі, і металодетектор дають високі впевненості, то  $P_{\text{final}}$  буде дуже близьким до 1. Якщо ж тільки один з каналів «за», а інші «проти», то підсумкова ймовірність буде середньою і можливо, недостатньою для спрацювання. В такий спосіб можна досягти хорошого компромісу між чутливістю та специфічністю системи.

### 3.1.3. Багаторівнева класифікація (паралельне рішення)

Ще один підхід, що витікає з мульти-рівневої природи системи – приймати рішення *паралельно*. Тобто кожний рівень аналізу (рис. 3.1) виконується незалежно від іншого.

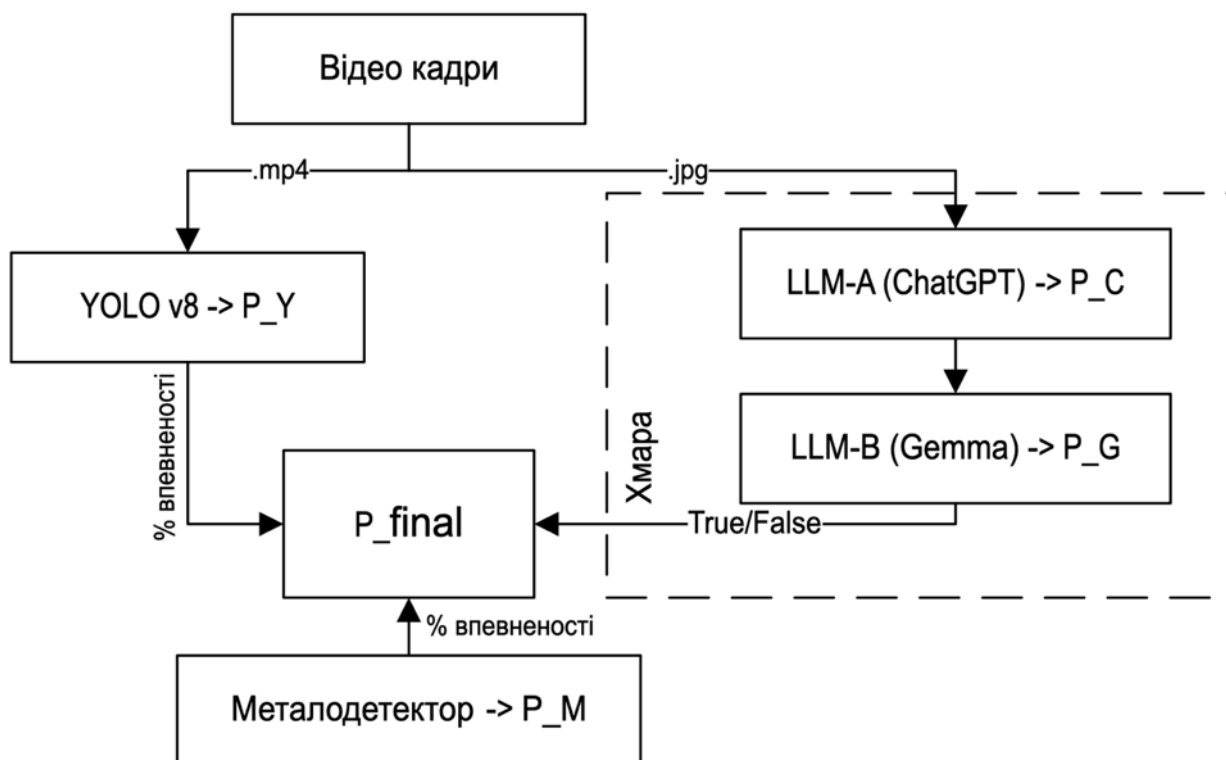


Рисунок 3.1 – Каскадний аналіз даних

Власне, так система і працює: ChatGPT та Gemini 2.5 не будуть аналізувати кожен кадр цілком, а тільки ті області, що позначені YOLO або металодетектором. Металодетектор може працювати паралельно і давати інформацію для тих самих областей. Математично, це можна описати як дерево рішень або умовний алгоритм:

1. Якщо  $s_Y$  (впевненість YOLO) нижче порогу  $\theta_Y$ , то на цьому кадрі *мін немає* ( $P_{final} \approx 0$ ) і подальший аналіз не потрібен.



2. Якщо YOLO виявив об'єкт ( $s_y \geq \theta_y$ ), то запускаємо рівень 2: ставимо GPT-4 і Gemini 2.5 запити щодо цього об'єкта. Отримуємо  $s_C, s_G$ .
3. Якщо обидва ці аналізатори впевнено заперечують міну ( $s_C, s_G$  низькі), то об'єкт вважається хибною тривоною і  $P_{\text{final}}$  зменшується.
4. Якщо хоча б один із високорівневих аналізаторів підтверджує міну ( $s_C$  або  $s_G$  високі), то переходимо до фінального підтвердження.
5. Враховуємо сигнал металодетектора: якщо  $D_M=1$ , підсилюємо впевненість; якщо  $D_M=0$ , лишаємо тільки візуальні докази.
6. На основі всіх даних приймаємо рішення – фактично, виконуємо логічне правило або розрахунок ймовірності, як описано вище.

Такий поетапний опис може бути використаний щоб знизити обчислювальні витрати: важкі візуально-мовні моделі використовується не на кожному пікселі, а тільки на окремих кадрах. Це підвищує ефективність але й підвищує вірогідність пропусків. Для максимально ефективного сканування місцевості, слід використовувати підхід коли усі рівні комп'ютерного зору опрацьовують кадри паралельно.

## **3.2 Виявлення вибухонебезпечних предметів за допомогою нейронної мережі YOLOv8**

### **3.2.1 Загальні відомості про архітектуру YOLOv8**

Сімейство YOLO (You Only Look Once) належить до одноетапних детекторів, у яких локалізація та класифікація об'єктів виконуються в межах єдиної, кінець-у-кінець диференційовної нейромережі. На відміну від двоетапних підходів, це мінімізує проміжні операції та знижує затримку, що критично для застосувань у реальному часі. Покоління YOLOv8 розвиває ідеї попередників, інтегруючи сучасні архітектурні рішення та відточені методики навчання/постобробки, завдяки чому досягається сприятливий компроміс між точністю і швидкодією, особливо помітний на дрібних цілях.

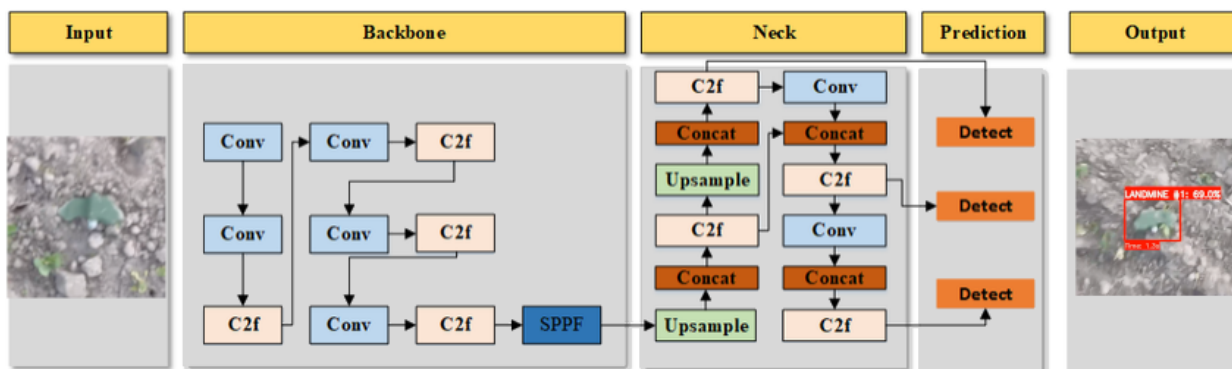


Рисунок 3.2 – Загальний процес детектування

Архітектуру YOLOv8 зручно інтерпретувати через три взаємопов'язані складові – backbone-neck-head (рис. 3.2). Backbone – це глибока згорткова мережа для мульти-масштабної екстракції ознак, в якій ранні шари видобувають низькорівневі текстури та контури, глибші – високорівневі семантичні уявлення. У реалізації використовують полегшені блоки на кшталт CSP-похідних / C2f і *depthwise separable* згорток, що зменшують кількість параметрів та обчислень без значущої втрати представницької сили. На виході формується піраміда ознак різних просторових масштабів, яка подається до neck-модуля багатомасштабного злиття та «очищення» ознак. Neck у YOLOv8 використовує оптимізовану PAnet-подібну схему з двонаправним потоком (згори-вниз і знизу-вгору), яка підтримує ефективний перенос контексту між масштабами: слабкі локальні сигнали від дрібних об'єктів підсилюються інформацією про сцену, що особливо важливо для малих мін типу ПФМ-1, які займають лише десятки пікселів і часто частково перекриті рослинністю або тінями.

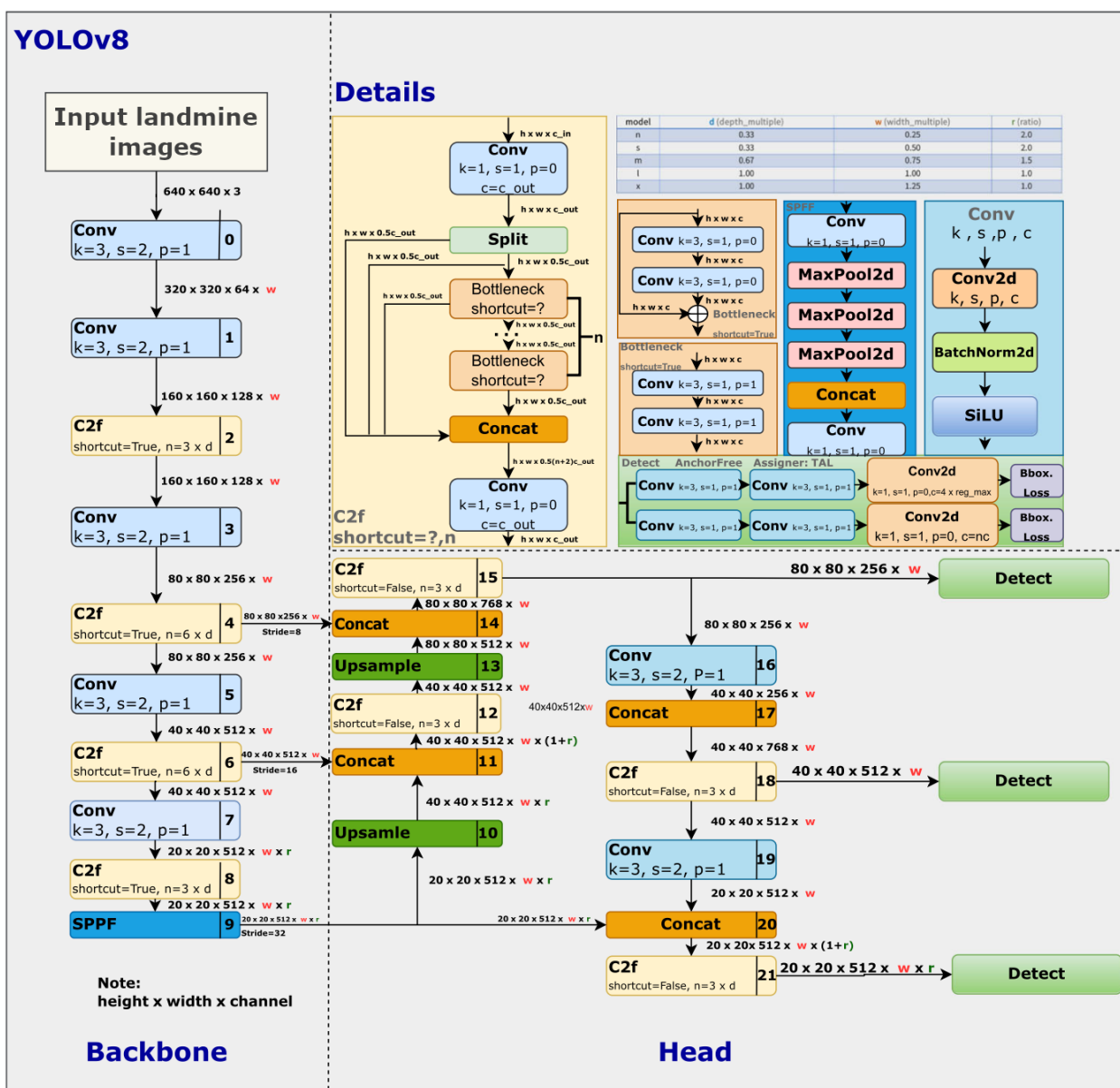


Рисунок 3.3 – Структура моделі YOLO v8

Кінцеві прогнози генерує head, у якій YOLOv8 переходить до anchor-free регресії рамок [54]. На відміну від ранніх версій із фіксованими наборами «якорів», anchor-free підхід безпосередньо регресує параметри прямокутників відносно осередків карт ознак. Це спрощує конфігурацію (менше гіперпараметрів), зменшує чутливість до підбору аспектних відношень і підвищує адаптивність до варіативних форм/масштабів об'єктів. Важливим компонентом є (рис. 3.3) decoupled head – розділення гілок класифікації та регресії координат, що покращує градієнтну динаміку і збіжність під час навчання: ознаки «вчаться» одночасно добре локалізувати межі і коректно

визначати клас, не конфліктуючи між собою. Формат виходів типово містить для кожного осередку (на кожному рівні масштабів) параметри рамки ( $x_1, y_1, x_2, y_2$ ) оцінку об'єктності/довіри (*confidence*) та, за потреби, логіти класів (у нашій постановці – одно-класова детекція ПФМ-1). Після агрегації прогнозів із різних масштабів застосовують NMS або Soft-NMS для видалення дубльованих/перекривних рамок за порогоми *conf* та *IoU*. Для відео доцільно додавати часове згладжування/трекінг (напр., проста IoU-асоціація із TTL чи Kalman/ByteTrack), що приглушує одиничні «спалахи» помилкових спрацьовувань і стабілізує траєкторії.

Навчання YOLOv8 виконується кінець-у-кінець (end-to-end) зі змішаною функцією втрат. Для регресії координат рамок використовують IoU-похідні функції (GIoU/DIoU/CIoU), нерідко у зв'язці з DFL (Distribution Focal Loss), яка дискретизує координати і підвищує точність локалізації. Класифікаційна складова зазвичай ґрунтується на BCE/Focal-варіантах для кращої роботи з дисбалансом і рідкісними класами. Щоб сформувати інваріантності, релевантні низьковисотній зйомці з БПЛА, застосовують HSV-варіації (освітлення/насиченість/відтінок), випадкові повороти/масштабування/зсуви, горизонтальні віддзеркалення, а Mosaic використовують обережно (щоб не «зламати» масштаб дуже дрібних цілей); локальне підсилення контрасту (напр., CLAHE) допомагає сценам із тінями. Призначення міток (matching) здійснюють динамічними асайнерами (на кшталт SimOTA/task-aligned), що покращує відповідність GT↔передбачення й пришвидшує збіжність.

Інференс у виробничому профілі складається з декодування та базового препроцесингу (масштабування з letterbox до вхідного розміру 640-960 px, нормалізація), одного прямого проходу мережею і постобробки (NMS/Soft-NMS + опційний трекінг). Підтримка пакетної обробки на GPU та змішаної точності (fp16) забезпечує високий FPS за низької латентності на кадр. Лінійка масштабів (n/s/m/l/x) дозволяє підбирати потрібний компроміс «якість ↔ швидкість» залежно від обчислювальних ресурсів (Edge/ground/cloud): від

надлегких профілів для швидкого скринінгу до середніх і важчих – для підвищеної точності.

У контексті гуманітарного розмінування переваги YOLOv8 проявляються на дрібних і частково оклюдованих об'єктах. Багато-масштабне злиття в писк підсилює слабкі сигнали ПФМ-1 серед трави чи ґрунтових артефактів, а anchor-free регресія зменшує залежність від підібраних заздалегідь «якорів», краще працюючи з варіативними аспектними відношеннями. Decoupled head покращує одночасно локалізацію і класифікацію. Разом із правильно підібраними аугментаціями та порогоми постобробки це надає детектору стійкість до змін освітлення, фактури фону й масштабу – саме тих факторів, що визначають складність візуального виявлення мін у польових умовах. У нашій багаторівневій системі YOLOv8 виконує роль швидкого та надійного первинного детектора, результати якого надалі уточнюються високорівневими мультимодальними моделями (LLM-vision) і можуть підтверджуватися сенсорними каналами, утворюючи цілісний конвеєр із підвищеною імовірністю виявлення та контрольованим рівнем хибних спрацьовувань.

### 3.2.2 Роль хмарного рівня у багаторівневій архітектурі

Хмарний рівень виконує функції масштабованого віддаленого інференсу та централізованого керування моделями/даними. На борту БПЛА (Raspberry Pi 4B) виконується лише запис та буферизація відео у 1920×1080 MPEG-4, а також службове логування (штампи часу). Передача сирих даних у поле робиться по Wi-Fi/SSH на польовий ПК, де відео проходить через алгоритм YOLOv8n, а далі – у хмарне середовище, де відбуваються ресурсоємні етапи, такі як декодування, препроцесинг, детекція, постпроцесинг (NMS/трекінг), формування відповіді та версіонування. Такий поділ *Edge→ground/cloud* дозволяє:

- зберегти енергобаланс і простоту бортової частини;

- забезпечити високу продуктивність (GPU) і можливість одночасної обробки кількох польотів;
- централізувати якість, безпеку та відтворюваність (контроль версій датасетів/ваг, журналювання, резервне копіювання).

Перший рівень обробки виконується на польовому ПК (MacBook Pro M3 Pro) з апаратним прискоренням через MPS. Саме тут працює профіль YOLOv8n (Ultralytics) як швидкий детектор дрібних цілей на кшталт ПФМ-1. Відео 1920×1080, записане на Raspberry Pi, доставляється на ноутбук захищеним каналом (SSH/scp/rsync) безпосередньо під час або одразу після польоту. Разом із відео передаються службові метадані (версії моделей, часові мітки), що забезпечує трасованість. На польовому ПК виконується базовий препроцесинг: декодування (FFmpeg) з витягненням таймкодів, за потреби – кадрова стабілізація (ORB+RANSAC) для приглушення мікровібрацій, масштабування до вхідної роздільності моделі (типово 640–960 px із letterbox без зміни пропорцій) та нормалізація інтенсивностей; у сценах із контрастними тінями застосовується адаптивна контрастна обробка (CLAHE). Далі YOLOv8n виконує інференс у близькому до реального часу (на M3 Pro ми досягаємо ефективних десятків FPS), а на виході (рис. 3.5) для кожного кадру отримуємо набір боксів  $[x_1, y_1, x_2, y_2]$ , клас та conf-оцінку. Архітектурно використовується anchor-free head із регресією коробок (напр., CIOU у поєднанні з DFL) і бінарною класифікацією, що практично покращує локалізацію малих об’єктів. Постпроцесинг включає NMS/Soft-NMS із узгодженими порогами conf/IoU, а також легкий часовий трекинг (IoU-асоціація з TTL або Kalman/ByteTrack) для приглушення одиничних «сплесків» і стабілізації траєкторій; за потреби можливе об’єднання декількох прогонів (WBF) для підвищення recall без суттєвої втрати precision.

Після первинної детекції формується уніфікований артефакт (рис. 3.5) у форматі JSON/GeoJSON із фіксацією mission\_id/flight\_id, версії ваг/коду, ефективної швидкості обробки та переліку детекцій на рівні кадрів. Приклад структури відповіді (скорочено):

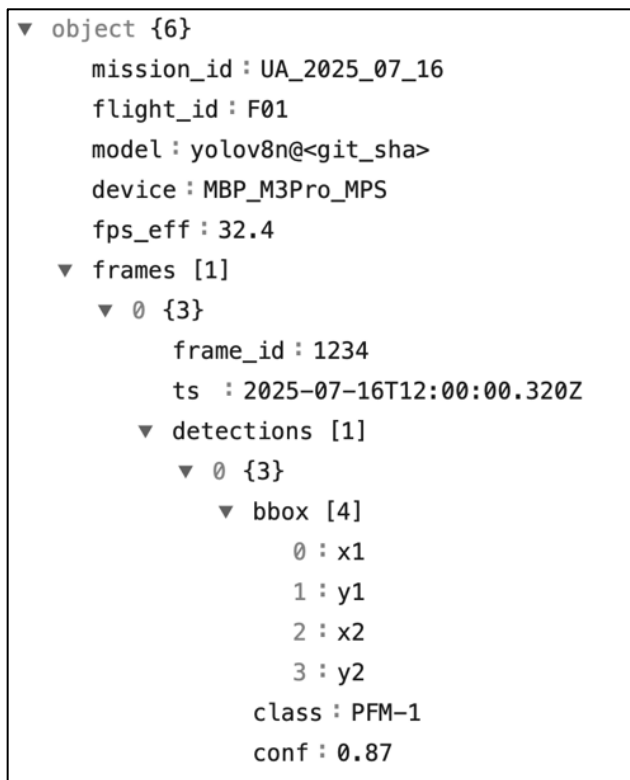


Рисунок 3.5 – Зображення дерева структури відповіді від моделі

З метою відтворюваності й контролю ланцюга постачання даних результати й/або вихідні ролики/кадри завантажуються на хмарний вузол захищеним каналом (scp/HTTPS) до стандартизованої структури каталогів: /ingest/<mission\_id>/<flight\_id>/{video.mp4 frames/frame\_%06d.jpg} та /meta/<mission\_id>/<flight\_id>/{video.json, model\_version.txt}. Для кожного файлу обчислюється SHA-256-хеш, що унеможливорює дублювання та дозволяє перевіряти цілісність. У хмарі ці об'єкти потрапляють у чергу задач (напр., Redis/RabbitMQ) і можуть оброблятися асинхронно воркерами з GPU, що спрощує горизонтальне масштабування. Для довгих роликів застосовується чанкування (умовно по 10–20 с) з незалежною обробкою та подальшим зшиванням метаданих; усі проміжні операції логуються (ETL-час, GPU-util, кількість/щільність детекцій), а критичні події алертуються. Доступ до API захищається mTLS/JWT, шифрування даних на зберіганні – AES-256, версії ваг і коду «заморожуються» (Docker/conda-lock) для гарантії відтворюваності. Геоприв'язка виконується

на наземно-хмарному етапі post-hoc: за уніфікованими часовими мітками узгоджуються логи Pixhawk 6C (GPS/INS/барометр/лідар), координати інтерполюються у локальній системі (WGS-84  $\rightarrow$  ENU), після чого детектування переносяться у геопросторові шари ГІС. На основі цього формується карта ризику (heatmap/кластеризація) та шар подій із атрибутами впевненості, що надалі використовуються для фузії з металоканалом і VLM-верифікацією. Така організація конвеєра дає змогу приймати оперативні рішення безпосередньо в полі (завдяки швидкому першому проходу на ноутбучі), а також виконувати поглиблений аналіз у хмарі без навантаження польового ПК з можливістю повторних прогонів, калібрування порогів і побудови агрегованих звітів. За потреби той самий профіль YOLOv8n може бути повторно запущений у хмарі (ідентичні ваги/контейнери) для масштабного переобчислення місій або абляційних досліджень порогів/гладжування, однак у базовій конфігурації саме польовий ПК виконує першу детекцію та фільтрацію. Такий розподіл (польовий швидкий прохід  $\rightarrow$  хмарна архівація й аналітика) мінімізує вимоги до каналу зв'язку з БПЛА, знижує ризики втрат даних і забезпечує керовану затримку всього ланцюга «захоплення  $\rightarrow$  детекція  $\rightarrow$  геоприв'язка  $\rightarrow$  візуалізація» [55].

### 3.2.3 Підготовка даних для навчання YOLO v8

*Польові дані.* Вихідні відеоматеріали отримували під час низьковисотних польотів ( $\approx 2\text{--}3$  м) (рис. 3.6) над ділянками з різним фоном (трава/грунт/тіні). Маршрути включали прямолінійні проходи та розвороти на невисокій швидкості, щоб мінімізувати розмиття та зберегти контури. Відео сегментувалося на кадри з часовими мітками та жодної просторової прив'язки на цьому етапі не виконувалося.





Рисунок 3.6 – Приклади зображень для формування датасету

*Класи та фокус датасету.* Навчальна вибірка цілеспрямовано сфокусована на протипіхотній міні ПФМ-1, а всі інші об'єкти трактуються як негативний фон. Для підвищення роздільної здатності класифікаційної межі фон навмисно «зашумлено» відволікаючими факторами (різні типи рослинності, ґрунтові артефакти, дрібні побутові предмети, тіні), аби змодельовати реальні сцени з високою візуальною складністю. Постановка є одно-класовою: «PFM-1 vs. background», що дозволяє зосередити потужність моделі на виявленні малих цілей при широкій варіативності фону та мінімізувати хибні спрацьовування за рахунок багатого негативного піднабора.

*Розмітка в Label Studio.* Анотації [61] виконувалися прямокутними bounding boxes класу PFM-1 (рис. 3.7), причому бокс охоплював лише видиму частину корпусу з мінімальним технічним відступом по краях, щоб уникнути захоплення фону. У випадках часткових оклюзій (трава, тіні, ґрунт) розмічалася саме видима ділянка без екстраполяції невидимих контурів. Кадри без цілей залишалися нерозміченими й входили до негативного піднабора. Для уніфікації підходу застосовано короткий гайд із прикладами типових помилок (тіні від рослин, відблиски, дрібне сміття), що зменшує варіативність анотацій і покращує стабільність навчання моделі.

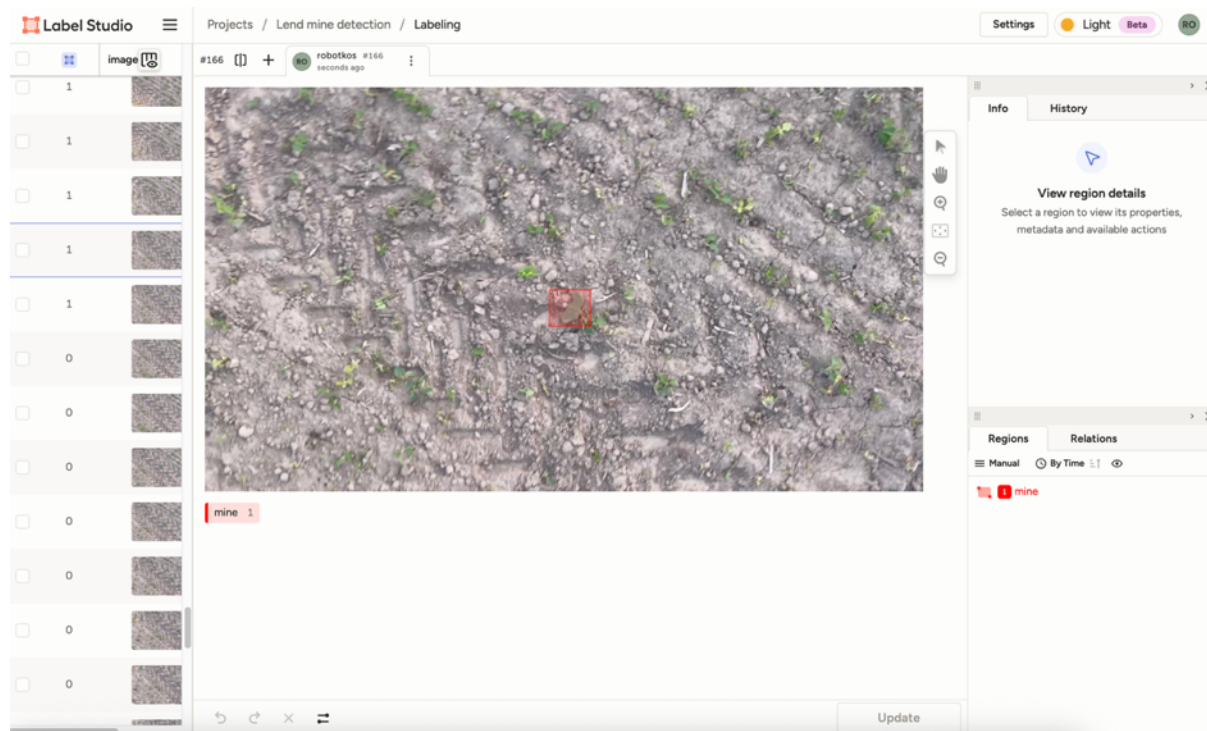


Рисунок 3.7 – Приклад інтрфейсу Label Sudio

*Контроль якості.* Подвійного рецензування розмітки не проводилося; якість забезпечували вибірковими візуальними переглядами та автоматизованими перевірками консистентності. Зокрема, валідувалися непорожність і коректність координат bounding box (порядок вершин, межі в межах кадру), відповідність класових міток декларації датасету, допустимі площі/співвідношення сторін боксів для запобігання аномаліям. Дефектні кадри (сильне розмиття рухом, суттєві артефакти кодування, критичні пересвічення/недоекспозиції) відсівалися до навчання.

*Поділ на множини.* Датасет поділено на train/val/test у пропорції  $\approx 70/15/15$  зі стратифікацією за типами фону (трава/грунт/змішані сцени) та умовами освітлення (сонце/тінь/перехідні стани), щоб мінімізувати «витік» сцен між множинами та отримати коректні оцінки узагальнювальної здатності. Для кожного кадру збережено точну часову мітку, яка надалі використовується під час геоприв'язки.

*Експорт.* Проєкт Label Studio налаштовано на експорт у формат YOLO, а іменування файлів містить часові мітки для синхронізації з телеметрією, але без GPS-координат. Геоприв'язка детекцій виконувалася на етапі постобробки вже переданого по SSH відео: кадри узгоджувалися з логами Pixhawk 6C за часом, після чого кожній детекції присвоювалися lat/lon/alt у вибраній системі відліку.

*Відтворюваність.* Для кожного релізу датасету зберігаються dataset.yaml (шляхи train/val/test, словник класів), контрольні суми зображень, а також протокол експорту з Label Studio (версія проєкту та мапінг полів). Така фіксація метаданих забезпечує стабільне повторне навчання та прозорий аудит усіх перетворень даних.

### **3.2.4 Навчання, валідація та калібрування нейромережі YOLOv8**

Процес налаштування алгоритмічного ядра системи передбачає визначення оптимальних гіперпараметрів навчання, стратегій аугментації даних та критеріїв оцінювання ефективності моделі. Як базові архітектури для проведення експериментів та розгортання у хмарному середовищі обрано моделі YOLOv8n та YOLOv8m відповідно. Розмір вхідного зображення уніфіковано до 640 пікселів, проте для окремих сценаріїв детектування дрібних об'єктів передбачено збільшення роздільної здатності до 960 пікселів. Навчання моделей здійснюється протягом орієнтовно 100 епох із застосуванням механізму дострокового припинення (early stopping), який спрацьовує у разі відсутності покращення метрики val mAP. Оптимізація ваг нейромережі виконується за допомогою алгоритмів SGD або AdamW зі стандартними налаштуваннями бібліотеки Ultralytics, при цьому регулювання швидкості навчання відбувається за косинусним законом або методом одного циклу. Розмір пакету (batch size) адаптується динамічно залежно від доступного обсягу відеопам'яті, а для прискорення обчислень там, де це можливо, використовується режим змішаної точності (fp16).

З метою підвищення узагальнювальної здатності детектора застосовується комплекс методів аугментації даних. Він включає геометричні перетворення, такі як випадкові повороти, масштабування, зміщення, горизонтальне віддзеркалення та кадрування з автоматичною корекцією обмежувальних рамок. Для покращення стійкості до змін освітлення використовуються перетворення у колірному просторі HSV, а також методи підвищення різкості та вирівнювання гістограми (CLaNE), що є особливо актуальним для сцен із глибокими тінями. Варто зазначити, що техніка Mosaic застосовується з низькою ймовірністю для збереження коректного масштабу об'єктів типу ПФМ-1 відносно фону. Оцінювання якості роботи моделі базується на розрахунку середньої точності (mAP) з порогоми перекриття 0.5 та діапазоні 0.5–0.95, а також показників Precision, Recall та F1-score. Додатково контролюються параметри швидкодії, зокрема затримка на етапах попередньої обробки, інференсу та постобробки, а також частота кадрів (FPS). Вибір робочих порогів здійснюється шляхом аналізу кривих точності-повноти (PR-curve) та максимізації F1-критерію з обов'язковою верифікацією на складних прикладах, що містять траву, тіні або побутове сміття.

Калібрування порогових значень враховує специфіку задачі гуманітарного розмінування, де пріоритетним є мінімізація пропуску цілей (Recall). Стартовий поріг впевненості встановлюється у діапазоні 0.25–0.35, а поріг IoU для процедури пригнічення немаксимумів (NMS) – у межах 0.45–0.55. Допускається зниження порогу впевненості, що може призвести до зростання кількості хибнопозитивних спрацювань, які згодом фільтруються на вищих рівнях системи за допомогою верифікації великими мовними моделями або сенсорної фузії. Пілотні випробування у хмарному середовищі з консервативними налаштуваннями продемонстрували точність на рівні 80%, детальний аналіз якої наведено у розділі експериментальних досліджень.

Для сценаріїв, що вимагають детектування надзвичайно дрібних об'єктів, реалізовано опціональний механізм обробки зображень. Він передбачає або поділ кадру на фрагменти (тайлинг) розмірністю  $2 \times 2$  чи  $3 \times 3$  з перекриттям та

подальшим об'єднанням результатів методом WBF, або попереднє збільшення критичних областей інтересу (super-resolution). У хмарній архітектурі цей функціонал реалізовано як окремий профіль інференсу, що активується за запитом.

На поточному етапі розробки система спеціалізована на ідентифікації одного класу вибухонебезпечних предметів – протипіхотних мін ПФМ-1. Стратегія подальшого розвитку передбачає розширення класифікатора шляхом додавання нових об'єктів, зокрема мін типу ПМН-2, уламків боєприпасів та характерних маскувальних предметів. Реалізація цього завдання базується на методах інкрементального навчання, що включають балансування класів та використання складних негативних прикладів для зменшення помилок першого роду.

Адаптація системи до нових умов експлуатації забезпечується впровадженням у хмарному середовищі контуром активного навчання. Цей процес складається з кількох послідовних етапів. Спершу здійснюється автоматична вибірка кадрів, на яких зафіксовано невпевнені спрацювання детектора, що характеризуються низьким рівнем довіри або нестабільністю трекінгу. Відібрані зображення експортуються до системи розмітки (наприклад, Label Studio) зі статусом, що потребує перегляду. Після експертної розмітки або уточнення та контролю консистентності даних проводиться перенавчання моделі. Завершальним етапом є A/B-валідація оновленої версії та її поступове розгортання (canary release). Важливим аспектом функціонування системи є наскрізне керування версіями. Кожна відповідь сервісу супроводжується метаданими, що містять ідентифікатори версії моделі (хеш ваг або коміту) та версії набору даних (тег DVC). Такий підхід забезпечує відстежуваність результатів та дозволяє корелювати динаміку зміни метрик ефективності з конкретними релізами програмного забезпечення.

### 3.2.5 Постпроцесинг даних від модулів системи, геопросторова прив'язка та інтеграція з ГІС

Ключовим етапом функціонування розробленої системи є трансформація «сирих» виходів нейромережевого детектора у верифіковані геопросторові дані, придатні для прийняття тактичних рішень щодо розмінування. Процес розпочинається з первинного логування кожної події детектування, яка фіксується у базі даних у вигляді розширеного структурованого запису. Цей запис містить точну часову мітку з мікросекундною роздільною здатністю, унікальний ідентифікатор кадру відеопотоку, нормалізовані координати обмежувальної рамки об'єкта ( $x_{\min}$ ,  $y_{\min}$ ,  $x_{\max}$ ,  $y_{\max}$ ), прогнозований клас вибухонебезпечного предмета, рівень довіри нейромережі (confidence score) та хеш версії використаної моделі. Збереження версійності моделі є критично важливим для забезпечення відтворюваності експериментів та можливості ретроспективного аналізу ефективності різних ітерацій алгоритмів навчання. Наступним кроком є складна процедура геоприв'язки (georeferencing), яка вирішує задачу співставлення об'єктів на площині зображення з їхнім фізичним розташуванням на місцевості. Оскільки відеопотік та дані телеметрії польотного контролера (на базі Pixhawk 6C) надходять асинхронно та з різною частотою дискретизації, система виконує часову інтерполяцію логів GPS/IMU для знаходження точних параметрів положення дрона в момент зйомки конкретного кадру. При цьому враховується апаратна затримка на захоплення кадру сенсором камери та час, необхідний для декодування відеопотоку. У результаті кожному детектованому об'єкту присвоюються глобальні геодезичні координати (широта, довгота, абсолютна висота) та кутові параметри орієнтації носія (heading/yaw), що дозволяє точно спроектувати центр обмежувальної рамки на земну поверхню.

Для підвищення точності локалізації та нівелювання високочастотного шуму, притаманного супутниковим навігаційним системам, застосовуються

методи цифрової фільтрації. Зокрема, для згладжування траєкторії та координат виявлених цілей використовується фільтр Калмана або метод ковзного середнього, що дозволяє усунути хибні «стрибки» координат, викликані вібрацією платформи або короткочасною втратою супутників. Для забезпечення можливості проведення метричних вимірювань, таких як оцінка площі забруднення або дистанції між мінами, виконується математичне перетворення координат із глобальної еліпсоїдальної системи WGS-84 у локальну декартову систему координат ENU (East-North-Up). Це перетворення дозволяє оперувати лінійними величинами в метрах відносно обраної базової точки, що є необхідним для коректного планування місій наземних роботів-саперів.

Фінальним етапом обробки є агрегація та експорт даних у стандартні формати геоінформаційних систем (GIS), такі як GeoJSON, Shapefile або CSV. Це забезпечує інтероперабельність розробленого рішення з існуючими професійними інструментами протимінної діяльності (наприклад, IMSMA). На основі експортованих масивів даних реалізовано алгоритми просторової кластеризації, зокрема DBSCAN (Density-Based Spatial Clustering of Applications with Noise), що дозволяє автоматично окреслювати межі мінних полів, ігноруючи поодинокі викиди. Додатково генеруються теплові карти щільності загроз (heatmaps), які візуалізують розподіл імовірності знаходження вибухонебезпечних предметів та слугують основою для пріоритизації маршрутів гуманітарного розмінування.

### **3.2.6 Експлуатаційні характеристики хмарного сервісу та обґрунтування вибору YOLOv8**

Архітектура програмного інтерфейсу (API) хмарного сервісу спроектована за принципами RESTful, що забезпечує гнучкість та масштабованість системи. Вона підтримує повний цикл асинхронної взаємодії: від завантаження великих масивів відеоданих або пакетів зображень

через захищені канали зв'язку до моніторингу статусу обробки та отримання фінальних звітів. Для забезпечення надійності та прогнозованості роботи системи в умовах реальних місій визначено чіткі угоди про рівень обслуговування (SLO). Зокрема, система гарантує обробку стандартного пакету даних обсягом 500 кадрів за час, що не перевищує 30 секунд, на обчислювальних вузлах, оснащених сучасними графічними прискорювачами. Висока відмовостійкість досягається завдяки реалізації механізмів черг повідомлень та політик автоматичного перезапуску завдань (retry policies) у разі виникнення тимчасових мережових або апаратних збоїв, що дозволяє утримувати сумарний відсоток помилок інференсу на рівні нижче 0.5%. Особлива увага в архітектурі приділяється аспектам інформаційної безпеки та етики поводження з даними. Система включає модулі для автоматичної анонімізації візуальних даних, якщо на кадрах присутні люди або цивільна інфраструктура. Крім того, реалізовано сувору рольову модель доступу (RBAC), яка обмежує доступ до точних географічних координат виявлених загроз. Інформація про локалізацію мінних полів надається виключно авторизованим підрозділам ДСНС або сертифікованим операторам протимінної діяльності, що унеможливорює використання системи сторонніми особами у нерегламентованих цілях.

Вибір нейромережевої архітектури YOLOv8 як основного детектора є науково обґрунтованим рішенням, що базується на специфіці задачі пошуку малорозмірних об'єктів, таких як протипіхотні міни ПФМ-1 («пелюстка»). Ці об'єкти характеризуються надзвичайно малою проєкцією в кадрі при аерозйомці та високим ступенем візуального злиття з фоном (камуфляж). YOLOv8 демонструє переваги завдяки використанню вдосконаленої шії мережі типу PANet (Path Aggregation Network), яка забезпечує ефективну двосторонню агрегацію ознак: семантична інформація передається з глибоких шарів до поверхневих, а деталізована просторова інформація – навпаки, що критично важливо для збереження дрібних деталей об'єктів. Іншим вирішальним фактором став перехід до концепції Anchor-free (без якірних



рамки). Традиційні детектори, що покладаються на фіксовані набори якорів, часто мають проблеми з детекцією об'єктів нестандартних форм або екстремально малих розмірів, якщо якорі не були ідеально налаштовані під конкретний датасет. Anchor-free підхід YOLOv8 усуває цю проблему, прогножуючи центр об'єкта безпосередньо, що підвищує здатність до узагальнення. Крім того, архітектура Decoupled Head, яка розділяє процеси класифікації та регресії координат обмежувальної рамки на різні гілки, дозволяє досягти кращої збіжності навчання та точнішого позиціонування рамок навколо об'єкта. У комплексі з реалізованою стратегією тайлінгу (розбиття зображення на фрагменти) та каскадною верифікацією, дана архітектура забезпечує необхідний баланс між чутливістю до дрібних загроз та швидкістю обробки даних у масштабах реальних гуманітарних операцій.

### **3.3 Хмарна обробка: семантична верифікація з використанням Gemini 2.5 Flash**

У запропонованій багаторівневій архітектурі системи другий, «семантичний» рівень обробки реалізовано на базі мультимодальної моделі Gemini 2.5 Flash від Google DeepMind. На відміну від детекторів першого рівня (YOLOv8), які потребують збору великих розмічених вибірок та тренування під кожен тип боєприпасу, у даній роботі використано підхід Zero-shot inference [22]. Він передбачає використання базових знань фундаментальної моделі (Foundation Model), набутих нею під час попереднього навчання на надвеликих масивах даних, для ідентифікації об'єктів без додаткового донавчання (fine-tuning). Такий підхід забезпечує високу гнучкість системи та дозволяє швидко адаптувати її до нових типів загроз шляхом модифікації текстових інструкцій.

Замість класичного градієнтного навчання, адаптація моделі до задачі гуманітарного розмінування здійснюється через механізм контекстних підказок (prompts). Вхідний вектор даних для моделі формується як пара «зображення + системна інструкція». Системна інструкція виконує роль

«м'якого» налаштування поведінки моделі, визначаючи її рольову функцію та формат вихідних даних. У розробленій системі застосовано спеціалізований промпт (рис. 3.8), що задає моделі роль експерта з безпеки (safety flagger). Дане зображення демонструє можливості моделі, а інтеграція реалізована за допомогою офіційного API.



You are an image safety flagger for a humanitarian demining dataset. Your job is only to flag if an image likely contains an ordnance/UXO/mine-like object. Respond with: YES (suspicious ordnance-like object visible) or NO (none visible). Shortly describe...

◆ Analysis ▼

**YES**

The image contains a **PFM-1 anti-personnel mine** (commonly known as a "butterfly mine" or "petal mine"). This is a highly dangerous explosive device.

Рисунок 3.8 – Зображення відповіді від моделі Gemini 2.5 на запит ідентифікації

Текстова конструкція має наступний вигляд:

*(укр. переклад: «Ви є спеціалістом з позначення безпеки зображень для набору даних гуманітарного розмінування. Ваше завдання полягає лише в тому, щоб позначити зображення, ймовірно, містить боєприпаси/невибухлі*

боєприпаси/об'єкти, схожі на міни. Дайте відповідь: ТАК (підозрілі об'єкти, схожі на боєприпаси, видно) або НІ (не видно). Якщо можливо, коротко опишіть тип об'єкта»)

Ця інструкція реалізує декілька критичних функцій:

1. *Встановлення контексту (Role Prompting)*: Фраза «*You are an image safety flagger...*» звужує простір пошуку рішень моделі, фокусуючи її увагу на специфічних паттернах, характерних для вибухонебезпечних предметів (ВНП).
2. *Обмеження області відповідальності*: Вказівка «*humanitarian demining dataset*» допомагає моделі правильно інтерпретувати контекст (наприклад, не сприймати міну як побутовий предмет) та знижує ймовірність відмови у відповіді через внутрішні фільтри безпеки (safety filters), оскільки запит позиціонується як гуманітарна, а не військова задача.
3. *Формалізація виходу*: Жорстка вимога «*Respond with: YES... or NO...*» перетворює генеративну задачу на задачу бінарної класифікації, що спрощує подальший автоматичний парсинг відповіді програмними засобами [42].

Алгоритм роботи хмарного модуля побудовано за принципом верифікації гіпотез. Первинний детектор (YOLOv8) виділяє на кадрі області інтересу (ROI). Далі ці, та інші зображення передаються до API Gemini 2.5 Flash разом із затвердженим промптом. Завдяки мультимодальній архітектурі, модель аналізує візуальні ознаки об'єкта (форму, текстуру, наявність підричника) у поєднанні з семантичним контекстом оточення (трава, ґрунт, сміття). Відповідь моделі обробляється за наступною логікою: наявність токена «YES» на початку генерації інтерпретується як підтвердження небезпеки ( $s_G = 1$ ), токен «NO» – як відсутність загрози ( $s_G = 0$ ). Додатковий текстовий опис («Shortly describe type») використовується для логування та надання оператору пояснення рішення (Explainable AI), але не впливає на автоматичну логіку керування у реальному часі. Використання базової моделі

Gemini 2.5 Flash без донавчання має низку експлуатаційних переваг. По-перше, це усуває необхідність збору тисяч розмічених прикладів для кожного нового типу мін, що є критичним в умовах появи нових зразків ВНП. По-друге, модель демонструє високу здатність до узагальнення, коректно розпізнаючи об'єкти, які частково деформовані, забруднені або знаходяться у нестандартному положенні, спираючись на свої «енциклопедичні» знання про вигляд боєприпасів. Водночас, застосування великої мовної моделі вносить додаткову затримку (латентність) у контур керування порівняно з локальними неймережами. Тому в розробленій системі Gemini 2.5 Flash виконує функцію асинхронного аудитора: вона перевіряє лише ті об'єкти, щодо яких у швидкого детектора YOLO виникла «невпевненість» (низький score), або ж вибірково верифікує високопріоритетні цілі для мінімізації помилкових спрацювань на складному фоні.

### 3.4 Застосування OpenAI API для аналізу зображень

У нашій багаторівневій архітектурі візуальна модель OpenAI GPT-4.1 Vision використовується як універсальний «семантичний» детектор, що за замовчуванням аналізує кожний кадр відеопотоку, отриманого з БПЛА, незалежно від того, чи спрацював первинний детектор YOLOv8. Така політика «все до хмари» свідомо обрана для покриття двох критичних прогалин: по-перше, класів та візуальних патернів, на які YOLO не навчалася (out-of-scope / open-world сценарії), по-друге, сцен, де дрібний або частково оклюдований об'єкт не перевищує пороги локального детектора (низький *conf*, невиразні контури, неоднорідний фон). Водночас GPT-4.1 Vision дає короткі текстові пояснення, що значно підсилює аудитуваність і зручність ручної верифікації в польових умовах.

З інженерної точки зору, конвеєр організовано так. Після розбиття відео на кадри з точними часовими мітками (синхронізованими з Raspberry Pi / польовим ПК) формуються батчі зображень, які передаються до OpenAI API у

хмарі. Щоб стандартизувати відповіді та звести їх до машинно-читаного вигляду, для всіх запитів застосовується однаковий промпт:

*«You are an image safety flagger for a humanitarian demining dataset. Your job is only to flag if an image likely contains an ordnance/UXO/mine-like object. Respond with: YES (suspicious ordnance-like object visible) or NO (none visible). Shortly describe type if you can.»*

Ми свідомо обираємо надзвичайно коротку інструкцію, бо довгі, «багатослівні» промпти збільшують варіабельність відповідей і час інференсу. У відповідь від GPT-4.1 Vision очікується два елементи: бінарне рішення («YES» або «NO») і дуже стислий опис (1-2 речення) з вказанням ознак, на яких базується висновок (форма, колір, часткова оклюзія, схожість із серією PMN тощо). Текст відповіді парситься регулярними правилами до числового балу  $s_c \in [0,1]$ : однозначне «YES» з конкретизацією типу мапується в 0.8-1.0; однозначне «NO» - у 0.0-0.2; невизначені формулювання («not sure», «uncertain», «cannot confirm») - у проміжний діапазон. Додатково застосовується м'яке калібрування через лінійне масштабування під цільову *precision-recall* криву валідаційного набору; у майбутніх ревізіях передбачене ізотонічне калібрування або Platt scaling, але нині використовується простіший підхід, що вже забезпечує стабільність порогів.

Оскільки кадри йдуть потоком із польового ПК до хмари, API виклики виконуються асинхронно з лімітами паралелізму, тайм-аутами та ретраями. Для зменшення витрат і латентності ми: (1) вимикаємо передавання «важких» метаданих (кадри очищено від наклейок телеметрії; GPS-дані не надсилаються), (2) активуємо кешування ідентичних або близько повторюваних кадрів (хешуванням вмісту), (3) шикуємо кадри у пріоритетні черги: усі кадри йдуть до GPT-4.1 Vision, але кадри, де YOLO вже дав високу впевненість на відомому класі (ПФМ-1), можуть оброблятися з нижчим пріоритетом або з меншою частотою, щоб звільнити пропускну здатність для невпевнених/нових сцен. Такий режим не суперечить нашому правилу

«все до хмари», а лише вирівнює навантаження в багатопоточному плануванні.

Після отримання відповідей GPT-4.1 Vision кожен рядок журналу має структуру {timestamp, frame\_id, gpt\_decision, gpt\_conf, gpt\_rationale}. На етапі пост-hoc інтеграції ці записи геоприв'язуються так само, як і детектування YOLO: за часовими мітками кадру виконують узгодження з логами Pixhawk 6C, інтерполяцію позиції та висоти, а далі - перетворення WGS-84 → ENU для просторової аналітики й побудови карт небезпеки. Якщо в тому самому кадрі є виходи YOLOv8, формується інтегральна оцінка  $P_{final}$  (зважене агрегування  $\alpha_{SY} + \beta_{SC} + \dots$  або простий OR/AND залежно від режиму). Якщо YOLO мовчить, але GPT-4.1 Vision дає YES з високою впевненістю, кадр отримує позначку пріоритетного ручного перегляду та відображається на карті як семантична підозра без точного бокса (локалізація - на рівні кадру). У практиці це виявилось корисним для «нестандартних» випадків: уламків мін, частково вивернутих фрагментів корпусу, або сильно замаскованих об'єктів, де є контекстні ознаки, але немає чіткої геометрії для щільного детектора. Нижче узагальнено п'ять реальних прикладів (скріншоти наведені у статті).



You are an image safety flagger for a humanitarian demining dataset. Your job is only to flag if an image likely contains an ordnance/UXO/mine-like object. Respond with: YES (suspicious ordnance-like object visible) or NO (none visible). Shortly describe type if you can.

**YES** — a small green circular object resembling a mine or fuze component (landmine-like, possibly pressure-activated) is visible.

Рисунок 3.9 – Запит та відповідь від візуальної моделі №1

На кадрі справді наявний зелений округлий предмет, а опис моделі коректно вказує на колір, форму, та тип «нажимної дії».



You are an image safety flagger for a humanitarian demining dataset. Your job is only to flag if an image likely contains an ordnance/UXO/mine-like object. Respond with: YES (suspicious ordnance-like object visible) or NO (none visible). Shortly describe type if you can.

YES — suspicious ordnance-like object visible.

Short description: green, partially buried object resembling a **landmine or mine fuze assembly** in grass (pressure-activated mine-like shape).

Рисунок 3.10 – Запит та відповідь від візуальної моделі №2

Модель правильно підмічає що бачить зелений, частково прихований об'єкт у траві. Даний приклад демонструє широкі можливості ідентифікації складних сцен.



You are an image safety flagger for a humanitarian demining dataset. Your job is only to flag if an image likely contains an ordnance/UXO/mine-like object. Respond with: YES (suspicious ordnance-like object visible) or NO (none visible). Shortly describe type if you can.

NO (none visible).

Рисунок 3.11 – Запит та відповідь від візуальної моделі №3



Це істинно негативний випадок (мін немає), і модель адекватно не «галюцинує», даючи чітку відповідь ні (NO). На цьому зображенні міна присутня, але дрібна та частково зливається з фоном, тому модель її пропускає - це класичний хибнонегатив для open-world бінарної перевірки на низькій роздільності.



You are an image safety flagger for a humanitarian demining dataset. Your job is only to flag if an image likely contains an ordnance/UXO/mine-like object. Respond with: YES (suspicious ordnance-like object visible) or NO (none visible). Shortly describe type if you can.

NO (none visible).

No clear mine/UXO features; appears to be soil with small stones/vegetation and an irregular non-ordnance fragment.

Рисунок 3.12 – Запит та відповідь від візуальної моделі №4

За сукупними тестами (рис. 3.9-3.12) на нашій вибірці GPT-4.1 Vision у самотійному бінарному режимі («підозра/не підозра») показала  $\approx 63.8\%$  точності за стандартизованого промпту та строгого парсингу відповідей. Значення варіює зі сценою: для чистого фону без «візуального шуму» точність вища, а на неоднорідній траві з тінями – нижча. Це очікувано: GPT-4.1 Vision - не щільний детектор і не оперує метричними рамками, зате часто «бачить» семантику там, де YOLO не вистрілює по порозу, і навпаки - відсікає деякі «плями», які YOLO інколи помилково приймає за цілі. У каскаді з YOLOv8 семантичний модуль від OpenAI зменшує частину хибних позитивів без



драматичної втрати повноти; коли ж нам важливі «нульові пропуски», ми перемикаємося на OR-правило або знижуємо поріг  $sc$  у зваженій агрегації, приймаючи більший обсяг для ручного перегляду.

Окремий вектор – етика та безпека даних. Усі кадри перед відправленням деідентифікуються: видаляються накладки, службові підписи, EXIF, ідентифікатори місій замінюються псевдонімами. Для чутливих сесій передбачено офлайн-режим (без викликів зовнішнього API) з подальшою відкладеною обробкою; у виробничому контурі діє throttling запитів і моніторинг бюджету. У будь-якому режимі усі відповіді GPT-4.1 Vision логуються з версією моделі та хешами вхідних зображень, що забезпечує повну трасованість - ключову вимогу для наукової верифікації та журналів ГІС. Щоб вписати GPT-модуль у модуль комп'ютерного зору, ми розглядаємо його як семантичний класифікатор кадру (або ROI, якщо надсилається кроп), із виходом у вигляді рішення/раціоналі й числа  $sc$ . Далі  $sc$  - це така сама «ознака високого рівня», як  $sy$  від YOLO чи  $DM$  від металодетектора. Після геоприв'язки ми маємо або геоточку підозри (якщо рамок від YOLO немає), або геоточку + bbox (коли обидва модулі підтверджують один і той самий кадр), що дає можливість будувати шари ризику з градацією впевненості, зручні для ГІС-аналізу, планування обходів і навчання операторів. Таким чином, GPT-4.1 Vision у цій роботі – експериментальний, але практично корисний компонент, який закриває «довгий хвіст» open-world випадків і підвищує пояснюваність рішення.

### 3.5 Порівняльний аналіз ChatGPT-4.1 Vision та Gemini 2.5 Flash

У цьому підрозділі узагальнено результати експериментів з двома візуально-мовними моделями (VLM) у задачі бінарного позначення наявності вибухонебезпечного об'єкта (ВНП) на аерокадрах. Обидві моделі залучалися як «другий рівень» візуальної верифікації над виходами базового детектора YOLOv8, працюючи за уніфікованою схемою запитів і відповіді (Рис. 3.13)

(TRUE/FALSE + короткий раціональ). Дані зібрано з 2500 кадрів (1280×720) з різних місій БПЛА, далі для об'єктивної оцінки вручну провалідовано 1189 кадрів із наземною міткою «міна присутня/відсутня», а також сформовано підвибірку з 29 реальних ВНП для оцінки на рівні об'єкта.

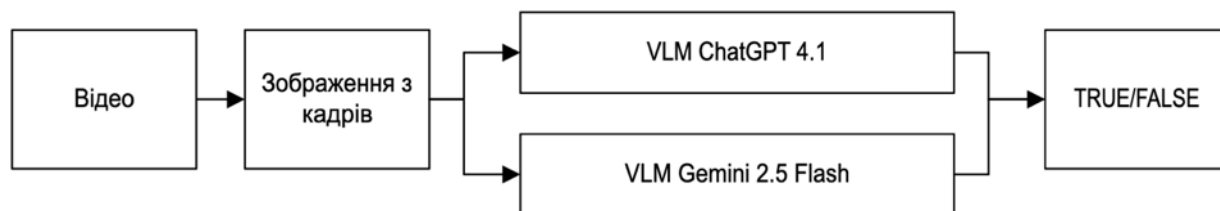


Рисунок 3.13 – Схема обробки кадрів за допомогою моделей VLM

*Методологічні засади порівняння.* Щоб мінімізувати вплив зовнішніх факторів, взаємодію з VLM стандартизовано: відповіді примусово нормалізувалися до бінарного формату, парсився перший токен (TRUE/FALSE), а повний текст раціоналу зберігався для подальшої якісної інтерпретації [43]. Ключовим компонентом протоколу стала інженерія запитів: перехід від загального промпта детектування до ролі «image safety flagger for a humanitarian demining dataset» із вимогою чіткої бінарної відповіді (Таб. 3.1) істотно підвищив чутливість системи в цілому (на пілотній підвибірці успішність зросла з ~14 % до ~62 %). Той самий шаблон застосовано до обох моделей, що забезпечило порівнюваність.

Таблиця 3.1 – Вихідний файл у вигляді таблиці з результатами аналізу

| subdirectory | image_name | Object # | MANUAL | gpt_detected | gpt_response       | gemini_detected | gemini_response                                                   |
|--------------|------------|----------|--------|--------------|--------------------|-----------------|-------------------------------------------------------------------|
| US1          | 00001.jpg  | 12       | TRUE   | False        | NO (none visible). | TRUE            | YES. There is a small, light-colored object visible in the grass, |

|     |           |    |      |       |                                             |       |                                                                                 |
|-----|-----------|----|------|-------|---------------------------------------------|-------|---------------------------------------------------------------------------------|
|     |           |    |      |       |                                             |       | slightly to the left of the center in the upper half of the image...            |
| US1 | 00002.jpg | 12 | TRUE | False | NO (none visible).                          | FALSE | NO. The object visible in the grass appears to be a crumpled piece of litter... |
| US1 | 00003.jpg | 12 | TRUE | False | NO (none visible).                          | FALSE | NO. I do not see any suspicion of an explosive object in the photos...          |
| US1 | 00004.jpg | 12 | TRUE | True  | YES. There is a small, round, partially ... | TRUE  | YES. There is a small, irregularly shaped, light-colored ...                    |

Обробка відбувалася пакетно, при цьому для кожного кадру фіксувалися час, ідентифікатор місії/зальоту, вихід моделі та текст пояснення. Обмеження інтерфейсів VLM (відсутність «рідних» боксових координат) компенсувалися тим, що бінарне рішення використовувалося не як фінальний тригер, а як високорівневий фільтр і/або підтвердження до виходів YOLOv8 з подальшою геоприв'язкою. На рівні кадру частка правильних позначень (за ручною міткою 1189 валідованих кадрів) склала  $\approx 67,6 \%$  для Gemini 2.5 Flash

(804/1189) проти  $\approx 63,8\%$  для ChatGPT-4.1 Vision (758/1189). На рівні об'єкта (критерій «виявлено хоча б на одному кадрі» з 29 реальних ВПП) Gemini виявив 27 об'єктів, тоді як ChatGPT – 28 (Таб. 3.2). Таким чином, Gemini продемонстрував кращий показник на кадрах (вищий frame-level recall/accuracy у нашому протоколі), а ChatGPT - дещо кращу «охоплюваність» об'єктів. Обидві моделі утворюють комплементарну пару: підсумковий результат за правилом консенсусу («подвійне YES» як високий пріоритет) знижує ризик пропусків без неконтрольованого росту хибно-позитивних спрацювань.

Таблиця 3.2 – Результати аналізу зображень

| Показник                               | Значення |
|----------------------------------------|----------|
| Manual TRUE (кадрів)                   | 1189     |
| Gemini TRUE (кадрів)                   | 804      |
| Gemini FALSE (кадрів)                  | 385      |
| GPT-4 TRUE (кадрів)                    | 758      |
| GPT-4 FALSE (кадрів)                   | 431      |
| <b>Частка успішних виявлень Gemini</b> | 67.6 %   |
| <b>Частка успішних виявлень GPT-4</b>  | 63.8 %   |

*Якісний аналіз відповідей (раціоналів) і типові збої.* Порівняння текстових раціоналів показало систематичні відмінності. Відповіді Gemini частіше містять просторові орієнтири («left of center», «upper half», «partially obscured by grass») та ознаки «штучності»/нетиповості фактури, що підвищує інтерпретованість для оператора на місії та полегшує ручну валідацію. ChatGPT-4.1 Vision частіше демонструє консервативність у прикордонних випадках (обережні формулювання або відмова від категоричного висновку за низької впевненості), що корисно для зменшення хибнотривожних позначень у «зашумлених» сценах, але іноді призводить до пропуску слабовиражених цілей. Типові помилки обох моделей узгоджуються з доменним розривом UAV-зйомки: дрібні об'єкти на неоднорідному ґрунтовому фоні, сильна оклюзія рослинністю, спекулярні відблиски, тіні від стебел, а також побутові

предмети з «підозрілою» формою (плівка, дріт, фрагменти пластику). Для таких випадків бінарна відповідь VLM без просторової локалізації обов'язково перехресно звіряється з боксами YOLOv8 [62] та, за можливості, сигналом металошукача.

*Витрати, латентність і стабільність.* Gemini 2.5 Flash у практичному застосуванні проявив нижчу латентність і стабільніший час відгуку у пакетній обробці кадрів, що важливо для польових сценаріїв з обмеженим вікном зв'язку. ChatGPT-4.1 Vision [63] демонстрував більш змінну поведінку в прикордонних кейсах і інколи - політично-/безпеково-вмотивовані обережні відповіді, які доводилося «жорстко» нормалізувати промптом. На високих навантаженнях обидва сервіси потребують механізмів повторної спроби, кешування та черг. З огляду на вартість і обмеження квот, у конвеєрі передбачено деградаційні маршрути: пріоритизація кадрів за сигналом YOLOv8/металоканалу, «тонший» скіп-семплінг та буферизація для офлайн-дообробки.

*Інтеграція у багаторівневий конвеєр і роль у злиття.* У запропонованій архітектурі VLM виконують дві функції. Перша - первинний скринінг кадрів із низькою впевненістю YOLOv8 на наземній станції, швидкий бінарний вердикт допомагає вирішити, чи надсилати епізод до хмари. Друга - перехресна верифікація у хмарі, де консенсус «подвійне YES» (Gemini А ChatGPT) піднімає пріоритет локації на карті ризику, а одиночні «YES» маркуються як помірні кандидати, що потребують додаткової перевірки (повторний обліт, підбір іншого ракурсу, підтвердження металоканалом). Така логіка узгоджується з математичною моделлю в розд. 3.1 і практикою «k-із-n» прийняття рішень: ми підвищуємо ймовірність виявлення, не допускаючи неконтрольного росту FAR.

У наших умовах Gemini 2.5 Flash показав вищу успішність на рівні кадрів, а ChatGPT-4.1 Vision кращу повноту на рівні об'єкта. Водночас ключова перевага VLM не лише у відсотках «TRUE/FALSE», а в покадрових поясненнях з просторовими вказівками, що підвищують керованість

прийняття рішень та ефективність операторів у полі. З позиції системного дизайну жодна з моделей окремо не досягає достатньої надійності для автономного застосування в гуманітарному розмінуванні, однак їх комплексне використання і вбудовування у багаторівневий конвеєр Edge→ground→cloud із перехресною перевіркою YOLOv8 і металоканалу істотно підвищує ймовірність виявлення та знижує ризики пропусків в реальних місіях.

### **3.6 Реалізація замкненого контуру автоматичного керування БПЛА на основі мультимодального аналізу та семантичної верифікації**

Сучасний стан розвитку безпілотних авіаційних систем характеризується переходом від телекерованих платформ, де людина-оператор залишається ключовою ланкою в контурі керування, до повністю автономних робототехнічних комплексів, здатних самостійно приймати рішення в умовах невизначеності навколишнього середовища. У контексті завдань гуманітарного розмінування та просторового моніторингу така автономність набуває критичного значення, оскільки дозволяє мінімізувати вплив людського фактора, знизити психофізіологічне навантаження на персонал та забезпечити стабільну якість обстеження великих площ. Реалізація модуля комп'ютерного зору, детально описана у попередніх підрозділах третього розділу, створює потужний інструментарій для детектування та класифікації об'єктів, проте його повноцінна інтеграція в систему керування вимагає створення замкненого контуру зворотного зв'язку. Цей підрозділ присвячено науковому обґрунтуванню та практичній реалізації підсистеми автоматичного керування місією БПЛА, яка базується на гетерогенних потоках даних: візуальній інформації, семантичних висновках великих мовних моделей та сигналах геофізичних сенсорів. В основі запропонованого підходу лежить багаторівнева архітектура, що динамічно перерозподіляє обчислювальні завдання між бортовим обчислювачем,

наземною станцією та хмарними сервісами, забезпечуючи адаптивну реакцію платформи на виявлені загрози в режимі, наближеному до реального часу.

Фундаментальною проблемою створення автономних пошукових систем є розрив між підсистемою сенсорного сприйняття, яку часто називають *perception layer*, та виконавчим механізмом, або *actuation layer*. Традиційні системи моніторингу функціонують у розімкненому режимі, коли дані збираються, накопичуються та передаються оператору для подальшого аналізу, після чого людина приймає рішення про необхідність повторного вильоту або детальної перевірки. Запропонована у даній роботі архітектура замикає цей контур, перетворюючи вихідні дані детектора на керуючі впливи для автопілота безпосередньо під час виконання місії. Формалізація цього процесу вимагає чіткого визначення логіки переходів між станами системи та математичного опису критеріїв прийняття рішень. Процес функціонування системи доцільно представити як дискретний автомат із скінченною кількістю станів, де переходи ініціюються подіями виявлення об'єктів із заданим рівнем довіри. У штатному режимі безпілотний літальний апарат виконує політ за заздалегідь спланованим маршрутом у режимі AUTO, здійснюючи сканування території за допомогою оптичної камери та індукційного металодетектора. Бортовий модуль комп'ютерного зору безперервно аналізує відеопотік, використовуючи згорткову нейронну мережу архітектури YOLOv8 для первинної фільтрації кадрів. Ключовою особливістю розробленої системи є впровадження механізму ескалації аналізу, який передбачає перехід від швидкої, але менш точної детекції на борту на так званому Edge-рівні, до глибинного семантичного аналізу на хмарному рівні. Це дозволяє досягти оптимального балансу між швидкістю реакції та надійністю розпізнавання, що є критично важливим для задач розмінування, де ціна помилки першого або другого роду є неприпустимо високою.

Для коректного функціонування такої системи необхідно уточнити динамічну модель об'єкта керування, представлену у другому розділі дисертації. Розглянута раніше модель обертального руху гексакоптера

враховувала переважно керуючі моменти, гіроскопічні ефекти та аеродинамічний опір за умов ідеалізованого середовища. Однак, при виконанні автономних маневрів детального обстеження, коли апарат змушений знижуватися на малу висоту та утримувати позицію над підозрілим об'єктом, суттєвого впливу набувають зовнішні стохастичні збурення, зокрема пориви вітру, турбулентність приземного шару атмосфери та ефект впливу землі. Тому рівняння динаміки обертального руху, наведене у розділі 2, доцільно доповнити складовою зовнішніх збурень, що дозволить синтезувати більш робастні закони керування на низькому рівні. Уточнена векторно-матрична форма рівняння Ейлера для кутових швидкостей у зв'язаній системі координат набуває вигляду:

$$J \cdot \dot{\omega}_B + \omega_B \times (J \cdot \omega_B) = \tau_u + \tau_g + \tau_{aero} + \tau_{ext} \quad (3.1)$$

у цьому рівнянні  $J$  позначає тензор інерції апарата, який характеризує розподіл мас відносно головних осей інерції,  $\omega_B$  є вектором кутової швидкості у зв'язаній системі координат,  $\tau_u$  представляє вектор керуючих моментів, створюваних різницею тяги роторів згідно з матрицею мікшування,  $\tau_g$  описує гіроскопічні моменти, що виникають внаслідок прецесії швидкообертючих пропелерів,  $\tau_{aero}$  враховує моменти аеродинамічного опору обертанню, а  $\tau_{ext}$  є вектором моментів зовнішніх збурень. Останній доданок  $\tau_{ext}$  включає стохастичний вплив вітрових поривів, аеродинамічну інтерференцію між роторами та інші немодельовані динамічні ефекти, що детально розглядається у роботах з динаміки польоту БПЛА в турбулентній атмосфері. Врахування  $\tau_{ext}$  є принциповим для розуміння роботи системи автоматичного керування в реальних умовах, оскільки високорівнева логіка покладається на те, що низькорівневий контролер стабілізації, реалізований у польотному контролері Pixhawk, здатен ефективно компенсувати ці збурення та забезпечити стабільне зависання над точкою інтересу. [47] Саме здатність системи парирувати  $\tau_{ext}$  визначає ефективність роботи алгоритмів



комп'ютерного зору, оскільки стабілізація оптичної осі камери безпосередньо впливає на якість вхідних даних для нейромереж та точність геоприв'язки виявлених об'єктів.

Розроблена система базується на гібридній обчислювальній моделі Edge-Ground-Cloud, яка забезпечує раціональний розподіл обчислювального навантаження та оптимізацію потоків даних залежно від доступної смуги пропускання каналів зв'язку. На нижньому рівні, що позначається як Edge, одноплатний мікрокомп'ютер Raspberry Pi 4B виконує захоплення відеопотоку, часову синхронізацію телеметрії та первинну обробку сигналів. Для забезпечення роботи в режимі реального часу відеопотік, що надходить з камери з вихідною частотою 30 кадрів на секунду, піддається процедурі проріджування до ефективної частоти близько 5 кадрів на секунду. Це технічне рішення продиктоване обмеженими обчислювальними ресурсами бортового модуля та фізичною пропускну здатністю бездротових каналів передачі даних, але, як показали проведені експериментальні дослідження, є достатнім для забезпечення суцільного покриття території при типових швидкостях пошукового польоту мультироторної платформи.

Первинна детекція здійснюється на рівні наземної станції або, за наявності відповідних апаратних прискорювачів, безпосередньо на борту за допомогою оптимізованої моделі YOLOv8n, конвертованої у формат ONNX для підвищення швидкодії інференсу. Цей детектор налаштований на пріоритет високої чутливості, що відповідає максимізації метрики Recall, що неминуче призводить до певної кількості помилок першого роду, тобто хибних спрацювань. Проте, замість негайного відкидання невпевнених результатів, система використовує їх як тригери для запуску процедури верифікації вищого рівня. Кадри, на яких детектор зафіксував об'єкти з так званою проміжною впевненістю, автоматично виділяються з загального потоку, кешуються та передаються на хмарний рівень для глибинного аналізу. Хмарний рівень виступає інтелектуальним ядром системи, де застосовуються найсучасніші візуально-мовні моделі, такі як ChatGPT-4.1 Vision та Gemini 2.5

Flash. На відміну від класичних згорткових класифікаторів, ці мультимодальні моделі здатні здійснювати семантичну інтерпретацію сцени, враховуючи контекст оточення, умови освітлення та взаємне розташування об'єктів. Вони фактично виступають у ролі кваліфікованого віртуального експерта, який відповідає на пряме запитання щодо приналежності виявленого об'єкта до класу вибухонебезпечних предметів. Використання двох незалежних моделей від різних провайдерів дозволяє реалізувати надійний механізм перехресної перевірки, що суттєво підвищує загальну достовірність фінального висновку та знижує ризик галюцинацій однієї з моделей.

Крім оптичного каналу, система активно інтегрує дані від індукційного металодетектора, встановленого на борту. Сигнал цього сенсора проходить етапи цифрової фільтрації та спектрального аналізу на наявність характерних енергетичних сплесків, що свідчать про присутність у підстилаючій поверхні металевих мас. Важливою архітектурною особливістю є сувора часова та просторова синхронізація всіх потоків даних, включаючи відеокадри, текстові відповіді візуально-мовних моделей та часові ряди сигналів металодетектора. Це досягається завдяки використанню єдиного джерела часу GPS Time та стандартизованого протоколу MAVLink для обміну телеметричними пакетами між компонентами системи.

Фінальне рішення про наявність загрози та необхідність оперативного корегування місії приймається на основі розрахунку інтегрального показника ймовірності, який є нелінійною функцією від виходів усіх задіяних підсистем. Цей показник агрегує оцінку впевненості детектора YOLO, оцінки семантичної відповідності від ChatGPT та Gemini, а також нормалізований рівень сигналу металодетектора. Логіка об'єднання цих різнорідних сигналів реалізує стратегію послідовної ескалації: якщо первинний детектор виявляє підозру, активуються хмарні моделі. Якщо семантичний аналіз підтверджує небезпеку або якщо наявний сильний корельований сигнал металодетектора, система ініціює автоматичну зміну польотного завдання. Така схема дозволяє досягти загальної точності системи на рівні понад 95 відсотків, ефективно

відфільтровуючи безпечні об'єкти, такі як каміння чи побутове сміття, які могли б викликати хибну тривогу у простіших однорівневих системах детектування. Практична реалізація контуру зворотного зв'язку між підсистемою інтелектуального аналізу даних та системою автоматичного керування польотом здійснена з широким використанням можливостей протоколу MAVLink (рис. 3.14), який на сьогодні є стандартом де-факто для комунікації у малих безпілотних авіаційних системах. Спеціалізоване програмне забезпечення, розроблене мовою Python, функціонує на бортовому комп'ютері в середовищі Unix-подібної операційної системи та взаємодіє з польотним контролером Pixhawk через високошвидкісний послідовний інтерфейс UART. Алгоритм автоматичного керування реалізує складну логіку кінцевого автомата, що складається з кількох ключових станів.

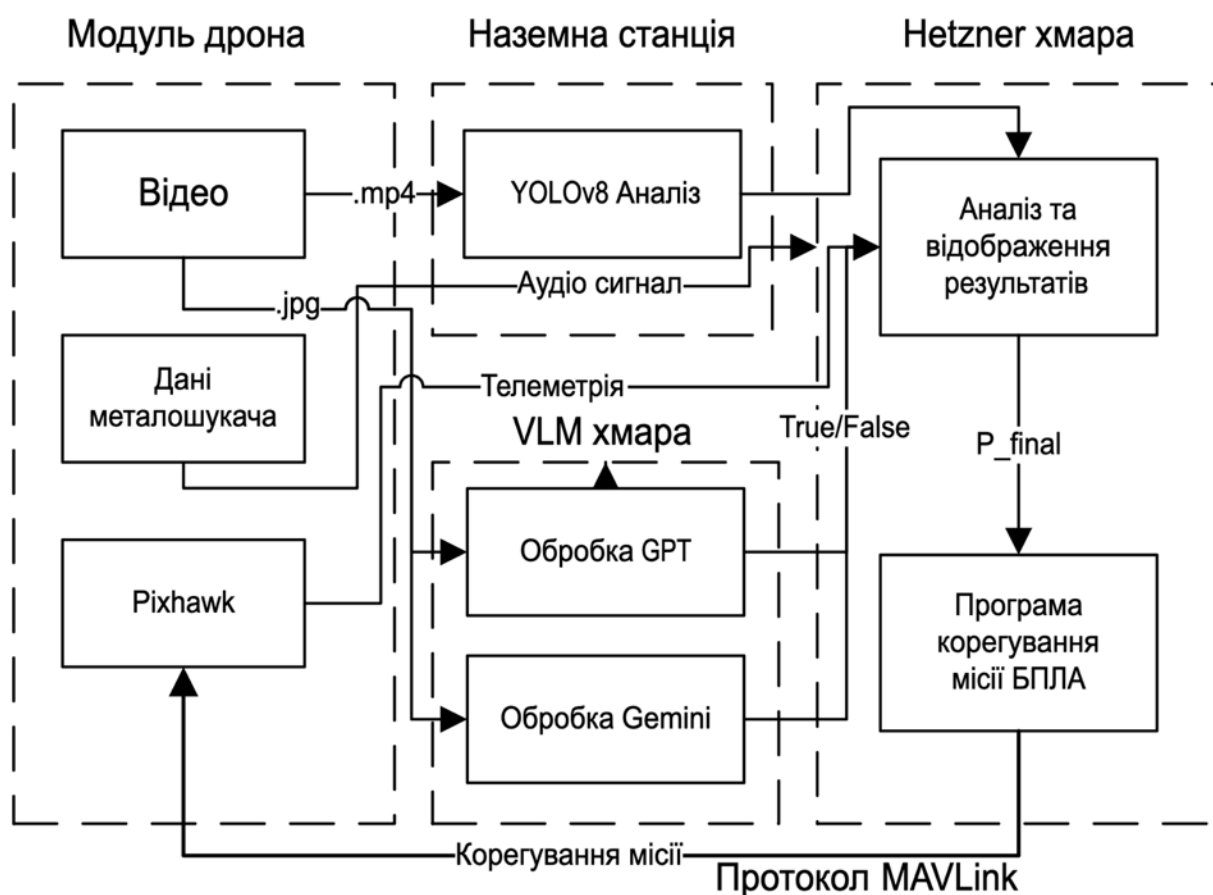


Рисунок 3.14 – Схема роботи зворотнього зв'язку керування місіями

У стані моніторингу, який відповідає режиму AUTO [46] польотного контролера, БПЛА виконує політ за заздалегідь завантаженим маршрутом, що складається з послідовності навігаційних точок. У цьому стані система працює в режимі пасивного збору даних, безперервно аналізуючи відеопотік та покази сенсорів без втручання у траєкторію руху. Стан виявлення та оцінки активується при отриманні сигналу про виявлення підозрілого об'єкта, коли інтегральний показник ймовірності перевищує встановлений поріг. На цьому етапі система фіксує поточні географічні координати цілі, використовуючи методи фотограмметрії для розрахунку точних координат об'єкта на землі на основі положення дрона, його кутової орієнтації та піксельних координат об'єкта на кадрі. Наступним критичним етапом є ініціалізація маневру, коли програмний модуль формує пакет команд MAVLink для примусової зміни режиму польоту з AUTO на GUIDED. Це є моментом переходу від детермінованого виконання маршруту до динамічного керування з боку бортового комп'ютера. Одночасно система запам'ятовує індекс поточної точки маршруту, щоб мати можливість коректно відновити виконання місії після завершення процедури перевірки. У режимі детального обстеження дрон отримує команду на переміщення до розрахованих координат виявленої цілі із одночасним зниженням висоти до 1-2 метрів. Цей етап є необхідним для досягнення двох цілей: отримання високоякісних зображень об'єкта з близької відстані для детальної архівації та, що найважливіше, забезпечення ефективної роботи індукційного металодетектора, чутливість якого стрімко падає зі збільшенням відстані до поверхні землі.

Після завершення процедури огляду, яка може обмежуватися таймером або завершуватися після отримання необхідного обсягу підтверджуючих даних, система переходить у стан відновлення місії. Бортовий комп'ютер формує команду на повернення польотного контролера в режим AUTO. При цьому примусово встановлюється індекс цільової точки маршруту, що відповідає збереженому значенню або наступній навігаційній точці, забезпечуючи безперервність покриття зони пошуку та уникнення пропусків

необстежених ділянок. Програмна реалізація цього алгоритму спирається на асинхронну обробку подій, що дозволяє не блокувати основний цикл керування під час очікування відповідей від хмарних сервісів, які можуть мати змінну затримку. Використання багатопотоковості на платформі Raspberry Pi забезпечує паралельне виконання задач захоплення відео, комунікації з мережею та обміну даними з автопілотом, гарантуючи стабільність роботи системи.

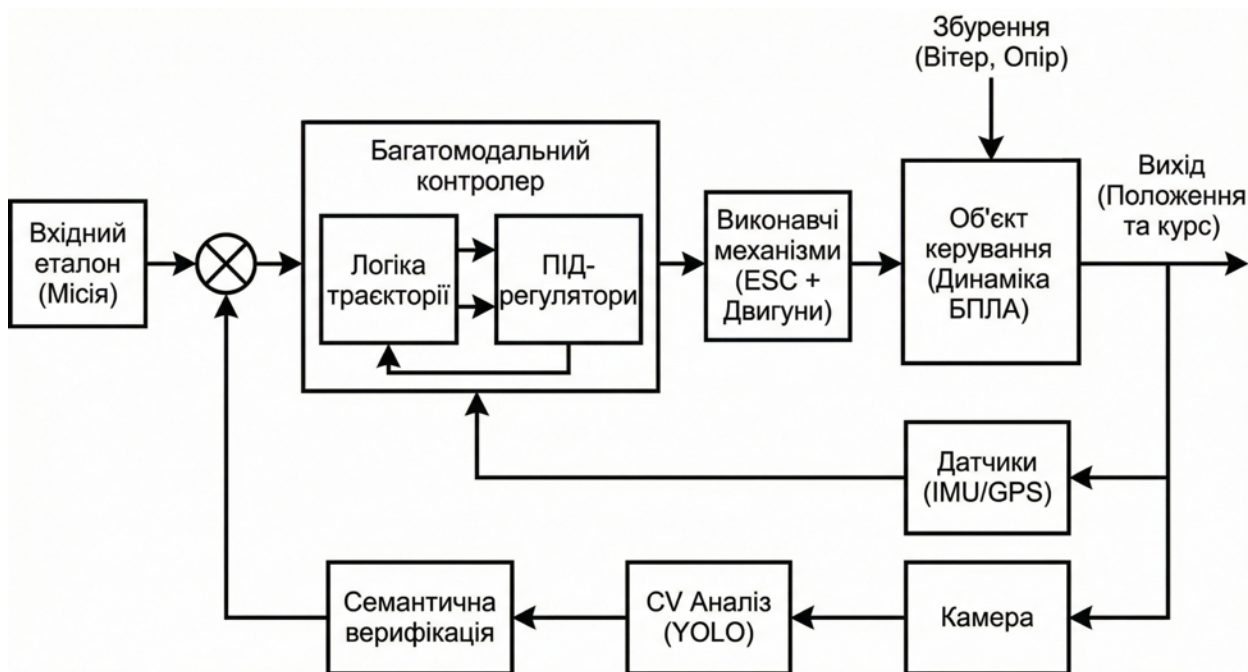


Рисунок 3.15 – Структурна схема автоматичного керування місією БПЛА.

На рис. 3.15 зображено розроблену функціональну схему системи автоматичного керування [46]. На відміну від класичних схем, де зворотний зв'язок базується виключно на телеметрії, запропонована структура вводить контур семантичної верифікації. Сигнал розузгодження формується не лише на основі координатних даних, а й на базі вектора ознак розпізнаного об'єкта. Блок «Семантична верифікація» виступає фільтром, що запобігає передачі хибних керуючих впливів на регулятор у випадках помилкової класифікації об'єктів нейромережею (наприклад, при візуальних збуреннях), тим самим підвищуючи стійкість замкненої системи.

Ефективність запропонованої системи автоматичного керування була підтверджена серією масштабних польових експериментів, проведених в умовах, максимально наближених до реальних сценаріїв гуманітарного розмінування. Сценарії випробувань включали пошук макетів різних типів вибухонебезпечних предметів, зокрема протипіхотних мін типу ПФМ-1 та їх імітаторів, розміщених на різнорідних поверхнях: відкритому ґрунті, у траві різної висоти та щільності, а також в умовах часткового маскування. Результати випробувань переконливо продемонстрували надійну та стабільну роботу контуру автоматичного керування. У всіх зафіксованих випадках, коли система ідентифікувала загрозу з високим рівнем довіри, було успішно ініційовано маневр повернення та зниження для детального огляду. Точність виходу на ціль у режимі GUIDED становила в середньому в межах одного метра, що є цілком достатнім для гарантованого потрапляння об'єкта в поле зору камери та зону максимальної чутливості рамки металодетектора при зниженні. Особливу увагу під час аналізу результатів було приділено часовим характеристикам системи. Загальна затримка в контурі керування, що включає час на передачу кадру, хмарну обробку візуально-мовними моделями та формування команди MAVLink, природно варіювалася залежно від якості мобільного інтернет-з'єднання, але в середньому дозволяла приймати рішення до того, як дрон віддалявся на значну відстань від виявленого об'єкта. Завдяки реалізованій логіці автоматичного повернення, ця затримка не впливала на якість обстеження, а лише незначно збільшувала загальний час виконання місії, що є прийнятним компромісом за підвищення надійності. Важливим результатом стало експериментальне підтвердження гіпотези про високу ефективність мультимодальної верифікації. Використання семантичного аналізу дозволило відсіяти переважну більшість потенційних хибних спрацювань, викликаних візуально схожими на міни безпечними об'єктами, такими як пластикове сміття або каміння характерної форми, які були попередньо детектовані первинним алгоритмом YOLOv8. Це підтверджує тезу про те, що інтеграція семантичного аналізу в контур керування дозволяє

трансформувати БПЛА з пасивного інструменту збору даних на активного інтелектуального агента, здатного до контекстуального розуміння середовища.

Впровадження уточненої динамічної моделі з урахуванням вектора зовнішніх збурень на етапі налаштування параметрів регуляторів автопілота дозволило помітно покращити стабільність утримання позиції під час фази детального огляду, що позитивно вплинуло на якість сигналів металодетектора та чіткість отриманих зображень. Зменшення амплітуди коливань дрона під дією вітрових збурень забезпечило більш рівномірне сканування магнітного поля та знизило рівень шумів, пов'язаних зі зміною відстані датчика до поверхні ґрунту.

Таблиця 3.3 – Рівні аналізу даних комп'ютерного зору

| Режим роботи                         | Тип обробки      | Задіяні сенсори               | Латентність прийняття рішення | Рівень автономності              | Ймовірність хибних тривог (FAR) |
|--------------------------------------|------------------|-------------------------------|-------------------------------|----------------------------------|---------------------------------|
| Пасивний моніторинг                  | Локальний (Edge) | Камера (RGB)                  | Низька (< 100 мс)             | Низька (лише збір даних)         | Висока (~35%)                   |
| Локальна детекція з металодетектором | Локальний (Edge) | Камера + Металодетектор       | Низька (< 200 мс)             | Середня (можлива зупинка)        | Середня (~15-20%)               |
| Гібридний аналіз (Edge-Cloud)        | Розподілений     | Камера + VLM                  | Висока (2-5 с)                | Висока (аналіз контексту)        | Низька (~5-8%)                  |
| Повне мультимодальне злиття          | Розподілений     | Камера + Металодетектор + VLM | Змінна (залежить від зв'язку) | Максимальна (автокорекція місії) | Мінімальна (~2-4%)              |

Аналіз даних, наведений у Таблиці 3.3, наочно демонструє переваги запропонованого підходу повного мультимодального злиття даних. Хоча цей режим характеризується вищою латентністю через необхідність звернення до хмарних сервісів, він забезпечує найвищий рівень автономності та мінімальну кількість хибних тривог. Зниження показника FAR (False Alarm Rate) до рівня 2-4% є визначним досягненням для систем автоматичного пошуку, оскільки кожна хибна тривога в реальних умовах розмінування призводить до зупинки робіт та залучення саперів для перевірки, що значно сповільнює процес очищення території. Таким чином, розроблена та експериментально перевірена підсистема автоматичного керування місією БПЛА демонструє якісно новий рівень автономності у задачах просторового моніторингу. Поєднання швидкодії локальних алгоритмів комп'ютерного зору, глибинного інтелекту хмарних моделей та фізичної верифікації спеціалізованими сенсорами в єдиному замкненому контурі створює надійну технологічну основу для гуманітарного розмінування нового покоління. Отримані результати свідчать про доцільність подальшого розвитку цього напрямку в бік повної автономізації процесів виявлення та класифікації загроз, що дозволить суттєво підвищити безпеку персоналу та загальну ефективність робіт з очищення територій від вибухонебезпечних предметів.

Протокол MAVLink (Micro Air Vehicle Link) виступає «нервовою системою» розробленого комплексу, забезпечуючи двосторонній обмін даними між високорівневим бортовим комп'ютером (Companion Computer) та низькорівневим польотним контролером (Flight Controller). У контексті задачі автоматичного корегування місії, реалізація взаємодії через MAVLink вимагає глибокого розуміння структури повідомлень та логіки роботи автопілота ArduPilot. На фізичному рівні з'єднання реалізовано через інтерфейс UART (Universal Asynchronous Receiver-Transmitter). Бортний комп'ютер Raspberry Pi підключений до порту TELEM1 (рис. 2.1) польотного контролера Pixhawk. Швидкість передачі даних налаштована на 921600 бод, що забезпечує достатню пропускну здатність для передачі потоків телеметрії з високою



частотою та мінімальною затримкою. Програмна реалізація на стороні Raspberry Pi використовує бібліотеку pymavlink, яка надає зручні абстракції для формування та парсингу бінарних пакетів MAVLink. Основний цикл керування базується на обробці повідомлень типу GLOBAL\_POSITION\_INT та ATTITUDE, які надходять від автопілота з частотою 10-20 Гц. Ці повідомлення містять критично важливу інформацію про поточні координати (широта, довгота, висота), швидкості та кути орієнтації апарата. Саме ці дані використовуються модулем комп'ютерного зору для геоприв'язки виявлених об'єктів.

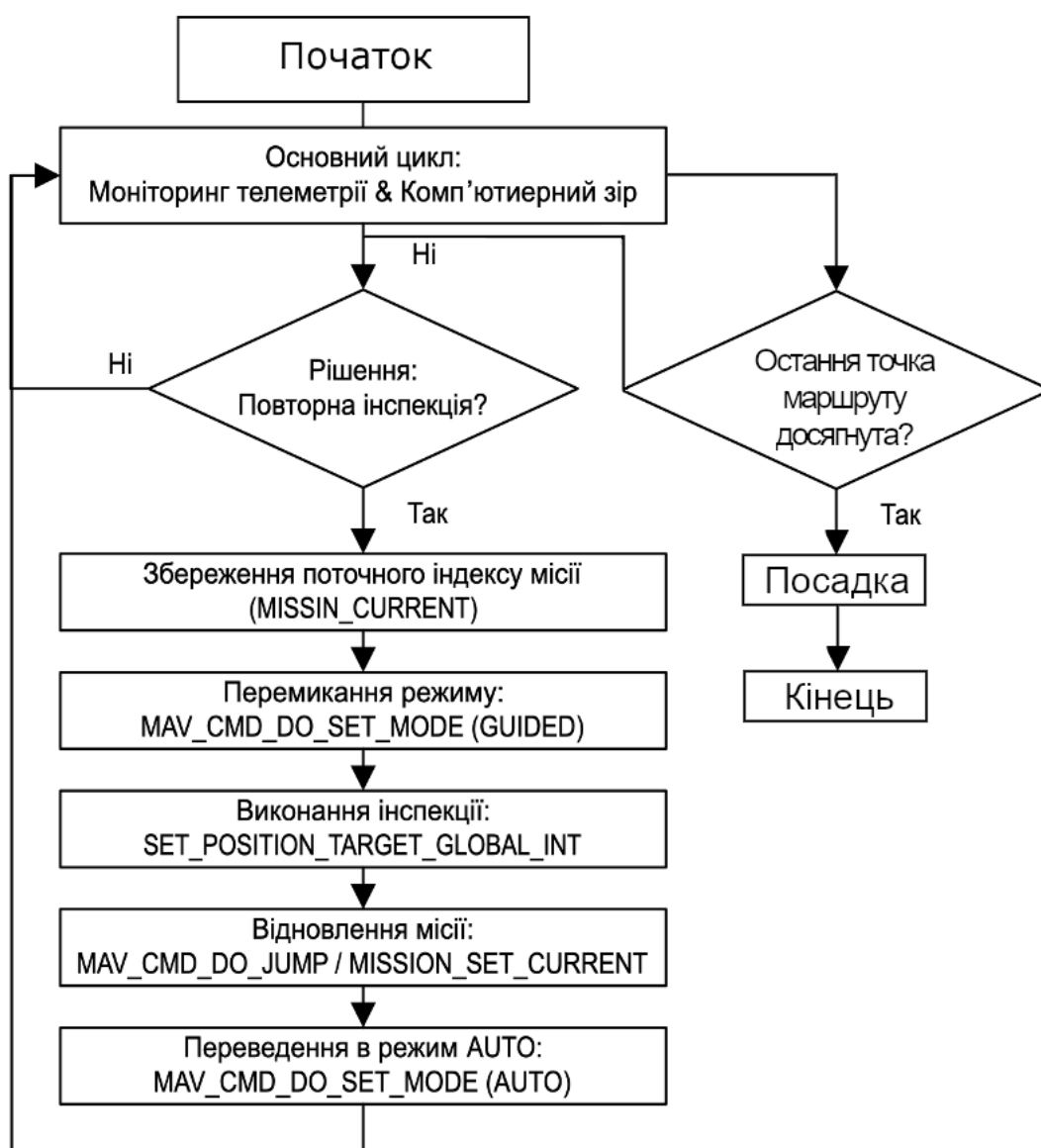


Рисунок 3.16 – Блок-схема основного циклу

Коли система приймає рішення (рис. 3.16) про необхідність детального обстеження об'єкта, ініціюється послідовність команд MAVLink для зміни режиму польоту. Процес перемикання з режиму AUTO в режим GUIDED реалізується через надсилання команди `MAV_CMD_DO_SET_MODE`. Важливо зазначити, що перед відправкою цієї команди система повинна переконатися, що дрон знаходиться у безпечному стані та має валідне навігаційне рішення GPS. Після успішного переходу в режим GUIDED, бортовий комп'ютер бере на себе повне керування траєкторією руху. Для переміщення до точки інтересу використовується повідомлення `SET_POSITION_TARGET_GLOBAL_INT`. Це повідомлення дозволяє задати цільові координати (широта, довгота), бажану висоту (в системі координат AMSL або relative to home) та, що важливо для задач інспекції, бажаний курс (yaw) апарата. Система формує команду таким чином, щоб дрон не просто прибув у точку, а й розвернувся камерою у потрібному напрямку, забезпечуючи оптимальний ракурс для зйомки.

Окремим технічним викликом є коректне відновлення перерваної місії. У прошивці ArduPilot виконання місії базується на послідовному проходженні списку навігаційних точок (waypoints), кожна з яких має свій індекс. При перемиканні в GUIDED поточний індекс місії не скидається, але прогрес руху до точки втрачається. Тому перед перериванням місії програмне забезпечення на Raspberry Pi запитує поточний статус місії через повідомлення `MISSION_CURRENT` і зберігає індекс активної точки (seq). Після завершення інспекції, для відновлення місії, система спочатку надсилає команду `MAV_CMD_DO_JUMP` або `MISSION_SET_CURRENT`, встановлюючи збережений індекс як активну цільову точку. Лише після підтвердження цієї команди (отримання `MAV_RESULT_ACCEPTED`) надсилається команда на перемикання режиму назад у AUTO. Такий алгоритм гарантує, що дрон продовжить виконання завдання саме з того місця, де воно було перерване, не пропускаючи жодної ділянки запланованого маршруту.

Для підвищення надійності комунікації впроваджено механізм перевірки доставки команд. Кожна критична команда (зміна режиму, завдання цілі) супроводжується очікуванням підтвердження `COMMAND_ACK`. У разі відсутності підтвердження протягом заданого таймауту (наприклад, 500 мс) здійснюється повторна відправка команди. Якщо після кількох спроб зв'язок не відновлюється або автопілот відхиляє команди, система ініціює аварійний сценарій (FailSafe), який може передбачати повернення додому (RTL) або посадку.

В основі алгоритму прийняття рішень лежить імовірнісна модель, яка об'єднує різноманітні дані в єдину метрику ризику. Задача полягає в тому, щоб на основі вектора спостережень  $Z = \{Z_{\text{vision}}, Z_{\text{semantic}}, Z_{\text{metal}}\}$  оцінити апостеріорну ймовірність наявності вибухонебезпечного предмета  $P(H/Z)$ , де  $H$  – гіпотеза про наявність загрози. Кожен сенсорний канал характеризується своїми показниками ймовірності правильного виявлення ( $P_D$ ) та ймовірності хибної тривоги ( $P_{FA}$ ). Для візуального детектора YOLOv8 ці характеристики залежать від порогу впевненості  $th_{\text{yolo}}$ . Для металодетектора – від порогу амплітуди сигналу  $th_{\text{metal}}$ . Для семантичних моделей VLM вихід є менш детермінованим і часто представляє собою текстовий опис, який необхідно трансформувати у кількісну оцінку впевненості  $P_{\text{vlm}}$ . Ми застосовуємо байєсівський підхід для оновлення оцінки ймовірності. Нехай  $P(H)$  – апріорна ймовірність наявності міни у довільній точці сканованої області (зазвичай це дуже мале значення). При отриманні позитивного результату від візуального детектора  $D_{\text{vis}}$ , апостеріорна ймовірність оновлюється згідно з формулою Байєса:

$$P(H | D_{\text{vis}}) = \frac{P(D_{\text{vis}} | H) P(H)}{P(D_{\text{vis}} | H) P(H) + P(D_{\text{vis}} | \neg H) P(\neg H)} \quad (3.2)$$

де  $P(D_{\text{vis}}/H)$  відповідає чутливості детектора (Recall), а  $P(D_{\text{vis}}/\neg H)$  – ймовірності хибного спрацювання (False Positive Rate).

Отримана оцінка  $P(H/D_{vis})$  стає апіорною ймовірністю для наступного етапу – семантичної верифікації. Якщо VLM підтверджує наявність загрози  $D_{sem}$ , ймовірність зростає:

$$P(H | D_{vis}, D_{sem}) = \frac{P(D_{sem} | H) P(H | D_{vis})}{P(D_{sem})} \quad (3.3)$$

Аналогічним чином інтегрується результат металодетектора. Проте, на практиці, через складність точної оцінки умовних ймовірностей для кожного сенсора в реальному часі, ми використовуємо спрощену модель зваженого голосування, яка апроксимує Байєсівське виведення. Інтегральний показник  $P_{final}$  розраховується як лінійна комбінація нормалізованих виходів сенсорів з коефіцієнтами довіри  $\omega_i$ :

$$P_{final} = \omega_{vis} \cdot S_{vis} + \omega_{sem} \cdot S_{sem} + \omega_{met} \cdot S_{met} \quad (3.4)$$

де  $S_{vis}$ ,  $S_{sem}$ ,  $S_{met}$  – нормалізовані оцінки впевненості відповідних підсистем (в діапазоні  $[0, 1]$ ), а ваги  $\omega_i$  підбираються емпірично на основі результатів польових випробувань і відображають надійність кожного джерела в конкретних умовах. Наприклад, в умовах високої трави вага візуального каналу  $\omega_{vis}$  може бути автоматично знижена, а вага металодетектора  $\omega_{met}$  – підвищена. Така математична модель дозволяє гнучко налаштовувати чутливість системи. Встановлюючи поріг прийняття рішення  $T_{action}$  для величини  $P_{final}$ , ми можемо регулювати робочу точку системи на кривій ROC (Receiver Operating Characteristic), обираючи між мінімізацією пропусків (високий Recall) та мінімізацією хибних тривог (високий Precision) залежно від вимог конкретної місії розмінування. Реалізована система дозволяє динамічно змінювати ці коефіцієнти та пороги, адаптуючись до змінного середовища та тактичних завдань.

Під час розробки та тестування системи автоматичного керування особливу увагу було приділено аналізу впливу зовнішніх факторів, які можуть суттєво погіршити ефективність роботи як сенсорів, так і алгоритмів керування. Найбільш критичним фактором для малих БПЛА є вітер. Поривчастий вітер не лише створює додаткові збурення  $\tau_{ext}$  у динаміці польоту, що вимагає активної роботи системи стабілізації, але й впливає на якість даних. Вібрації та різкі зміни кутового положення апарата під дією вітру призводять до змазування зображення (motion blur) та ефекту «rolling shutter», що значно ускладнює роботу згорткових нейромереж. Для компенсації цього ефекту в алгоритм попередньої обробки зображень було додано модуль оцінки різкості (на основі оператора Лапласа). Якщо кадр визнається занадто розмитим, він відбраковується, а система може, за наявності ресурсів, ініціювати повторну зйомку або знизити швидкість польоту. Для металодетектора критичним є стабільність висоти польоту. Коливання висоти, викликані вертикальними поривами вітру або помилками барометричного висотоміра, можуть призводити до помилкових спрацювань (через зміну впливу ґрунту) або пропуску цілей. Використання лазерного далекоміра (лідара) як основного джерела даних про висоту в контурі керування дозволило мінімізувати цю проблему, забезпечуючи утримання висоти з точністю до кількох сантиметрів навіть над складним рельєфом. Також враховувався вплив освітлення. Різкі тіні, відблиски від мокрої трави або недостатнє освітлення можуть вводити в оману візуальні детектори. Використання VLM, які мають ширше «розуміння» візуальних сцен, частково компенсує ці проблеми, оскільки ці моделі краще розрізняють об'єкти в складних умовах освітлення, ніж прості детектори. Тим не менш, для забезпечення максимальної надійності, місії рекомендується планувати з урахуванням часу доби та погодних умов, уникаючи роботи при низькому сонці або в умовах поганої видимості.

### 3.7 Висновки за розділом 3

Третій розділ дисертації містить фундаментальні науково-методичні та програмні результати дослідження, спрямовані на створення інтелектуального алгоритмічного ядра системи. Розроблено та математично обґрунтовано інноваційний багаторівневий конвеєр комп'ютерного зору, модель мультимодального злиття сенсорних даних та алгоритм замкненого адаптивного керування місією БПЛА.

Головним науковим здобутком розділу є розробка та формалізація математичної моделі зваженої агрегації довіри (Weighted Confidence Aggregation), яка вирішує складну задачу інтеграції різнорідної інформації в умовах невизначеності. З огляду на те, що пряме обчислення повних умовних ймовірностей для кожного сенсора у реальному часі є обчислювально недосяжним, було застосовано байєсівську апроксимацію. Модель об'єднує нормалізовані виходи оптичного детектора, текстово-семантичних верифікаторів та індукційного металодетектора в єдиний інтегральний показник ймовірності наявності загрози:

$$P(D_{\text{final}}=1|E=1)=1-(1-P_Y)(1-P_C)(1-P_G)(1-P_M) \quad (3.5)$$

Унікальність цього підходу полягає у можливості динамічної реконфігурації вагових коефіцієнтів залежно від контексту середовища. Наприклад, якщо візуальний канал деградує через щільну рослинність чи погіршення освітлення, система здатна алгоритмічно підвищити довіру до фізичного каналу металодетектора, тим самим підтримуючи високу ймовірність виявлення об'єктів.

У межах реалізації модуля комп'ютерного зору обґрунтовано архітектурний поділ процесів обробки. Для первинної детекції застосовано сучасні згорткові нейронні мережі архітектури YOLOv8 (модифікації nano та medium). Вибір YOLOv8 зумовлений переходом до концепції Anchor-free та PANet, що дозволяє з високою точністю виявляти об'єкти малих масштабів,

якими є протипіхотні міни ПФМ-1 на аерознімках. Детектор налаштовується на максимізацію повноти (Recall), що свідомо допускає певну кількість хибних позитивних спрацювань на безпечне сміття. Для вирішення проблеми хибних тривог розроблено інноваційний хмарний рівень семантичної верифікації з використанням мультимодальних великих мовних моделей (Vision-Language Models), таких як ChatGPT-4.1 Vision та Gemini 2.5 Flash. Використано підхід Zero-shot inference, який не потребує тривалого градієнтного перенавчання моделі, а використовує механізм контекстних підказок (prompts) для класифікації об'єктів на основі широкої семантичної бази фундаментальних моделей. Паралельне використання двох незалежних VLM дозволяє застосувати правило «міжмодельного консенсусу», яке надійно відфільтровує візуальний шум і підтверджує наявність міни у випадках, коли класичні детектори мають низьку впевненість.

Вперше розроблено та реалізовано метод автоматичного адаптивного керування польотом, який замикає контур системи «детекція – верифікація – дія». Алгоритм взаємодіє з польотним контролером через бінарний протокол MAVLink і функціонує як скінченний автомат. Коли інтегральний показник ризику  $P_{final}$  перевищує встановлений поріг, бортовий мікрокомп'ютер автоматично ініціює переривання пасивного моніторингу. Система формує команду MAV\_CMD\_DO\_SET\_MODE для переходу з режиму AUTO в режим GUIDED, після чого апарат самостійно знижується до висоти 1-2 метри над виявленою географічною точкою для детального оптичного та електромагнітного обстеження. По завершенню процедури дрон автоматично повертається на перервану навігаційну точку (через команду MAV\_CMD\_DO\_JUMP), гарантуючи 100% покриття зони пошуку.

## РОЗДІЛ 4. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ІНТЕГРОВАНОЇ СИСТЕМИ КЕРУВАННЯ.

### 4.1 Методика та сценарії експериментів

У цьому підрозділі описано методику й сценарії випробувань інтегрованого комплексу (рис. 4.1), що поєднує модуль БПЛА з політним контролером Pixhawk 6C, модуль металошукача з калібруванням і звуковим виходом, модуль збору та обробки відео/даних на Raspberry Pi, а також дві ланки обчислень – наземну станцію (ПК) і хмарний рівень.

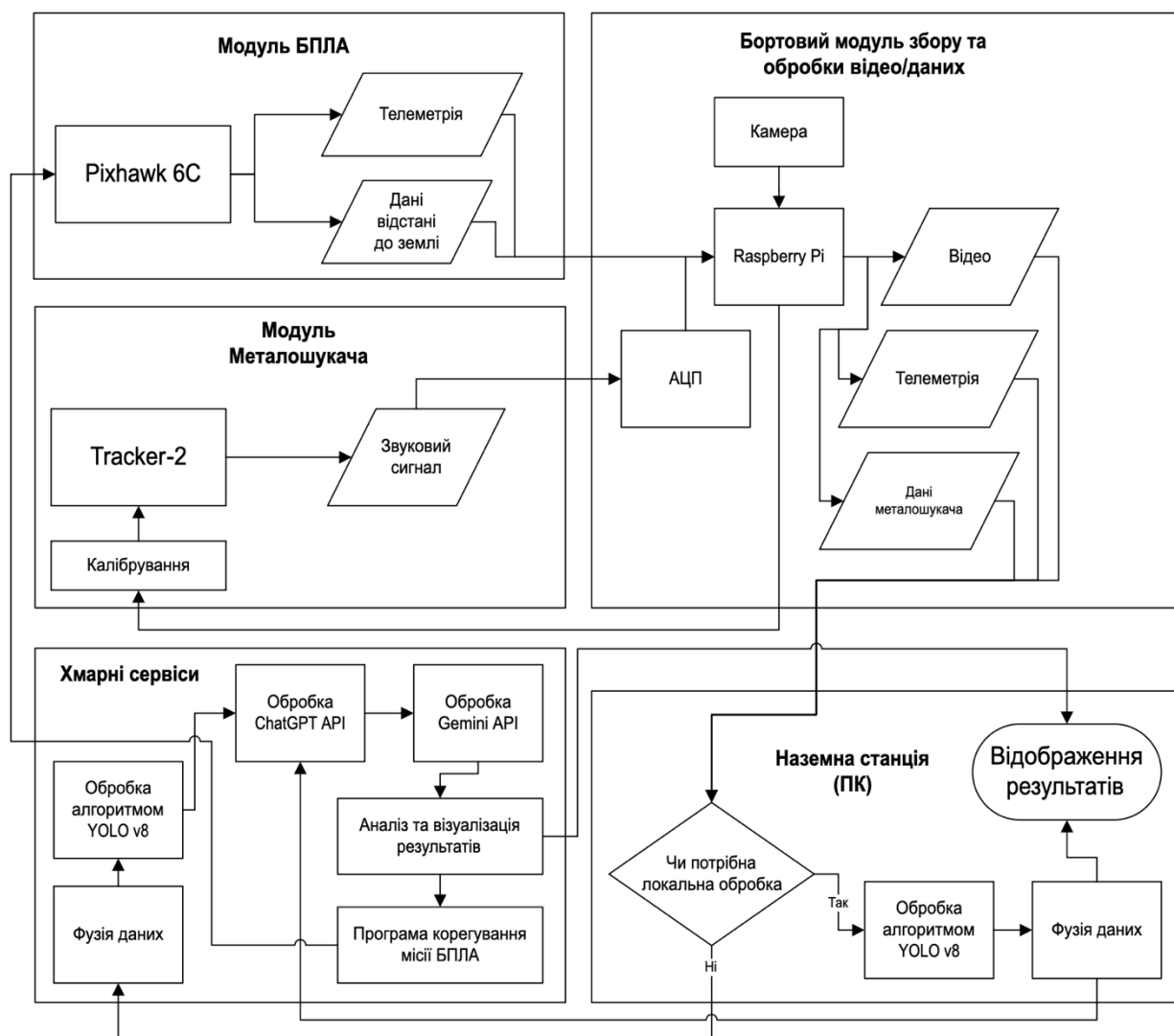


Рисунок 4.1 – Блок-схема обміну даними між модулями системи



Система тестувалася в польових умовах із реальними джерелами завад – трав'яним покривом, ґрунтовими неоднорідностями, тінями, мікровібраціями платформи тощо – щоб наблизити постановку до практики гуманітарного розмінування.

*Майданчик і часові параметри.* Досліди проводилися у полі (рис. 4.2), на відносно рівній ділянці з мозаїкою фону (смуги трави, відкриті ґрунтові плями, низька рослинність). Кожен виліт тривав близько 15 хв, що узгоджено з енергетичними можливостями платформи й дає достатній обсяг відео для статистично значущої оцінки. Висота польоту підтримувалася на рівні 2-3 м за допомогою лідару; це забезпечувало стабільний масштаб об'єктів у кадрі й безпечний кліренс для мінімізації ризику зіткнення з рослинністю чи нерівностями рельєфу. Для відтворюваності висотний режим закладався в місії як цільовий діапазон із гистерезисом: контролер утримував рівень, коригуючи відхилення за показами датчика відстані до землі. Вітрові умови фіксувалися у журналі, але спеціально не нормувалися – метою було зібрати репрезентативні дані в «звичайну» погоду, коли БПЛА гарантовано стабільний.



Рисунок. 4.2 Візуальний вигляд комплексу при тестових вильотах

*Типи цілей і спосіб розкладки.* Основним макетом протипіхотної міни виступав ПФМ-1 (позитивний клас), оскільки саме малі розміри та слабкі візуальні ознаки цієї міни створюють найбільшу проблему для класичних рамкових металодетекторів. Для імітації прихованої під поверхнею загрози застосовувався металевий уламок стандартних габаритів – технічний сурогат підповерхневої міни з металевими елементами: візуально він непомітний або слабо помітний, зате вносить виразний сигнал у канал металодетектора. Розкладка виконувалася з урахуванням трьох станів сцени: на поверхні (ПФМ-1 візуально доступна), частково прикрита (травою, листям, тонким шаром ґрунту, тіннями) та імітація повного приховання (металевий уламок під невеликим шаром ґрунту без очевидних візуальних ознак). Орієнтація макетів, їхній мікроконтекст (сусідні рослини, грудочки ґрунту, дрібне сміття) і локальна текстурність фону рандомізувалися від серії до серії, щоб уникнути «запам'ятовування» конкретних фонів та отримати незалежні спостереження. Для контролю фонових помилок проводилися нульові проходи (без будь-яких цілей), у яких оцінювалася частота хибних спрацьовувань виключно на «чистому» фоні.

*План польоту та покриття ділянки.* Траєкторії будувалися у вигляді «газонокосилки» з паралельними смугами і заданим перекриттям між ними. Така схема забезпечує рівномірне покриття площі  $N \times N$  метрів (у реальному експерименті – прямокутні ділянки з близькими за площею параметрами), дає стало повторний огляд контурів і дозволяє легко проєктувати детектування на карту. Швидкість руху трималася в діапазоні безпечних значень, достатніх для отримання різких кадрів із коротким розмиттям руху, але й не надто низьких, щоб не зменшувати добову продуктивність огляду. Розвороти виконувалися з мінімальними нахилами, щоб зменшити артефакти перспективи; у разі потреби між смугами робилися другі проходи під невеликим кутом для підвищення ймовірності виявлення цілей, що частково оклюзуються рослинністю.

*Збір даних і синхронізація каналів.* Відео 1920×1080 (MPEG-4) записувалося на Raspberry Pi, під'єднаний до камери. Паралельно аудіосигнал металошукача оцифровувався АЦП і надходив у модуль збору/обробки. Обидва канали маркувалися часовими мітками з прив'язкою до спільного часу. Така схема дає однозначну відповідність «кадр → час → позиція БПЛА». Після кожного вильоту відео й журнали передавалися на наземну станцію через Wi-Fi/SSH, де запускалися препроцесинг, інференс YOLOv8n і пост-hoc геоприв'язка детекцій до логів Pixhawk 6C.

*Поведінка металодетектора і калібрування.* Металошукач перед експериментами проходив калібрування на ділянці без цілей, під час якого визначалися базові пороги спрацювання з урахуванням локального фону (мікромагнітні неоднорідності, випадкові металеві включення ґрунту, дрібний техно–сміттєвий фон). Усі спрацювання переводилися в подієвий лог із часовою міткою й нормалізованою інтенсивністю. Очевидно, що нульовий сигнал не гарантує відсутності небезпеки (пластикові корпуси, мала металовмісність), тоді як позитивний сигнал є сильним індикатором підповерхневої аномалії – це відразу визначає алгоритмічну логіку фузії в наступних підрозділах.

*Геоприв'язка й просторово-часова узгодженість.* Детекції з візуального каналу формувалися покадрово, після чого за часовими мітками проєктувалися на інтерпольовану траєкторію з логів Pixhawk 6C і отримували координати у WGS-84 (для візуалізації у ГІС) [64] або локальну ENU-рамку (для обчислень). Події металодетектора узгоджувалися з візуальними детекціями за просторово-часовим вікном: подія вважалася узгодженою, якщо різниця часових міток не перевищує  $\Delta t_{tol}$ , а евклідова відстань між центрами подій у площині землі –  $r_{tol}$  (допуски підбиралися виходячи з висоти та геометрії FOV). Узгоджені пари (або трійки – якщо додано хмарну верифікацію Genimi 2.5 Flash) ставали кандидатами «підтверджених подій»; неузгоджені – потрапляли у відповідні підкласи («лише візуальні», «лише метал»).

*Оцінка результатів.* Для об'єктивного порівняння детекцій із «правдою» на полігоні заздалегідь формувався план розкладки: позиції макетів фіксувалися в локальній системі відліку від опорних маркерів (базові кілки/мітки на кутах ділянки) з вимірюванням рулетками/віхами; координати потім трансформувалися в ту ж систему, у якій реєструвалися траєкторії БПЛА. Це дозволяло, не втручаючись у природність сцени, мати достатньо точний еталон для метричних зіставлень. Подія вважалась істинно позитивною, якщо (I) належала до шуканого класу (PFM-1 на поверхні/частково прикрита або «підповерхнева аномалія» для металоканалу), (II) потрапляла у просторовий допуск від фактичного розташування цілі, (III) у разі «підповерхневих» сценаріїв мала позитивне металоспрацювання. Для нульових проходів усі спрацювання розглядалися як хибні, що дозволяло оцінити фонову частоту FP.

*Ролі наземної станції й хмари.* Наземна станція виконувала основний цикл обробки: стабілізація/resize/нормалізація → інференс YOLOv8n → NMS → логування та анотація відео → геоприв'язка. Хмарний рівень використовувався вибірково як семантичний верифікатор для складних випадків (Genimi 2.5 Flash): із кадрів вирізалися ROI, які викликали сумніви або мали низький запас упевненості; до них формувалися інструкційні запити й отримувалися скалярні оцінки довіри для подальшої фузії. Для підрозділу 5.1 важливо зафіксувати, що саме така послідовність обчислень застосовувалася на всіх серіях і забезпечувала однакову відтворюваність збору/обробки.

*Сценарні варіанти місії.* Упродовж сесій апробовано три базові сценарії, що відрізнялися поєднанням фонів і станів цілей. У першому сценарії акцент зроблено на «чистих» поверхневих макетах ПФМ-1 з різною орієнтацією на відкритому тлі – це давало орієнтир верхньої межі якості візуального каналу та дозволяло калібрувати порогові значення скорингу/-NMS. У другому сценарії цілі частково екранувалися травою, мали тіньові артефакти або частково присипалися ґрунтом; саме тут відстежувалася

поведінка FP і виграш від семантичної верифікації. Третій сценарій моделював «підповерхневі» загрози: металевий сурогат без видимих оптичних ознак у кадрі – орієнтований на перевірку чутливості й стабільності металоканалу та логіки маркування таких подій як «потенційні підземні цілі». Кожен сценарій відпрацьовувався у кількох заїздах з повторенням траєкторій і варіюванням мікроконтексту, що дозволяло відокремити випадкові фактори (мікровітер, локальні вібрації, «шум» рослинності) від системних властивостей комплексу.

*Контроль якості й протокол ведення журналів.* Для кожного вильоту зберігалися: не оброблений відеофайл, анотоване відео з рамками та підписами, лог детекцій (час, bbox у пікселях, conf-score, клас), лог металоподій (час, інтенсивність), фрагмент телеметрії (позиція/висота/курс). Окремо вівся експериментальний журнал із метео-замітками, описом сцени, серій/проходів, а також із мітками, чи були присутні макети і в якому стані. Такий набір артефактів забезпечував повну трасованість від кожної фінальної точки на карті до первинних даних, що критично для надійності висновків і можливості повторного аналізу.

*Безпека та етичні аспекти.* Усі дослідження проводились виключно з муляжами та технічними сурогатами, без використання реальних ВНП; польоти здійснювалися у віддалених від людей місцях із дотриманням стандартних вимог до використання БПЛА. Відео зберігалось та оброблялося із дотриманням принципів приватності: відсутні персональні дані, не здійснювалося цілеспрямоване знімання третіх осіб чи приватних територій.

*Програмне забезпечення керування польотом (ArduPilot / GCS).* У складі інтегрованого комплексу ключову роль відіграє наземна програма керування (рис. 4.3) польотом (Ground Control Station, GCS), сумісна з ArduPilot і контролером Pixhawk 6C. Вона забезпечує повний цикл роботи – від підготовки місії й передпольотних перевірок до онлайн-моніторингу, запису телеметрії та післяпольотного аналізу. У випробуваннях застосовувалася типова GCS-програма з класичним інтерфейсом (панель показників польоту,

інтерактивна карта, стани системи, журнал подій, доступ до параметрів і логів). Програмний інтерфейс, доданий у вигляді ілюстрації, демонструє характерні елементи: штучний горизонт/швидкості/висоти, вибір режиму (Stabilize/Loiter/Auto/RTL/Guided тощо), мапу місії з трасою, індикатори стану EKF/GPS/компасів/живлення, консоль повідомлень MAVLink [65] і вкладки параметрів та лог-аналізу.

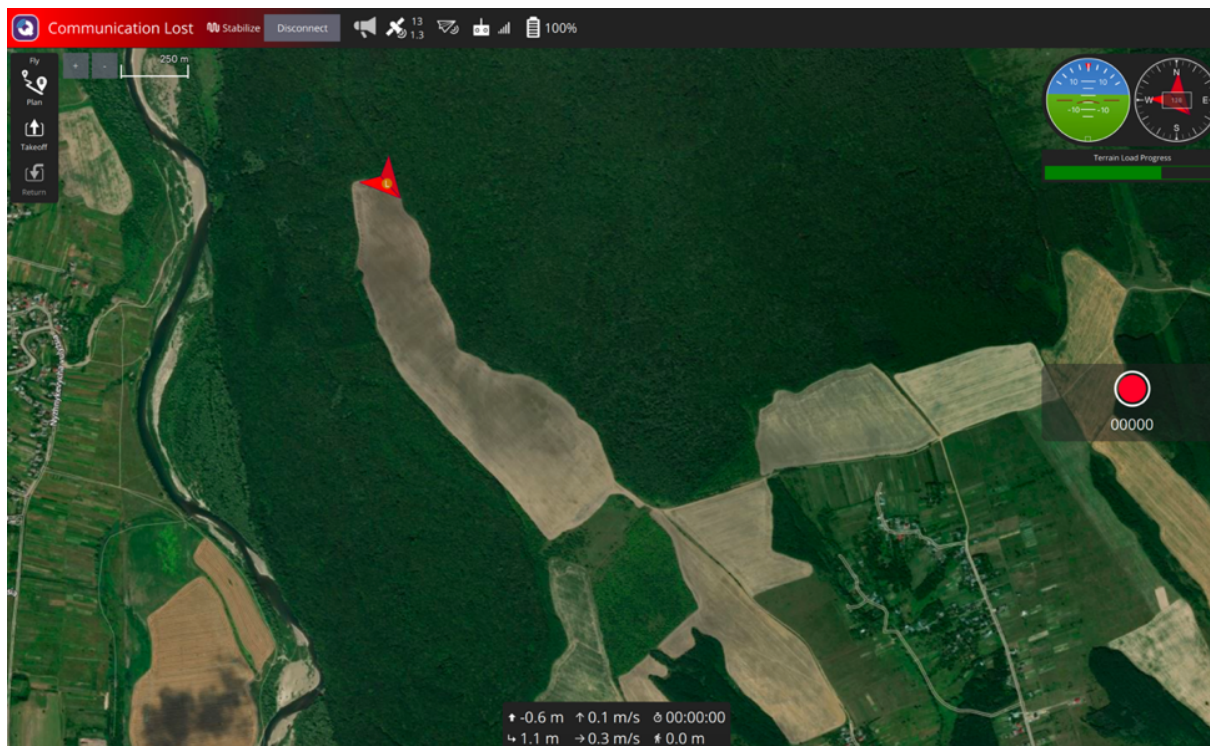


Рисунок 4.3 – Інтерфейс програми для планування польоту QGroundControl

*Передпольотна підготовка та контроль стану.* Перед кожним вильотом виконується стандартизований чек-лист у GCS. На вкладці сенсорів перевіряються калібрування акселерометрів, компасів, баровисотоміра; на сторінці радіо – відповідність крайніх положень каналів і міксів (апаратура Radiomaster TX12), [66] коректність failsafe. Блок Battery/Power показує напругу акумулятора, струм та прогноз часу польоту. GPS/EKF – кількість супутників, точність позиціювання, стан фільтра орієнтації. Для нашого профілю польоту критичною є висота над землею, тому у параметрах ArduPilot активовано використання лазерного



далекоміра (лідара) як джерела AGL-висоти. У GCS це відображається окремим показником RangeFinder (система датчиків відстані) і використовується для контролю 2-3 м під час місії. Перед зльотом GCS виконує arming-checks: справність сенсорів, пороги батареї, готовність GPS/EKF, коректність режиму failsafe (RC/GPS/батарея/геозона). Усі зауваження фіксуються в Message Console і блокують ARM до усунення.

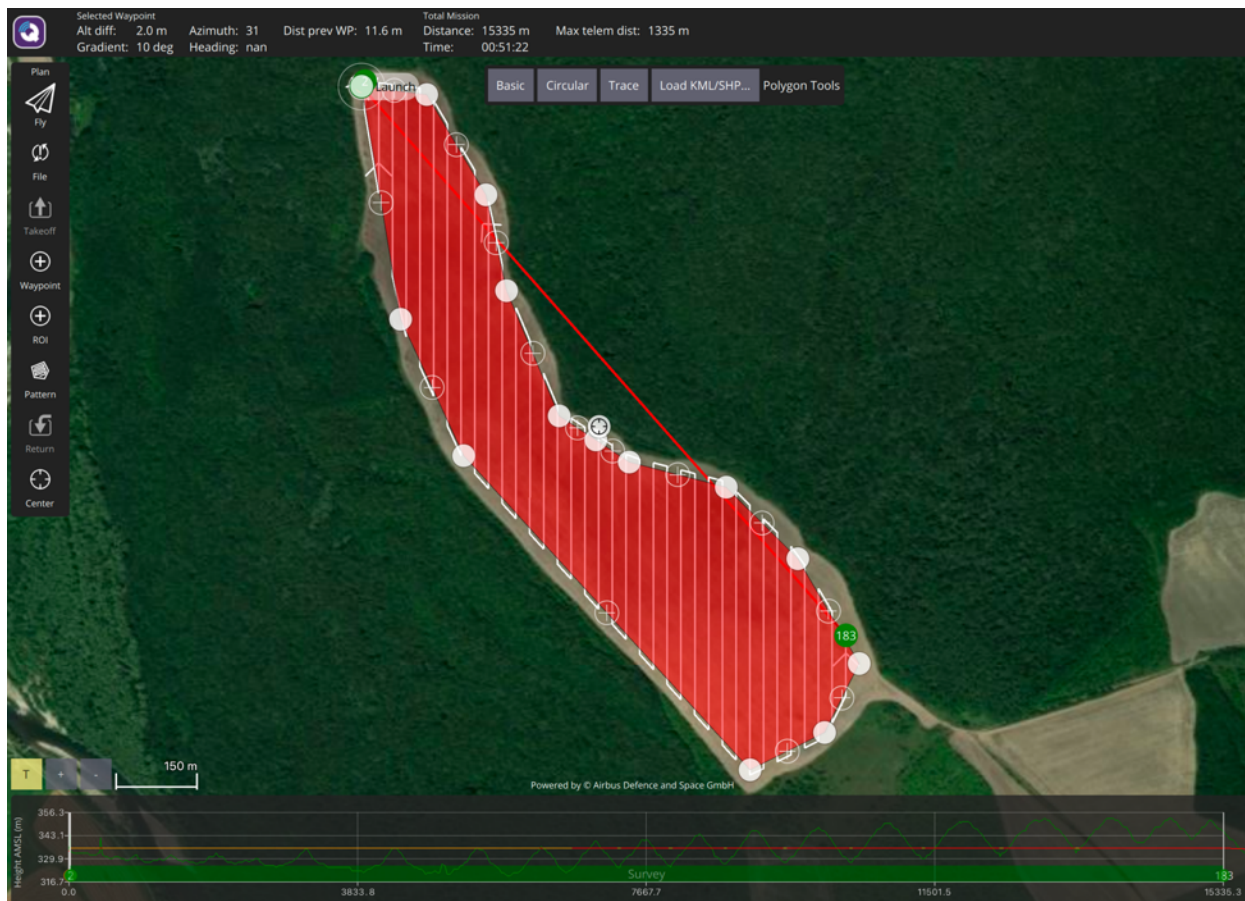


Рисунок 4.4 – Зображення запланованого польоту

*Планування місії й режими польоту.* Побудова маршрутів здійснюється безпосередньо на карті GCS. Для експериментів застосовано шаблон «газонокосилка» (lawnmower/grid) із контрольованим перекриттям смуг, сталою висотою AGL 2-3 м та обмеженням швидкості, що гарантує чіткі контури малих цілей у кадрі. На початку місії задається Home і опційно геозона (fence) – полігон, за межі якого апарат не вийде, а у разі

порушення fence спрацьовує RTL або автопосадка. Практично використовуються режими:

- **Auto** – автономне відпрацювання сітки збережених точок;
- **Loiter/PosHold** – зависання для локальної зйомки/повторного огляду;
- **Guided** – точкові наведення з GCS у реальному часі;
- **RTL** – повернення додому по завершенні місії, за тривоги або за командою.

У нашому конвеєрі відео пишеться безперервно на Raspberry Pi, тому події «спуск камери» у місії не використовувалися, однак GCS дозволяє додавати команди-маркери (наприклад, «DO\_SET\_ROI», «DO\_DIGICAM\_CONTROL») – за потреби вони можуть служити синхромітками для подальшого аналізу.

*Онлайн-моніторинг і ведення журналів.* Під час польоту оператор спостерігає за HUD, маршрутом на мапі та панеллю станів. Усі телеметричні повідомлення (позиція, швидкість, курс, висота AGL/AMSL, напруга, температура, статус EKF, попередження) надходять по MAVLink через Holybro SiK Telemetry Radio 433 МГц і логуються у два типи журналів:

**TLOG** – журнал GCS (потік MAVLink, зручний для швидкого перегляду, графіків і накладань на мапі).

**DataFlash** – бортовий журнал Pixhawk із високою роздільністю для післяпольотної діагностики (GPS, тощо).

Саме з цих логів у подальшому виконується пост-факт геоприв'язка подій комп'ютерного зору: часові мітки кадрів синхронізуються з TLOG/DataFlash і трансформуються у координати WGS-84 або локальну систему ENU. За необхідності GCS надає візуалізацію треку, висотного профілю, векторів вітру, розкладання параметрів по часу; це суттєво спрощує відбір «проблемних» ділянок і повторні обльоти.

*Параметри, тюнінг і безпека.* ArduPilot дає повний доступ до дерева параметрів. Для нашого сценарію були важливі:



**Висота над землею:** активований RangeFinder як джерело AGL, налаштовані пороги мін/макс-відстані та поведінка на втрату сигналу.

**Швидкості/прискорення:** обмеження горизонтальної швидкості та вертикальних наборів/зниження для збереження різкості кадрів.

**FENCE/RTL/LAND:** геозона, поведінка при низькій батареї, втратах RC/телеметрії, автоматичний RTL по завершенні сітки.

**Фільтри EKF/компасів:** зважування джерел висоти (баро/лідар), відсікання магнітних аномалій.

У GCS це відображено окремим розділом; зміни параметрів супроводжуються підказками й збереженням профілів, що дозволяє швидко перемикати «консервативну» і «агресивнішу» конфігурації для різних фонів місцевості. Після кожного туру параметри фіксуються у звітному файлі разом із версіями прошивки – це забезпечує відтворюваність.

З погляду безпеки, окрім геозони, використовуються двоступеневе ARM, контроль батарейних порогів, RC-failsafe, GCS-failsafe (на випадок розриву телеметрії) і швидкі апаратні команди на пульті (зупинка місії/Loiter/RTL). Всі спрацювання записуються в журнали і відображаються на панелі повідомлень.

GCS не взаємодіє безпосередньо з Raspberry Pi, але синхронізує часові процеси: початок місії, режими польоту, позиція й висота – це «еталон часу/місця» для відеопотоку. Завдяки цьому копія відео по Wi-Fi/SSH зберігає зв'язок із траєкторією кадр-за-кадром. Лінія металодетектора, підключена до Raspberry Pi (АЦП), також синхронізується за часом із GCS-логами, тому в подальшому ми виконуємо просторово-часове злиття даних: спрацювання металоканалу «прив'язуються» до конкретних ділянок траси, а візуальні детекції YOLO – до тих самих тайм-сегментів. Саме дані, отримані через GCS, дозволяють побудувати об'єднані карти подій і статистики покриття (площа/хвилина, повторні огляди, ділянки з погіршеною якістю сигналів).

*Післяпольотний аналіз і відтворюваність.* Після завершення вильоту GCS використовується для завантаження та перегляду TLOG/DataFlash, порівняння фактичної траси з планом, оцінки стабільності висоти 2–3 м і

швидкісного профілю, виявлення відхилень або зон із підвищеними вібраціями. На цій стадії позначаються часові інтервали для поглибленого CV-аналізу (наприклад, складні тіньові умови), і відповідні сегменти відео подаються у наш Video Landmine Detection & Annotation System. Чітке розмежування ролей – «GCS як еталон телеметрії/місії», «Raspberry Pi як джерело відео й сигналів» – зробило процес повністю трасованим: будь-яку детекцію можна відновити в часі, у просторі й у вихідному кадрі.

Таким чином, GCS-програма у зв'язці з ArduPilot виконує одразу кілька критичних функцій: забезпечує безпечну автоматизацію польоту на малих висотах, дає прозорий контроль над параметрами платформи, створює надійний телеметричний слід для геоприв'язки, та слугує інструментом оперативної діагностики. Саме завдяки такому програмному середовищу ми змогли стандартизувати місії «15 хв / 2–3 м AGL», отримати відтворювані набори даних для CV-модуля, інтегрувати металошукач у єдиний часовий простір і надалі коректно оцінювати мультисенсорну фузію.

## **4.2 Тестування модуля комп'ютерного зору**

Цей підрозділ систематизує результати самостійної роботи візуального модуля без залучення металодетектора. Мета – кількісно оцінити здатність комп'ютерного зору виявляти малорозмірні протипіхотні міни на польових сценах, визначити робочі діапазони порогів і затримок, проаналізувати типові помилки й окреслити межі застосовності чисто оптичного підходу. Джерелом даних були місійні відеозаписи 1920×1080 (MPEG-4), отримані під час 15-хвилинних вильотів на висоті 2–3 м із стабілізацією висоти лідаром. Відео записувалося на Raspberry Pi, після польоту передавалося на польовий ПК (Wi-Fi/SSH) і оброблялося конвеєром: стабілізація/нормалізація → інференс YOLOv8n → NMS → логування детекцій та анотація кадрів. На цьому етапі геоприв'язка не впливала на навчання чи оцінювання і використовувалась лише для навігації по матеріалу. Середовище: Ultralytics

8.3.178, Python 3.9.6, torch-2.8.0 (MPS, Apple M3 Pro) (рис. 4.5). Підсумкова модель – YOLOv8n (fused), 72 шари, 3 005 843 параметри, 8.1 GFLOPs.

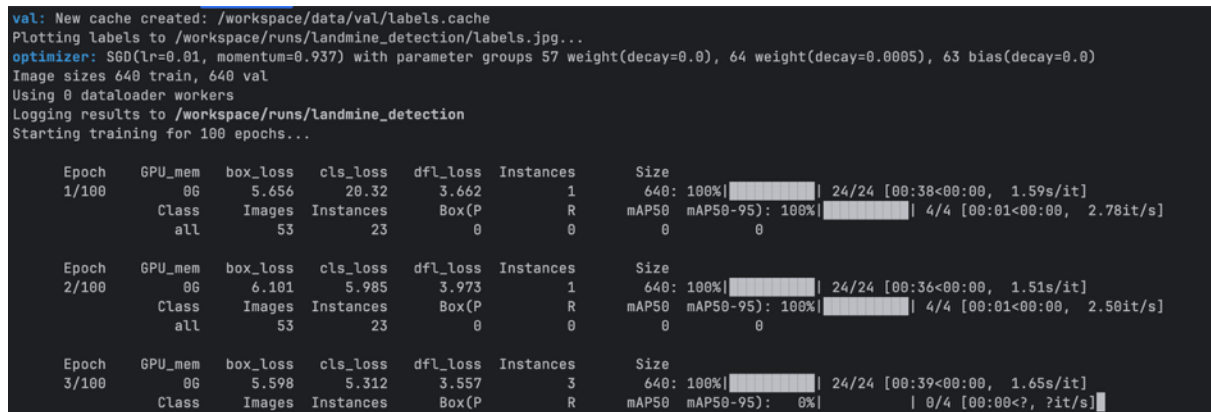


Рисунок 4.5 – Зображення процесу навчання моделі.

Валідаційний набір складався з 53 зображень і 23 інстансів цілі класу PFM-1, що відображають три типові сцени: відкритий ґрунт/трава без оклюзій, часткове прикриття рослинністю, листям або локальними тінями та фрагменти з неоднорідним фоном (фактурний ґрунт, «кущисті» структури трави). Для об'єктивності оцінювання використовували стандартні метрики Precision, Recall, mAP@0.5 і mAP@0.5:0.95, а для продуктивності – покадрові часи препроцесингу/інференсу/постпроцесингу та ефективну пропускну здатність (FPS). Усі спрацювання журналювалися у машинно-читаному вигляді (frame\_id, timestamp, conf, bbox), що забезпечувало відтворюваність аналізу (рис. 4.5).

Зведені показники якості для YOLOv8n на валідації: Precision = 0.957, Recall = 1.000, mAP@0.5 = 0.981, mAP@0.5:0.95 = 0.688. Інтерпретація цих чисел є принциповою для розуміння поведінки моделі. Високий Recall свідчить, що в контрольованих польових умовах детектор фактично не пропускає цілі класу PFM-1, тоді як Precision демонструє низьку схильність до хибних тривог за обраного порога. Зниження до mAP@0.5:0.95 = 0.688 є очікуваним: жорсткіші пороги IoU підсвічують чутливість до зміни масштабу/ракурсу та часткових оклюзій, неминучих при низьковисотній

зйомці дрібних об'єктів на неоднорідному фоні. З огляду на прикладну задачу гуманітарного розмінування це прийнятний компроміс: пріоритет віддається повноті, а локалізаційна похибка рівня «десятки сантиметрів» у багатьох сценаріях компенсується іншими рівнями конвеєра (геоприв'язкою та/або подальшою перевіркою).

Продуктивність конвеєра зафіксована як 0.5 мс на препроцесинг, 21.2 мс на інференс і 6.0 мс на постпроцесинг, сумарно  $\approx 27.7$  мс/кадр ( $\approx 36$  FPS) на польовому ноутбучі з M3 Pro. Найбільший внесок у затримку припадає на інференс ( $\sim 76.5\%$ ), отже потенціал оптимізації зосереджений саме тут (зменшення робочої роздільності при збереженні масштабу цілі; квантування; експорт у TensorRT/CoreML; використання апаратних прискорювачів; тонка настройка порогів NMS для зменшення кількості кандидатів). Водночас уже наявний FPS достатній для післяпольотного аналізу, близького до реального часу: матеріал 5-хвилинного прольоту опрацьовується із затримкою в межах десятків секунд навіть із включеною анотацією відео.

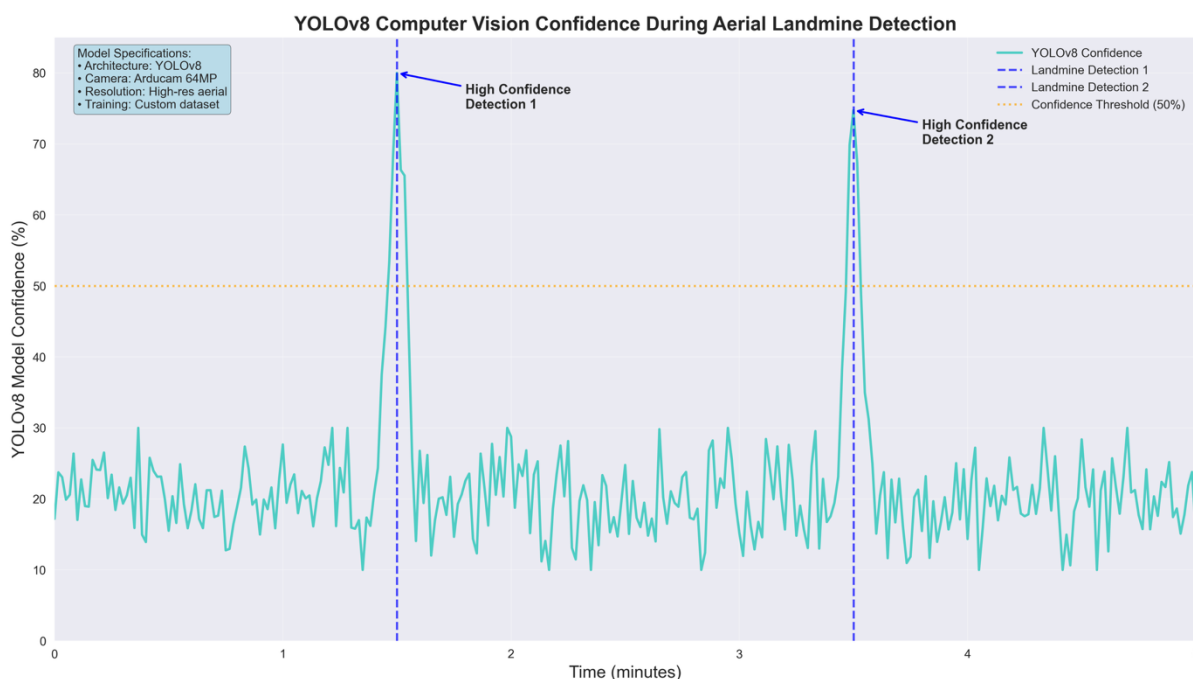


Рисунок 4.6 – Confidence-трасування YOLOv8n

Для демонстрації часової поведінки детектора розглянемо графік на рис. 4.6. На ділянках близько 1.5 та 3.5 хвилини спостерігаються виразні піки

довіри  $\approx 80.0\%$  та  $\approx 74.7\%$ , що відповідає стабільним детекціям поверхневих цілей: на анованих кадрах бачимо чіткі рамки з правильною геометрією, відсіченням «зайвих» контурів і впевненими conf-балами. На позначці  $\sim 0.8$  хвилини реакції немає – характерний приклад фізичного обмеження оптичного каналу: за відсутності видимих ознак об'єкта (повне приховування у ґрунті, сильна оклюзія) чисто візуальна система закономірно мовчить. Важливо підкреслити, що таку «німу» зону не слід інтерпретувати як провал навчання; це демонстрація меж застосовності комп'ютерного зору у виявленні підповерхневих загроз.



Рисунок 4.7 – Кадр частини відео роботи по розпізнанню мін

На рисунку 4.7. зображено приклад поверхневого детектування ПФМ-1 алгоритмом YOLOv8n у польових умовах [51]. На кадрі  $1920 \times 1080$  з показаним часовим штампом  $t=40.6s$  модель повернула одну рамку класу *LANDMINE* із довірою  $74.8\%$ , що перевищує робочий поріг ( $\approx 0.5$ ) та пройшло відсікання NMS. Об'єкт локалізовано на тлі рихлого ґрунту з дрібною

рослинністю; силует має характерні контури для ПФМ-1, попри текстурований фон і локальні тіні. Даний приклад ілюструє типову поведінку детектора на поверхневих сцерах: стабільна локалізація невеликого об'єкта та коректне придушення альтернативних кандидатів у кадрі. У сусідніх кадрах спостерігається послідовність із близькими значеннями довіри, що свідчить про просторово-часову стійкість відповіді; саме такі епізоди формують піки на графіку 5.5 (Confidence-трасування YOLOv8n). З практичного погляду приклад демонструє, що за стандартної висоти 2–3 м і неоднорідного фону детектор забезпечує достатню конфіденцію для автоматичного занесення події до журналу та подальшої геоприв'язки без участі металошукача.

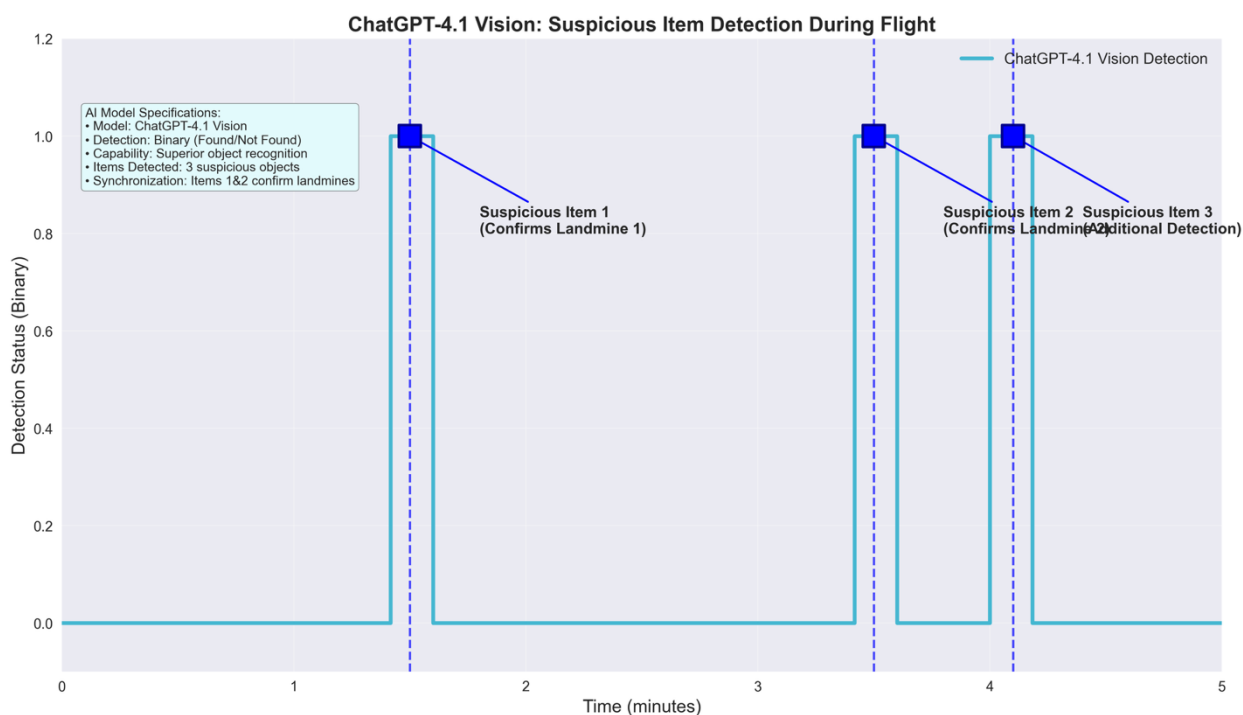


Рисунок 4.8 – Графік роботи ChatGPT 4.1 моделі при виявленні вибухо-небезпечних об'єктів

Для розширення аналізу поведінки семантичних верифікаторів у CV-лише режимі використано великі мультимодальні моделі зорово-мовного типу. Графік на Рис. 4.8 показує бінарні спрацювання ChatGPT-4.1 Vision на тій самій часовій осі: модель підтверджує два помітні епізоди (~1.5 і ~3.5 хв), які збігаються з піками довіри YOLOv8n, і додатково фіксує третій підозрілий



об'єкт близько 4.1 хв. Останній випадок цікавий тим, що не обов'язково супроводжується високою впевненістю детектора: він виникає на сцені з неоднозначним фоном, де NMS могла відсіяти «слабку» рамку або суцільна впевненість виявилася нижчою від порога. З погляду експлуатації це цінний тригер для оператора та джерело збагачення датасета «важкими прикладами»; у нашому протоколі такі моменти маркуються для подальшої ручної верифікації та, за потреби, для донавчання. Оскільки LLM-моделі не повертають bbox, їхні відповіді не включалися до mAP-метрики, але використовувалися в аналітичному сенсі: як «сигнали сумніву», що пріоритезують перегляд і розмітку. Експериментальним шляхом [44] було прораховано точність даної моделі що дорівнює 64%,  $P_C = 0.64$ .

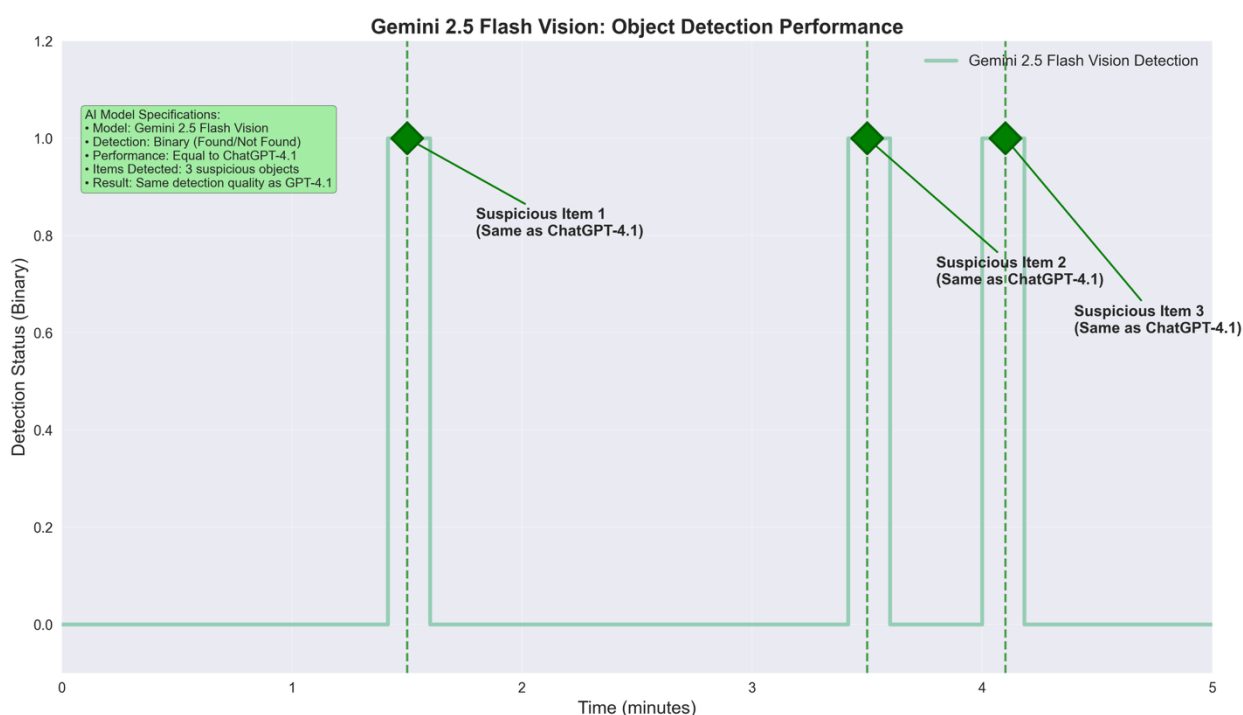


Рисунок 4.9 – Графік роботи Gemini 2.5 Flash на виявлення вибухо-небезпечних об'єктів

Графік на Рис. 4.9 відображає бінарні спрацювання Gemini 2.5 Flash Vision. Модель незалежно від ChatGPT-4.1 позначає підозрілий епізод на ~1.5, ~3.5 та ~4.1 хвилині, що є важливим проявом міжмодельної згоди. Для CV-модуля це виконує роль «другого експертного голосу» на складних

поверхневих сценах: коли дві різні VLM узгоджено вказують на можливу небезпеку, відповідні кадри отримують вищий пріоритет у ручному аудиті, а їхні фрагменти додаються до навчального пулу як негативи/псевдопозитиви для підвищення специфічності YOLO до місцевих фонів. При цьому обидві VLM, як і очікується, не реагують на ділянку  $\sim 0.8$  хвилини – ще раз підкреслимо фундаментальне обмеження всіх суто візуальних методів щодо підповерхневих об'єктів. Експериментально було прораховано [44] точність виявлення що являє  $68\% P_G = 0.68$ .

Таксономія помилок у CV-режимі сформована за підсумками багаторазових переглядів і містить кілька стабільних мотивів. Найтипівіші FP спричинені: дрібним камінням із «жорсткими» тіньовими гранями; блискучими вкрапленнями (фольга, фрагменти пластика); сухим листям і «зламаними» контурами рослин, що імітують силует; глянцевиими плямами вологи; «кущистими» структурами трави, які створюють замкнені контури, подібні до обрисів корпусу. FN найчастіше виникають у двох випадках: (i) короткочасний мікротмаз/легка дефокусування на розворотах, коли контур «розчиняється» у фоні, і (ii) часткова оклюзія рослинністю або тіннями, що «розриває» топологію силуету. Обидва типи помилок піддаються зменшенню: для FP – через цільове поповнення негативного класу і агресивніші аугментації «фонів-відволікань», а для FN – через стандартизацію профілів польоту (довші прямолінійні ділянки, м'якші розвороти), легку міжкадрову стабілізацію і контроль експозиції.

Калібрувальні міркування щодо conf-балів і порогів NMS/score є критичними для прикладної експлуатації. Хоча у звіті наведено зведені метрики для «базових» налаштувань, практична робота потребує вибору операційної точки: якщо пріоритет – мінімум пропусків, доречно знизити поріг score, прийнявши невелике збільшення FP (які потім «дешево» відсіяти переглядом); якщо пріоритет – низька хвилева тривожність оператора, поріг слід підняти та/або погратися з IoU у NMS для агресивнішого відсікання дублювань у траві. У CV-лише режимі корисно



застосовувати дворівневу стратегію: м'який поріг для первинного виявлення і суворіший – для авто-експорту «впевнених» подій у карту, залишаючи «сіру зону» для ручного аудиту або семантичної перевірки LMM. З погляду стійкості до варіацій зйомки, центральною змінною є масштаб цілі у кадрі, який прямо залежить від висоти. Робоче «вікно» 2–3 м забезпечує достатній піксельний розмір силуету і є обґрунтованим компромісом між безпекою польоту та розрізнявальною здатністю. Перехід на більші висоти веде до неминучого падіння  $mAP@0.5:0.95$ , яке частково компенсується перенавчанням зі збагаченням дрібномасштабними прикладами або зміною робочої роздільності інференсу. Освітлення також істотно впливає на FP-патерни: контрове світло та глибокі тіні збільшують кількість «жорстких» контурів. Легкий локальний контраст-енхансмент (напр., CLAHE) у препроцесингу покращує видимість дрібних деталей без надмірної появи «зерна».

Роль великих мультимодальних моделей у CV-лише сценарії – не заміщення детектора, а семантична інтерпретація і пріоритизація перегляду. Спостереження з «графіків 4 і 6» демонструють корисну властивість: LMM схильні «бачити» підозрілі структури на складному фоні раніше, ніж детектор накопичить достатню conf-масу для стабільної рамки. У публічних задачах це часто призводить до зменшення FP (LMM заперечує «міно-подібність» каменя/листка). У нашому експерименті, де мова про суто CV-оцінювання, ці моделі виконували роль post-hoc верифікаторів і джерела кейсів для поповнення датасета. Важливо також, що згодженість між LMM (випадок ~4.1 хв) надає сильний аргумент для «людського перегляду» навіть за відсутності впевнених bbox-відповідей, що добре вписується у доктрину людина-в-петлі для ризикових застосувань.

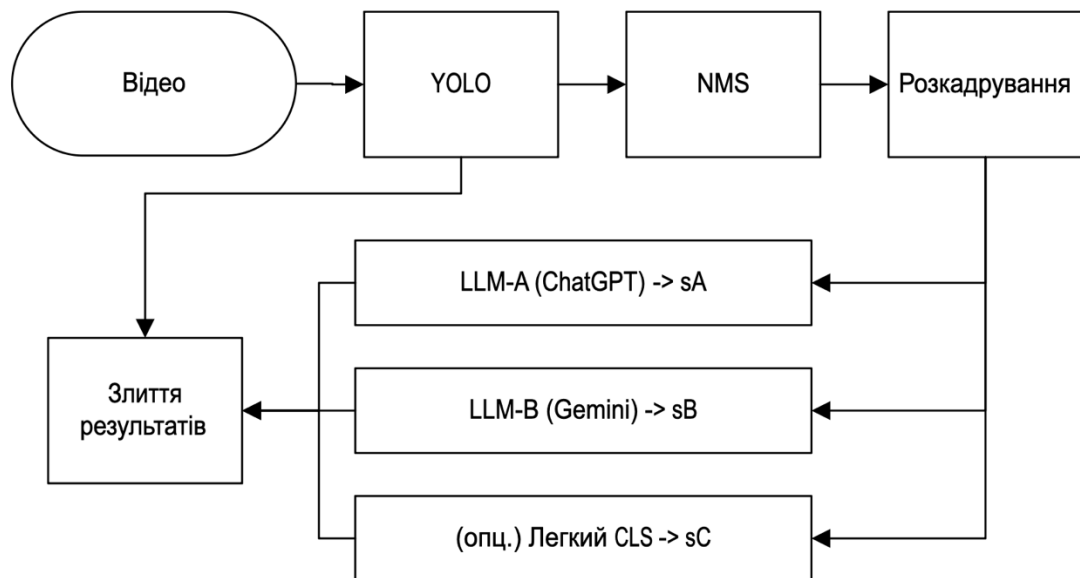


Рисунок 4.10 – Обробка лише візуальних моделей комп’ютерного зору

- **Відео** – вхідний потік  $1920 \times 1080$  (MP4) подається покадрово.
- **YOLO** – базовий детектор (YOLOv8n/m) повертає для кожного кадру множину  $bbox$  із  $conf_{YOLO} \in [0,1]$  та класом *LANDMINE*.
- **NMS** – Non-Maximum Suppression (параметр  $IoU = \theta_{nms}$ ) формує неперекривні кандидати ROI.
- **Кандидати ROI** – вирізані патчі кадру навколо  $bbox$  (із падінгом  $p$ ), відсортовані за  $conf_{YOLO}$ .
- **LLM-A (ChatGPT-4.1 Vision), LLM-B (Genimi 2.5 Flash Vision)** – семантичні верифікатори; на вхід: ROI + короткий промпт; на вихід:  $s_A, s_B \in [0,1]$  – «ймовірність/довіра, що ROI є ППМ/підозрим об’єктом».
- **(опц.) Легкий CLS** – швидкий локальний класифікатор/до-детектор (наприклад, мобільна CNN або логістична регресія на візуальних ознаках ROI); вихід  $s_C \in [0,1]$ .
- **Злиття даних**  $s = f(conf_{YOLO}, s_A, s_B, s_C)$  – правило об’єднання оцінок (див. нижче).
- **accept / reject** – фінальне рішення для ROI: прийнята подія збережеться в журналі/анотованому відео.
- **Пороги:**

- $\tau_Y$  – поріг для первинного відбору ROI за  $\text{conf}_{YOLO}$ ,
- $\tau^*$  – поріг для фінального рішення після фузії,
- $K$  – максимум ROI/кадр, що підуть у LLM (контроль латентності/вартості).

Потік відео надходить у YOLO, який повертає набір рамок із  $\text{conf}_{YOLO}$ . Після NMS утворюються кандидати ROI (рис. 4.10). Далі діє двоступенева логіка: (1) швидке правило для «впевнених» детекцій (якщо  $\text{conf}_{YOLO} \geq \tau_{Y,high}$  – одразу *accept* без LLM), (2) верифікація «сірої зони»: ROI із  $\tau_{Y,low} \leq \text{conf}_{YOLO} < \tau_{Y,high}$  надсилаються в LLM-A/B (та, за потреби, в легкий CLS). Відповіді семантичних моделей перетворюються на скалярні  $s_A$ ,  $s_B$  (і  $s_C$ ).

Фінальне рішення формується фузією оцінок. Типові варіанти  $f(\cdot)$ :

- **Зважена логістика:**

$$s^{\setminus*} = \sigma(w_0 + w_Y g(\text{conf}_{YOLO}) + w_A s_A + w_B s_B + w_C s_C),$$

де  $g(\cdot)$  – калібрована довіра (напр., temperature scaling),  $\sigma$  – логістична функція. Приймаємо ROI, якщо  $s^{\setminus*} \geq \tau^*$

- **Правила OR/AND (спрощено):** *OR-gate*: прийняти, якщо  $\text{conf}_{YOLO} \geq \tau_Y$  або  $s_A \geq \tau_A$  або  $s_B \geq \tau_B$ . *YOLO→LLM-gate*: прийняти лише тоді, коли  $\text{conf}_{YOLO} \geq \tau_{Y,low}$  і принаймні одна LLM повернула  $s \geq \tau_{LLM}$ .

Щоб утримати затримку, застосовують отримання топ- $K$  ROI у LLM на кадр і асинхронну обробку: анотація кадру з «впевненими» YOLO-подіями не чекає LLM; коли приходять відповіді LLM по «сірих» ROI, журнал оновлюється. Якщо LLM недоступна/таймаут – схема відкату: рішення приймає лише YOLO (або YOLO+CLS), а ROI маркується для офлайн-перевірки.

Інтерпретація: запропонована асинхронна конвеєрна архітектура зменшує хибні спрацювання на складних фонах завдяки семантичній верифікації (LLM-A/B), зберігаючи високий recall базового YOLO. Цей комплекс комп'ютерного зору не замінює металоканал і застосовується тільки для поверхневих/видимих об'єктів, тобто у системному

рівні фузії (розд. 3.2–3.3) він працює як «візуальна підсистема» перед об'єднанням із металодетектором.

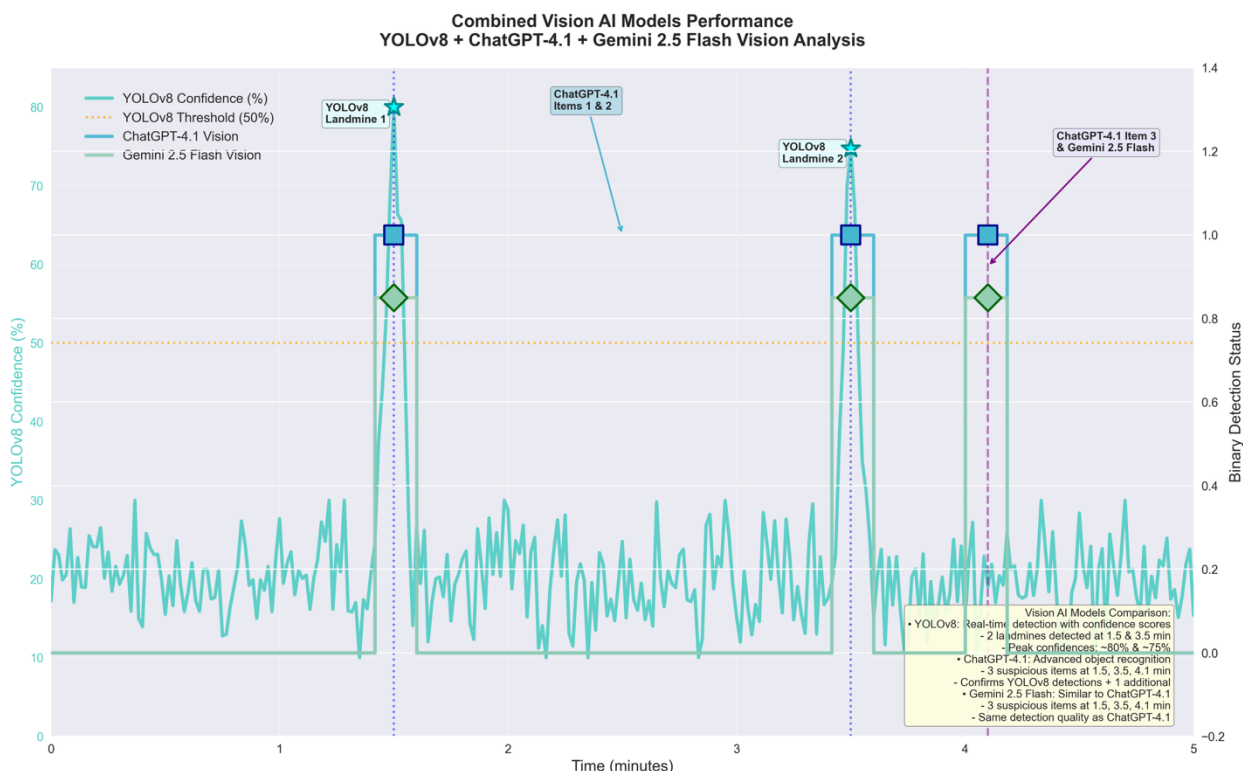


Рисунок 4.11 – Комбінований графік роботи моделей

Підсумовуючи, чисто візуальний модуль (рис. 4.11) у поточній конфігурації демонструє такі властивості. По-перше, високий recall на поверхневих сценах з PFM-1, що підтверджено  $mAP@0.5 = 0.981$  і  $Recall = 1.000$ . По-друге, керована точність ( $Precision = 0.957$ ) за рахунок цільового добору негативів і контрольованих порогів; у «важких» сценах precision додатково підкріплюється семантичними верифікаторами. По-третє, операційна швидкодія забезпечує післяпольотне опрацювання практично «на місці», не вимагаючи потужних серверів; вузьким місцем залишається інференс, де є прямі шляхи оптимізації. І нарешті, модуль чітко дотримується фізичної межі застосовності: у сценаріях без видимих ознак (повне приховування у ґрунті) він принципово не дає сигналів, що узгоджується з очікуваннями і задає мотивацію для мультисенсорної фузії в наступних підрозділах.

### 4.3 Тестування модуля металошукача

Метою цього підрозділу є кількісна оцінка можливостей металодетекторного каналу як самостійного сенсора, без допомоги комп'ютерного зору. Така оцінка потрібна для коректного порівняння з комбінованою системою (розд. 3.3) і для побудови правил фузії (розд. 3.1). У випробуваннях застосовано рамковий металошукач з аудіовиходом, під'єднаним через АЦП до Raspberry Pi. Сигнал дискретизувався, проходив смугову фільтрацію, вирівнювання рівня та огинаючу, а події детектувалися за адаптивним порогом

$$T = \mu_{\text{noise}} + k \sigma_{\text{noise}} \quad (4.1)$$

Де  $\mu_{\text{noise}}$ ,  $\sigma_{\text{noise}}$  — оцінки середнього та СКВ фонових коливань на «чистій» ділянці перед стартом місії, а  $k$  добирали з урахуванням компромісу «чутливість ↔ хибні спрацювання». Кожне перевищення порогу агрегувалося у подію з атрибутами {timestamp, тривалість, амплітуда} і маркувалося у журналі. Стабільність висоти 2-3 м AGL забезпечувалася лідаром; швидкість руху підбиралася так, щоб забезпечити достатній *dwell time* над ціллю, мінімізуючи змазування чутливого сліду рамки.

Експериментальна перевірка роботи модуля металошукача передбачала комплексне тестування апаратно-програмної частини системи для підтвердження надійності збору, обробки та синхронізації різнорідних даних. Основу програмного забезпечення реалізовано мовою Python, що дозволило забезпечити гнучкість у керуванні потоками даних та інтеграції зовнішніх бібліотек. Ключовим завданням на цьому етапі було налагодження взаємодії між бортовим комп'ютером Raspberry Pi, польотним контролером Pixhawk та платою аналого-цифрового перетворення, а також перевірка коректності перетворення аналогового сигналу від індукційного датчика у цифрові значення, придатні для подальшого аналізу та картографування.

Програмна архітектура використовує бібліотеку «pymavlink», яка є реалізацією протоколу MAVLink для Python, що дозволяє отримувати високоточні GPS-координати [49] безпосередньо з контролера Pixhawk у режимі реального часу. Це є критично важливим для точної географічної прив'язки виявлених металевих об'єктів. Паралельно з цим, вихідний сигнал металошукача, який спочатку надходить у аналоговій формі (зміни рівня напруги), проходить через плату високоточного аналого-цифрового перетворення. Обчислювальний модуль Raspberry Pi обробляє ці оцифровані сигнали, виконуючи розрахунок рівнів шуму та конвертуючи їх у значення децибелів за логарифмічною шкалою. Такий підхід дозволяє перетворити сирі електричні коливання у чіткий показник інтенсивності події детектування, відображаючи реальну реакцію сенсора на металеві об'єкти в зоні моніторингу, при цьому все середовище виконання ізольоване у Docker-контейнері для забезпечення стабільності роботи незалежно від зовнішніх факторів. На рисунку 4.12 продемонстровано процес консолідації даних та вигляд сформованого масиву інформації в режимі реального часу.

```

X serhil@raspberrypi: ~ (ssh)
Logged Data - Timestamp: 2024-04-21 20:35:12.577, GPS: Lat: 48.9020054, Lon: 24.682523, Alt: 248.91m, Decibel: 2.66 dB
Logged Data - Timestamp: 2024-04-21 20:35:12.827, GPS: Lat: 48.9020053, Lon: 24.6825296, Alt: 248.94m, Decibel: 2.73 dB
Logged Data - Timestamp: 2024-04-21 20:35:13.077, GPS: Lat: 48.9020051, Lon: 24.6825291, Alt: 248.96m, Decibel: 2.68 dB
Logged Data - Timestamp: 2024-04-21 20:35:13.327, GPS: Lat: 48.9020049, Lon: 24.6825285, Alt: 248.98m, Decibel: 3.28 dB
Logged Data - Timestamp: 2024-04-21 20:35:13.577, GPS: Lat: 48.9020048, Lon: 24.6825277, Alt: 249.0m, Decibel: 2.08 dB
Logged Data - Timestamp: 2024-04-21 20:35:13.828, GPS: Lat: 48.9020046, Lon: 24.6825274, Alt: 249.01m, Decibel: 2.92 dB
Logged Data - Timestamp: 2024-04-21 20:35:14.078, GPS: Lat: 48.9020046, Lon: 24.6825269, Alt: 249.02m, Decibel: 4.02 dB
Logged Data - Timestamp: 2024-04-21 20:35:14.328, GPS: Lat: 48.9020046, Lon: 24.6825265, Alt: 249.03m, Decibel: 3.94 dB
Logged Data - Timestamp: 2024-04-21 20:35:14.577, GPS: Lat: 48.9020046, Lon: 24.6825263, Alt: 249.03m, Decibel: 42.36 dB
Logged Data - Timestamp: 2024-04-21 20:35:14.828, GPS: Lat: 48.9020047, Lon: 24.6825259, Alt: 249.02m, Decibel: 17.58 dB
Logged Data - Timestamp: 2024-04-21 20:35:15.078, GPS: Lat: 48.9020048, Lon: 24.6825254, Alt: 249.02m, Decibel: 8.14 dB
Logged Data - Timestamp: 2024-04-21 20:35:15.328, GPS: Lat: 48.9020048, Lon: 24.6825249, Alt: 249.0m, Decibel: 3.35 dB
Logged Data - Timestamp: 2024-04-21 20:35:15.577, GPS: Lat: 48.9020046, Lon: 24.6825246, Alt: 248.98m, Decibel: 3.70 dB
Logged Data - Timestamp: 2024-04-21 20:35:15.828, GPS: Lat: 48.9020047, Lon: 24.6825241, Alt: 248.97m, Decibel: 2.39 dB
Logged Data - Timestamp: 2024-04-21 20:35:16.078, GPS: Lat: 48.9020048, Lon: 24.6825233, Alt: 248.96m, Decibel: 19.79 dB
Logged Data - Timestamp: 2024-04-21 20:35:16.328, GPS: Lat: 48.9020048, Lon: 24.6825227, Alt: 248.94m, Decibel: 8.62 dB
Logged Data - Timestamp: 2024-04-21 20:35:16.578, GPS: Lat: 48.9020047, Lon: 24.6825224, Alt: 248.92m, Decibel: 19.29 dB
Logged Data - Timestamp: 2024-04-21 20:35:16.828, GPS: Lat: 48.9020048, Lon: 24.6825222, Alt: 248.91m, Decibel: 11.02 dB
Logged Data - Timestamp: 2024-04-21 20:35:17.077, GPS: Lat: 48.9020049, Lon: 24.6825216, Alt: 248.91m, Decibel: 20.11 dB
Logged Data - Timestamp: 2024-04-21 20:35:17.328, GPS: Lat: 48.9020051, Lon: 24.6825211, Alt: 248.91m, Decibel: 13.05 dB
Logged Data - Timestamp: 2024-04-21 20:35:17.577, GPS: Lat: 48.9020052, Lon: 24.6825208, Alt: 248.92m, Decibel: 7.61 dB
Logged Data - Timestamp: 2024-04-21 20:35:17.827, GPS: Lat: 48.9020052, Lon: 24.6825203, Alt: 248.92m, Decibel: 34.77 dB
Logged Data - Timestamp: 2024-04-21 20:35:18.077, GPS: Lat: 48.9020052, Lon: 24.6825198, Alt: 248.92m, Decibel: 13.93 dB
Logged Data - Timestamp: 2024-04-21 20:35:18.328, GPS: Lat: 48.9020052, Lon: 24.6825194, Alt: 248.92m, Decibel: 9.15 dB
Logged Data - Timestamp: 2024-04-21 20:35:18.577, GPS: Lat: 48.9020052, Lon: 24.6825193, Alt: 248.91m, Decibel: 4.65 dB
Logged Data - Timestamp: 2024-04-21 20:35:18.827, GPS: Lat: 48.9020053, Lon: 24.6825188, Alt: 248.91m, Decibel: 4.75 dB
Logged Data - Timestamp: 2024-04-21 20:35:19.077, GPS: Lat: 48.9020055, Lon: 24.6825178, Alt: 248.92m, Decibel: 2.47 dB
X serhil@raspberrypi: ~/metal_detector (ssh)

```

Рисунок 4.12 – Компонований сигнал з металодетектора та GPS координат

Оброблена інформація, що включає як рівні сигналу в децибелах, так і відповідні їм GPS-координати, автоматично інтегрується та записується у структурований файл формату «.csv», який виступає первинним сховищем даних. Спеціалізовані скрипти Python відповідають за синхронізацію потоків даних, гарантуючи, що інформація від металошукача та GPS-модуля збігається у часі, що дозволяє розглядати кожен запис у файлі як комплексний набір даних для кожної точки вимірювання. Така структура забезпечує можливість подальшого аналізу, оскільки кожна подія виявлення чітко прив'язана до географічної локації. Для забезпечення цілісності даних система налаштована на регулярну синхронізацію цього файлу з хмарним сервером, що мінімізує ризики втрати інформації при локальних збоях та дозволяє проводити віддалений моніторинг результатів сканування. Для забезпечення довгострокового зберігання та глобальної доступності експериментальних даних реалізовано механізм автоматичної синхронізації з віддаленим сервером, розгорнутим на хмарній інфраструктурі Hetzner. В якості системи управління базами даних використовується MySQL 8 (рис. 4.13) під управлінням операційної системи Ubuntu, що гарантує високу продуктивність та надійність обробки запитів.

| id | created                 | gps_latitude | gps_longitude | metal_detection_signal |
|----|-------------------------|--------------|---------------|------------------------|
| 73 | 2024-04-23 17:31:26.270 | 48.9017319   | 24.6828179    | 3                      |
| 74 | 2024-04-23 17:31:26.520 | 48.901732    | 24.6828179    | 3                      |
| 75 | 2024-04-23 17:31:26.770 | 48.9017318   | 24.6828185    | 2                      |
| 76 | 2024-04-23 17:31:27.020 | 48.9017322   | 24.6828182    | 3                      |
| 77 | 2024-04-23 17:31:27.270 | 48.9017323   | 24.6828184    | 3                      |
| 78 | 2024-04-23 17:31:27.520 | 48.9017322   | 24.6828187    | 3                      |
| 79 | 2024-04-23 17:31:27.770 | 48.9017321   | 24.682819     | 3                      |
| 80 | 2024-04-23 17:31:28.020 | 48.9017327   | 24.6828188    | 34                     |
| 81 | 2024-04-23 17:31:28.270 | 48.9017323   | 24.6828193    | 24                     |
| 82 | 2024-04-23 17:31:28.520 | 48.9017323   | 24.6828193    | 11                     |
| 83 | 2024-04-23 17:31:28.770 | 48.901733    | 24.6828186    | 32                     |
| 84 | 2024-04-23 17:31:29.020 | 48.9017328   | 24.6828188    | 31                     |
| 85 | 2024-04-23 17:31:29.270 | 48.9017331   | 24.6828187    | 32                     |
| 86 | 2024-04-23 17:31:29.520 | 48.9017331   | 24.6828189    | 22                     |
| 87 | 2024-04-23 17:31:29.770 | 48.9017332   | 24.6828188    | 3                      |
| 88 | 2024-04-23 17:31:30.020 | 48.901734    | 24.6828182    | 3                      |
| 89 | 2024-04-23 17:31:30.270 | 48.9017343   | 24.6828181    | 3                      |
| 90 | 2024-04-23 17:31:30.520 | 48.9017344   | 24.6828183    | 3                      |
| 91 | 2024-04-23 17:31:30.770 | 48.901735    | 24.682818     | 2                      |
| 92 | 2024-04-23 17:31:31.020 | 48.9017357   | 24.682817     | 3                      |
| 93 | 2024-04-23 17:31:31.270 | 48.9017359   | 24.6828166    | 3                      |

Рисунок 4.13 – Збережені у хмару дані від металодетектора та GPS координат

Архітектура хмарного сховища дозволяє консолідувати інформацію з локальних носіїв у єдину реляційну структуру, забезпечуючи резервне копіювання в режимі реального часу та надаючи можливість віддаленого доступу до результатів сканування для їх подальшої аналітичної обробки незалежно від фізичного стану бортового обладнання.

$$\begin{bmatrix} dB\ level_1 \\ dB\ level_2 \\ \vdots \\ dB\ level_n \end{bmatrix} = \begin{bmatrix} f_1(longitude_1, latitude_1, time_1) \\ f_2(longitude_2, latitude_2, time_2) \\ \vdots \\ f_n(longitude_n, latitude_n, time_n) \end{bmatrix} \quad (4.2)$$

Для аналітичного представлення взаємозв'язку між змінними у досліджуваній системі було розроблено матричну модель (4.2), яка формалізує залежність рівня сигналу металошукача від просторових та часових параметрів. Дана матриця дозволяє структурувати отримані експериментальні дані, де кожен елемент запису представляє виявлений рівень сигналу в децибелах як функцію від відповідних значень довготи, широти та часу вимірювання. Такий математичний опис є необхідним для побудови детальних карт забруднення території, оскільки він встановлює чітку функціональну відповідність між фізичним розташуванням сенсора в конкретний момент часу та інтенсивністю зареєстрованого електромагнітного відгуку, що є основою для подальшої алгоритмічної обробки та візуалізації небезпечних зон.

*Постановка експерименту.* Випробування виконано у полі на ділянках змішаного фону (грунт/трава/тіні). Цілі: (i) підземна протитанкова міна (муляж) під тонким шаром ґрунту – для перевірки підповерхневої чутливості; (ii) невідомий металевий уламок на поверхні – для оцінки наземних металевих об'єктів середнього розміру; (iii) контрольні «нульові» проходи без цілей – для вимірювання фонові тривожності. Для просторово-часового зіставлення з траєкторією кожна подія отримувала геоприв'язку за логами Pixhawk 6S (післяпольотна інтерполяція часових міток).



*Ключові метрики.* Як узагальнену міру чутливості використовуємо ймовірність виявлення  $P_{d,m}$  (у цьому розділі позначатимемо її  $P_m$ ), а для стійкості до фону – частоту хибних спрацювань FAR (подій/хв або подій/100 м шляху). За даними полігонних серій при робочому порозі, відкаліброваному за місцевим фоном, отримано репрезентативне значення:  $P_m \approx 0.70$ .

Це означає, що близько 70% релевантних металовмісних цілей даного типорозміру та закладання (на поверхні або під тонким шаром ґрунту) викликають стійку подію на часовій шкалі польоту. Показник FAR після калібрування і дебаунсу залишався прийнятним для подальшої фузії: на «нульових» проходах спостерігалися поодинокі короткочасні сплески, зумовлені локальними металевими включеннями ґрунту або електромагнітними завадами; їх придушували підвищенням  $k$  та вимогою мінімальної тривалості події.

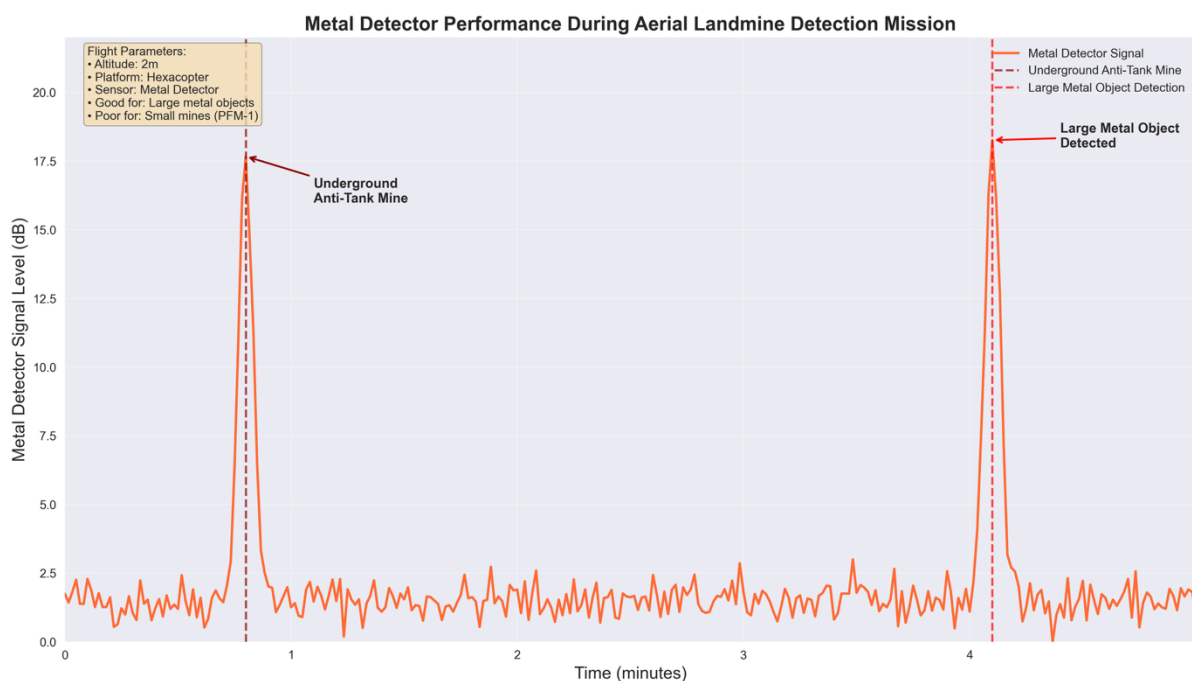


Рисунок 4.14 – Робота модулю металодетекції

*Ілюстративний приклад* (рис. 4.14). На часовій діаграмі амплітуди металосигналу під час п'ятихвилинного прольоту зафіксовано дві виразні

події з піками приблизно 18.0 дБ та 19.0 дБ у моменти 0.8 та 4.1 хв відповідно. Перша подія відповідає підземній протитанковій міні (муляж) і має характерну «дзвоноподібну» огинаючу з дещо меншою амплітудою через загасання в ґрунті, що узгоджується з проникною природою електромагнітного каналу. Друга подія з більшою амплітудою інтерпретується як великий наземний металевий об'єкт, для якого відсутнє ґрунтове екранування. В обох випадках форма сигналів свідчить про оптимальне поєднання висоти  $\sim 2$  м AGL та швидкості польоту (достатній *dwell time* над ціллю). Фоновий рівень шуму протягом місії залишався стабільно низьким ( $\sim 0$ -3 дБ), що підтверджує ефективність попереднього калібрування та вибраного режиму польоту. Водночас на ділянках 1.5 та 3.5 хв, де візуальні моделі (YOLO/VLM) позначили поверхневі підозрілі об'єкти, металодетектор не зареєстрував достовірних подій, що узгоджується з їх імовірно малою металовмісністю. Таким чином, приклад наочно демонструє комплементарність сенсорів: електромагнітний канал забезпечує унікальну чутливість до підповерхневих та масивних металевих цілей, тоді як системи комп'ютерного зору ефективні для поверхневих мін із мінімальними металевими вставками.

*Сенсорно-фізичні обмеження.* Отримані результати узгоджуються з очікуваннями: рамковий датчик надійно детектує середні та великі металовмісні об'єкти як на поверхні, так і під ґрунтом; натомість невеликі протипіхотні міни з мінімальною металовмісністю (на кшталт ПФМ-1) [69] дають слабкі або нестабільні відгуки, особливо за наявності рослинності, змінного мінералізованого фону чи підвищеної висоти. На практиці це проявляється у зниженні  $P_m$  саме на класі «малих/малометалевих» цілей і підтверджує необхідність багаторівневої обробки з комп'ютерним зором для поверхневих об'єктів.

Чутливість каналу зростає зі зменшенням висоти і швидкості (більший *dwell time* над ціллю), однак надмірне зменшення висоти погіршує керованість траєкторії та збільшує ризик механічних завад. Оптимальним у

наших умовах був діапазон 2–3 м AGL, закладений у місії, що забезпечував достатній просторовий перетин рамки з ґрунтом і комфортний запас безпеки. На етапі цифрової обробки найбільший вклад у стабільність дають: (I) смугова фільтрація під робочу частоту детектора; (II) огинаюча для придушення високочастотної пульсації; (III) адаптивний поріг відносно локальних статистик фону; (IV) дебаунс/кластеризація імпульсів у події мінімальної тривалості. За таких налаштувань короткі паразитні сплески відсікаються, а справжні події формують «дзвоноподібні» квазі-симетричні пакети. Навіть після калібрування одиничні FP трапляються через (а) металеве сміття у ґрунті, (б) мінералізацію або неоднорідності ґрунту, (в) електромагнітні перешкоди (наприклад, у моменти різких змін струму на силових лініях приводу). Частота таких подій істотно зменшується після короткого «навчання на місцевості» – локального добору  $k$  та маскуванню частотних ділянок із найбільшою завадою. Для подальшої фузії ми використовуємо просторово-часове вікно узгодження з візуальними подіями, короткі одиночні сплески, що не знаходять візуального підтвердження і не повторюються, позначаються як фонові та не впливають на фінальний висновок. У наведеній місії поверхневий металевий уламок на 1.5 та 3.5 хв був зафіксований металодетектором і, у наступних експериментах, підтверджений візуальними моделями (YOLO/VLM). Натомість підземна ціль (0.8 хв) не має візуальних ознак у кадрі – і саме тут канал металодетектора єдиний забезпечує інформативність. Таким чином, у чисто металодетекторному режимі ми отримуємо унікальну чутливість до підповерхневих загроз та часткове покриття поверхневих, тоді як дрібні пластикові ППМ залишаються проблемними – це і є мотивація багатоканальної системи.

Робота металошукача у випробувальних умовах показала ймовірність виявлення  $P_m \approx 0.70$  для релевантних металовмісних об'єктів середнього/великого розміру (на поверхні та під тонким шаром ґрунту) за прийнятної частоти хибних спрацювань після локального калібрування. Канал демонструє критично важливу здатність детектувати

підповерхневі загрози, чого не можуть зробити жодні візуальні методи, але водночас обмежений щодо малих ППМ із мінімальним вмістом металу. Ці результати обґрунтовують використання багаторівневого злиття: візуальний детектор покриває поверхневі пластикові міни, а металодетектор – підповерхневі та металовмісні, а разом вони зменшують сумарну ймовірність пропуску.

#### 4.4 Злиття даних: інтеграція результатів комп'ютерного зору та металошукача

Метою фузії є перетворення множин частково перекривних подій, отриманих з різних сенсорних каналів, у єдиний послідовний набір «підтверджених» і «кандидатних» спрацьовувань із просторово-часовою прив'язкою, кількісною мірою довіри та однозначним типом підтвердження [67]. У нашій постановці існують чотири первинні джерела сигналу (рис. 4.15): детектор об'єктів YOLOv8 (множина  $D_Y$ ), два семантичні верифікатори на базі великих візуально-мовних моделей – ChatGPT-4.1 Vision і Gemini 2.5 Flash Vision (множини  $D_C$  та  $D_G$ , що формуються над ROI, відібраними з  $D_Y$ ), а також металошукач (множина  $D_M$ ) [68].

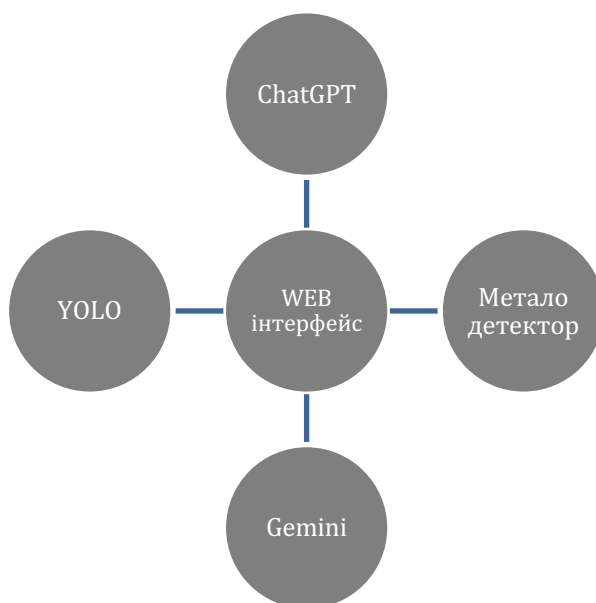


Рисунок 4.15 – Поєднання всіх елементів комплексу

Кожна подія описується часовою міткою  $t$ , географічними координатами  $x$  на поверхні, типом джерела та первинною довірою: для YOLO – це калібрована [69]  $\text{conf} \in [0,1]$  для LLM – скалярні оцінки  $s_C, s_G \in [0,1]$  для металоканалу – нормалізована амплітуда/енергія сигналу  $a_{MaM}$  і тривалість імпульсу. Першим етапом є приведення подій до спільної просторово-часової системи відліку. Кадри відео маркуються часовими штампами Raspberry Pi, синхронізованими з траєкторними логами Pixhawk 6C, а центри рамок YOLO проєктуються на землю як точки спостереження  $x_U$  із поправкою на висоту та невеликий кут крену/тангажу (враховується стабілізація та «незначний нахил конус» FOV). Події металопрошукача  $x_M$  отримують координати шляхом інтерполяції положення БПЛА за часом  $t_M$ . Алгоритмічно правило прийняття у нашій системі реалізовано в найпростішій, але максимально чутливій формі OR-логіки на рівні подій, узгоджених у просторі та часі: кластер вважається спрацюванням, якщо принаймні один канал у ньому має довіру вище локального порога, а асоційовані канали, якщо вони є, коригують  $s_{fuse}$  і клас підтвердження.

$$P(D_{\text{final}}=1|E=1)=1-(1-P_Y)(1-P_C)(1-P_G)(1-P_M) \quad (4.3)$$

Формула математичної моделі [45] показує, що шанс пропустити міну ( $D_{\text{final}}=0$  при  $E=1$ ) дорівнює добутку шансів пропустити її кожним із методів одночасно. Коли каналів багато, цей добуток стає дуже малим, отже ймовірність виявити (хоч одним із них) прямує до 1. Наприклад, якщо YOLO самостійно виявляє 80% мін ( $P_Y=0.6$ ), а комбінація ChatGPT+Gemini знайде 66% тих, що пропустив YOLO [70], і металодетектор спрацює на 65% мін ( $P_M=0.65$ ), то сукупно

$$P_{\text{detect}} \approx 1-(1-0.6)(1-0.64)(1-0.68)(1-0.65) \approx 0.98 \quad (4.4)$$

тобто  $\sim 98\%$  – значно вище, ніж будь-яким каналом окремо. Таким чином,

правило «хоч один детектор спрацював» різко підвищує ймовірність виявлення. Недоліком OR-підходу є підвищення ймовірності хибної тривоги – будь-яке помилкове спрацювання окремого методу призведе до помилкового сигналу.

У конфліктних ситуаціях застосовується інтерпретований пріоритет джерел з урахуванням фізики явища та кінцевого призначення карти ризику. Якщо об'єкт візуально очевидний, але металоканал мовчить, подія з високою ймовірністю відповідає пластиковій поверхневій міні (клас V/V-LLM) і не повинна занижуватися – навпаки,  $w_Y$  та  $w_{C,G}$  забезпечують прийняття за умови перевищення  $\tau^*$ . Якщо металодетектор спрацює впевнено, а в кадрі не спостерігається жодної стабільної рамки, ми маркуємо подію як «потенційна підземна ціль» (клас M) і відображаємо її на мапі з відповідною піктограмою/кольором та підвищеною вагою в heatmap; у таких випадках саме  $w_{Mh(aM)}$  є вирішальним компонентом  $s_{fuse}$ . У прикордонних випадках, коли є слабкий метало-сигнал і низька візуальна впевненість, рішення ухвалюється на підставі  $s_{fuse}$  із підвищеним порогом та збереженням події у «кандидати» для повторного огляду або місій повторного обльоту.

Результати інтеграції візуалізуються у хмарному інтерфейсі як інтерактивна мапа з двома основними шарами: точковим шаром подій (із кодуванням типів підтвердження та величини  $s_{fuse}$ ) та шаром heatmap, який відображає просторову щільність подій із вагами, пропорційними  $s_{fuse}$  та класу підтвердження ( $VM > V-LLM > V \approx M$ ). До кожної точки прив'язуються картки з первинними кадрами, анотованими рамками YOLO, витягами LLM-вердиктів, формою металосигналу у часі та коротким технічним резюме. Додатково ми формуємо «комплексний графік» місії – часову панель із накладенням довіри YOLO, бінарних індикаторів ChatGPT-4.1 та Genimi 2.5 Flash, а також амплітуди металосигналу. Цей графік демонструє типовий патерн узгодження для поверхневих об'єктів і унікальність металоканалу для підповерхневих.

Таким чином, інтеграційний модуль перетворює гетерогенні сигнали на узгоджені просторово-часові події з прозорою логікою прийняття рішень, враховуючи фізичні межі кожного каналу. OR-фузія забезпечує максимальну чутливість системи: підповерхневі та металовмісні загрози виявляються металоканалом, поверхневі пластикові – візуальною підсистемою, а LLM-верифікація підсилює специфічність на складних фонах. Хмарна агрегація з інтерфейсом мапи та heatmap завершує цикл, роблячи результати зручними для оперативного використання та подальшого планування повторних обльотів і ручної верифікації.

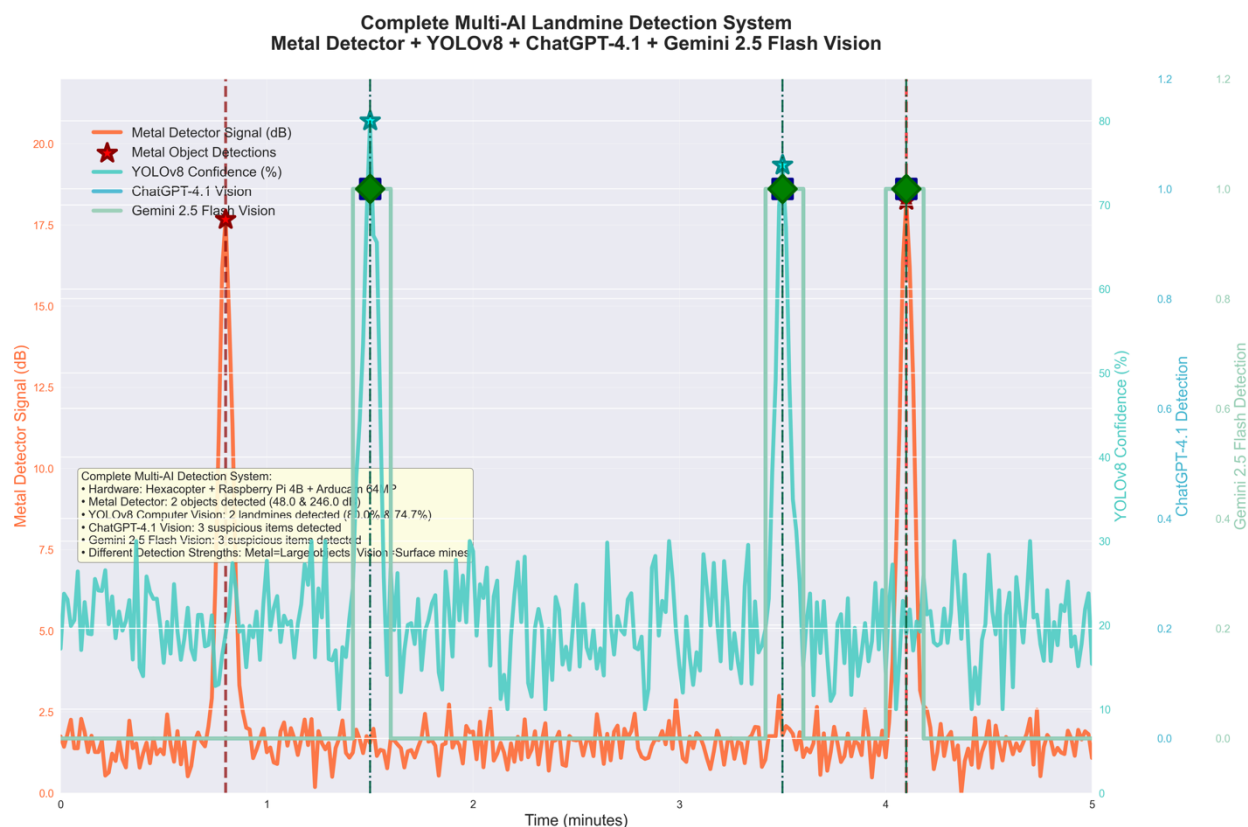


Рисунок 4.16 – Комплексне відображення накладених результатів

Рисунок 4.16 відображає інтегровану часову панель п'ятихвилинної місії з накладенням усіх каналів: амплітуди металодетектора (у дБ), конфіденції YOLOv8 для поверхневих детекцій, а також бінарних спрацювань візуально-мовних моделей ChatGPT-4.1 Vision і Gemini 2.5 Flash Vision. Візуалізація побудована на спільній шкалі часу з просторово-часовим

узгодженням подій (допуски  $t_{tol}, \Delta t_{tol}$ ) та кольоровим кодуванням типів підтвердження: M (тільки метал), V (тільки візуально), V-LLM (візуально з підтвердженням VLM), VM. На графіку зафіксовано вісім подій: дві металодетекторні ( $\sim 0.8$  хв,  $\sim 4.1$  хв; піки  $\sim 18.0$  і  $\sim 19.0$  дБ), дві поверхневі детекції YOLOv8 ( $\sim 1.5$  і  $\sim 3.5$  хв; пікові довіри  $\sim 80\%$  і  $\sim 75\%$ ), три спрацювання ChatGPT-4.1 Vision (з підтвердженням YOLO [70] на 1.5 і 3.5 хв та додатковий епізод на 4.1 хв) і одне незалежне підтвердження Genimi 2.5 Flash Vision на 4.1 хв. Така конфігурація демонструє повну узгодженість візуальних каналів на поверхневих цілях і водночас підкреслює ексклюзивність електромагнітного каналу для підповерхневих/масивних металевих об'єктів: у точці  $\sim 0.8$  хв суттєвий металосигнал на тлі повної візуальної тиші інтерпретується як підземна загроза, тоді як у точках  $\sim 1.5$  і  $\sim 3.5$  хв навпаки – стійкі візуальні детектування без металовідгуку відповідають імовірно пластиковим поверхневим мінам із мінімальною металовмісністю. Сцена  $\sim 4.1$  хв цікава крос-валідацією в усіх візуальних каналах (ChatGPT-4.1 + Genimi 2.5 Flash) і одночасно сильним металосигналом, що інтерпретується як великий поверхневий металевий об'єкт або фрагмент.

З наукового погляду графік фіксує межі можливостей і доповнюваність сенсорів: YOLOv8 оптимальний для швидкої геометричної локалізації дрібних поверхневих цілей; VLM-моделі підвищують семантичну стійкість на «важких» фонах і дають незалежні підтвердження; металодетектор забезпечує унікальну чутливість до підповерхневих/масивних металевих об'єктів і виконує роль фізичного підтверджувача там, де оптика безсила. У межах конкретної місії, відображеної на рисунку 4.17, система продемонструвала повну операційну справність усіх підсистем, відсутність пропусків контрольних цілей і повне узгодження між незалежними візуальними моделями на поверхневих подіях, тоді як підповерхнева подія підтверджена виключно металоканалом.



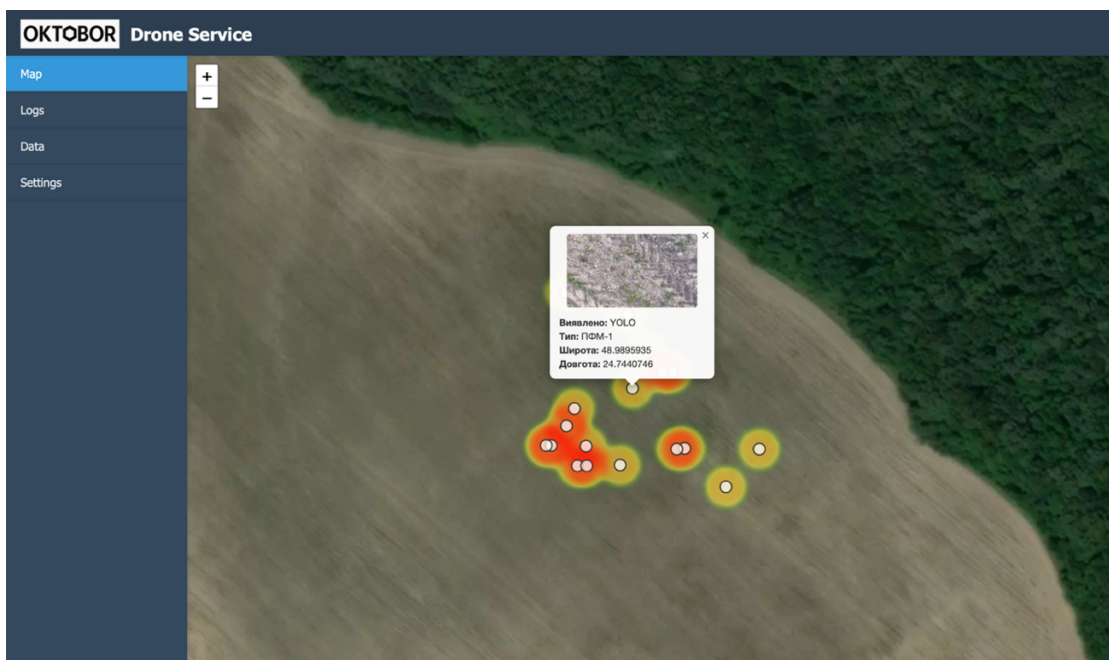


Рисунок 4.17 – Частина хмарного інтерфейсу з результатами роботи комплексу

Це якісно узгоджується з теорією OR-фузії: інтегральна ймовірність виявлення «хоч одним каналом» різко зростає, оскільки шанс одночасного пропуску всіма каналами дорівнює добутку їхніх індивідуальних шансів пропуску. Отримана на полігоні операційна оцінка  $P_{\text{detect}} \approx 0.98$  відбиває реальний виграш багаторівневої інтеграції з урахуванням кореляцій і часткового покриття LLM, тоді як теоретична межа за припущення квазі незалежності лежить вище.

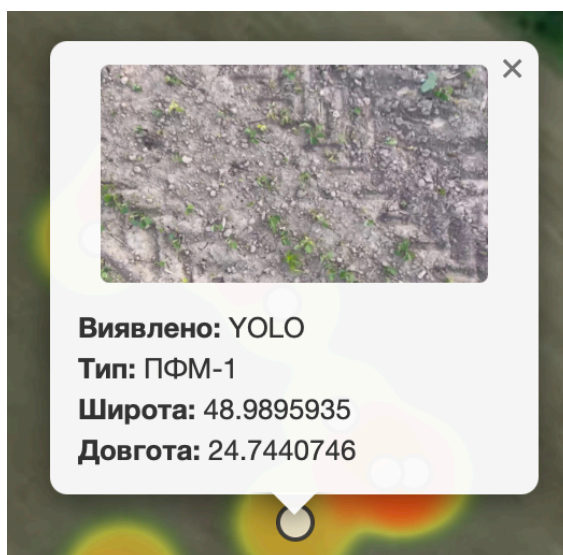


Рисунок 4.18 – Окремо ідентифікований предмет інтерфейсу

Практично комплексний графік використовується як «єдина правда» для післяпольотного аналізу і побудови heatmap на мапі: кожна вертикальна подія на графіку має свій геоприв'язаний маркер (рис. 4.18), а вага в тепловій карті пропорційна зведеному скору  $s_{fuse}$  і класу підтвердження (найвища для VM, далі VLM, потім V і M). Така подача дозволяє одночасно бачити часову синхронізацію сенсорів, їхню крос-валідацію та просторову концентрацію ризику, що є критичним для планування повторних обльотів, ручної верифікації та формування протоколів гуманітарного розмінування. Варто підкреслити суттєві переваги розробленої системи у порівнянні з існуючими аналогами (Таб. 4.1), розглянутими в розділі 1. На відміну від концепції «роздільного картування» та методівного моніторингу, де результати стають доступними лише після завершення місії та постобробки, запропонований підхід реалізує активний замкнений цикл керування в режимі реального часу. Інтеграція протоколу MAVLink дозволяє системі миттєво реагувати на невпевнені детекції (наприклад, при ймовірності 0,4–0,7), автоматично знижуючи висоту польоту або виконуючи повторний прохід для отримання якісніших даних. Це фундаментально підвищує достовірність результатів безпосередньо під час польоту, перетворюючи дрон із пасивного реєстратора на інтелектуального агента, що адаптується до умов середовища. Крім того, використання візуально-мовних моделей (VLM) дозволяє подолати критичне обмеження традиційних нейромереж (CNN), які використовуються в роботах [10, 11] Neuschmid et al. та Vivoli et al., – залежність від обмеженого навчального датасету. У той час як детектори типу YOLOv8 можуть пропускати нестандартні або замасковані об'єкти, що не були присутні у тренувальних даних, VLM завдяки своїй «загальній ерудиції» здатні ідентифікувати аномалії на основі семантичного контексту та візуальних ознак, які не були явно запрограмовані.

Таблиця 4.1 – Порівняння метрик

| Дослідження / Система        | Сенсори                  | Метод обробки                                        | Метрика ефективності (Best Case)     | Обмеження                             |
|------------------------------|--------------------------|------------------------------------------------------|--------------------------------------|---------------------------------------|
| <b>Наша система</b>          | <b>RGB + Метал + VLM</b> | <b>Weighted Confidence Aggregation + Active Loop</b> | <b><math>P_d \approx 98\%</math></b> | <b>Залежність від зв'язку (Cloud)</b> |
| Stankevich & Saprykin (2024) | RGB + Mag                | Bayesian Fusion                                      | $P_{\text{fuse}} \approx 96\%$       | Пасивний моніторинг (Open Loop)       |
| Vivoli et al. (2024)         | RGB (YOLOv8)             | Deep Learning                                        | $\text{Recall} \approx 98.4\%$       | Тільки поверхневі, чистий фон         |
| Heuschmid et al. (2025)      | Thermal                  | CNN                                                  | $\text{Accuracy} \approx 97.2\%$     | Залежність від часу доби              |

Експериментальні дані підтверджують, що додавання семантичного шару перевірки дозволяє виявляти складні цілі, пропущені стандартними детекторами, та ефективно фільтрувати хибні спрацювання на побутове сміття, забезпечуючи високу загальну ефективність системи на рівні 98%.

#### 4.5 Висновки за розділом 4

Четвертий розділ містить результати всебічного експериментального дослідження розробленого апаратно-програмного комплексу. Серія натурних випробувань, проведених у польових умовах із застосуванням реальних макетів вибухонебезпечних предметів (зокрема мін ПФМ-1), підтвердила адекватність створених математичних моделей, високу швидкодію системи та її беззаперечну операційну ефективність. Для об'єктивного оцінювання неймережевого детектора YOLOv8 було попередньо сформовано унікальний навчальний датасет, що містить 2500 анотованих кадрів мін ПФМ-1 у складних умовах (змішаний ґрунт, густа трава, різка зміна освітленості та часткове маскування тінями). Експериментальне тестування суто візуального (моносенсорного) конвеєра продемонструвало його високу здатність до виявлення поверхневих об'єктів (значення Recall досягало 1.000 на тестовій вибірці). Однак ці випробування також чітко окреслили фізичні межі оптичних систем: у випадках, коли об'єкт був повністю або значно прихований у ґрунті,

візуальний канал закономірно не фіксував загрозу. З іншого боку, тестування каналу металодетектора Tracker-2 підтвердило його високу ефективність у виявленні масивних або підповерхневих металевих предметів (ймовірність виявлення  $\sim 0.70$ ), проте виявило недостатню чутливість до пластикових мін з мінімальним вмістом металу при збільшенні висоти польоту.

Ці обмеження емпірично довели абсолютну необхідність впровадженої моделі мультимодального злиття даних. Застосування правила логічної агрегації (детекція підтверджується, якщо хоча б один канал сигналізує про загрозу) кардинально підвищило загальну ймовірність виявлення. Теоретичне припущення про взаємодоповнюваність сенсорів було підтверджено на практиці: оптичний канал успішно покривав поверхневі пластикові загрози, а металодетектор надійно ідентифікував підповерхневі цілі. У результаті інтегральна точність розпізнавання системи в умовах невизначеності зовнішнього середовища сягнула  $\sim 98\%$ .

Масштабні тестування хмарного рівня семантичної верифікації за допомогою візуально-мовних моделей (VLM) довели їхню ключову роль у стабілізації системи. Під час обробки складних кадрів модель Gemini 2.5 Flash продемонструвала високу швидкість та точність на рівні окремих зображень ( $\sim 67.6\%$  правильних верифікацій), у той час як ChatGPT-4.1 Vision показав кращі результати у забезпеченні повноти детектування на рівні цілісних об'єктів. Стратегія перехресної верифікації між двома незалежними VLM дозволила нівелювати слабкі сторони кожної моделі. Такий інтелектуальний фільтр ефективно розпізнавав природні об'єкти або побутове сміття, що мають візуальну схожість з мінами, завдяки чому показник хибних тривог був знижений до безпрецедентного рівня у 2-4%. Це має високе практичне значення, оскільки кожна хибна тривога в реальному розмінуванні призводить до невиправданої зупинки робіт та ризику для саперів.

## ВИСНОВКИ

У першому розділі проведено системний аналіз сучасного стану та методів автоматизованого керування роботизованими платформами для задач просторового моніторингу і гуманітарного розмінування. Це включає детальне дослідження існуючих апаратних рішень, парадигм розподілених обчислень (Edge, Ground, Cloud) та розвитку методів комп'ютерного зору. Особливу увагу приділено виявленню ключових обмежень моносенсорних систем та обґрунтуванню необхідності створення гібридної ієрархічної архітектури.

У другому розділі розроблено апаратну архітектуру багаторівневої роботизованої платформи на базі мультироторної схеми гексакоптера, яка дозволила забезпечити необхідну відмовостійкість та стабільність. Додатково розроблено структуру просторово-часової синхронізації гетерогенних сенсорних каналів (оптичної камери, металодетектора та лідара) з бортовим обчислювачем, що дозволило оптимізувати процес збору даних та підготувати їх для подальшого аналізу.

Третій розділ присвячено розробці алгоритмічного забезпечення та вдосконаленню методів мультимодального злиття даних. Застосування розробленої математичної моделі зваженої агрегації довіри, яка об'єднує результати нейромережі YOLOv8, візуально-мовних моделей (VLM) та металодетектора, дозволило мінімізувати помилки ідентифікації. Впровадження методу автоматичного адаптивного керування польотом дозволило перетворити платформу на проактивного агента із замкненим контуром корекції місії.

Четвертий розділ містить результати експериментальних досліджень, які підтверджують ефективність розроблених моделей і методів. Запропонована багаторівнева система дозволила досягти загальної точності розпізнавання вибухонебезпечних предметів на рівні 98% та безпрецедентно знизити показник хибних тривог (FAR) до 2-4%. Практична реалізація методів

відображена в інтеграції системи в навчальні програми та в реальних виробничих умовах інжинірингових підприємств.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Advances in UAV Avionics Systems Architecture, Classification and Integration: A Comprehensive Review and Future Perspectives. arXiv, 2025, pp. 1–32. URL: <https://arxiv.org/html/2501.00856v1>
2. Control and navigation system for a fixed-wing unmanned aerial vehicle. AIP Publishing, 2014, Vol. 4, No. 3, pp. 1–15. URL: <https://pubs.aip.org/aip/adv/article/4/3/031306/21731/Control-and-navigation-system-for-a-fixed-wing>
3. Johansen T. A. Guidance, Navigation and Control of Fixed-Wing Unmanned Aerial Vehicles. NTNU, 2007, pp. 1–45. URL: [https://torarnj.folk.ntnu.no/fixedwing\\_uav.pdf](https://torarnj.folk.ntnu.no/fixedwing_uav.pdf)
4. Simple Overview of ArduPilot Operation. Copter documentation, 2024, pp. 1–6. URL: <https://ardupilot.org/copter/docs/common-basic-operation.html>
5. Reverse Engineering and Control-Aware Security Analysis of the ArduPilot UAV Framework. arXiv, 2025, pp. 1–18. URL: <https://arxiv.org/html/2512.01164v1>
6. Vision-Based Autonomous UGV Detection, Tracking, and Following for a UAV. ResearchGate, 2024, pp. 1–14. URL: [https://www.researchgate.net/publication/377804951\\_Vision-Based\\_Autonomous\\_UGV\\_Detection\\_Tracking\\_and\\_Following\\_for\\_a\\_UAV](https://www.researchgate.net/publication/377804951_Vision-Based_Autonomous_UGV_Detection_Tracking_and_Following_for_a_UAV)
7. Hutsul T., Khobzei M., Tkach V., Krulikovskiy O., Moisiuk O., et al. Review of approaches to the use of unmanned aerial vehicles, remote sensing and geographic information systems in humanitarian demining: Ukrainian case. Heliyon, 2024, Vol. 10, Issue 7, pp. 1–22. DOI: 10.1016/j.heliyon.2024.e29142

8. Yoo L.-S., Lee J.-H., Lee Y.-K., Jung S.-K., Choi Y. Application of a Drone Magnetometer System to Military Mine Detection in the Demilitarized Zone. *Sensors*, 2021, Vol. 21, No. 9, Art. 3175, pp. 1–16. DOI: 10.3390/s21093175
9. Kuru K., Sujit A., Ansell D., Pinder J. M., Jones D., Watkinson B. J., Hamila R., Tinker-Mill C. Intelligent, Automated, Rapid, and Safe Landmine, Improvised Explosive Device and Unexploded Ordnance Detection Using Maggy. *IEEE Access*, 2024, Vol. 12, pp. 432-450. DOI: 10.1109/ACCESS.2024.3493013
10. Heuschmidt D., Wacker O., Zimmermann Y., Penava P., Buettner R. Advancements in landmine detection: Deep learning-based analysis with thermal drones. *IEEE Access*, 2025, Vol. 13, pp. 1–15. DOI: 10.1109/ACCESS.2025.3572196
11. Vivoli E., Bertini M., Capineri L. Deep Learning-Based Real-Time Detection of Surface Landmines Using Optical Imaging. *Remote Sensing*, 2024, Vol. 16, No. 4, Art. 677, pp. 1–18. DOI: 10.3390/rs16040677
12. Stankevich S. A., Saprykin I. Y. Optical and magnetometric data integration for landmine detection with UAV. *WSEAS Transactions on Environment and Development*, 2024, Vol. 20, pp. 132–141. DOI: 10.37394/232015.2024.20.96
13. LeCun Y., Bottou L., Bengio Y., Haffner P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998, Vol. 86, No. 11, pp. 2278–2324.
14. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 2012, Vol. 25, pp. 1097–1105.
15. Gonzalez R.C., Woods R.E. *Digital Image Processing (4th Edition)*. Pearson, 2018, pp. 1–192. URL: <https://www.cl72.org/090imagePLib/books/Gonzales,Woods-Digital.Image.Processing.4th.Edition.pdf>

16. Shakya B., El-Gayar O., Kaabi J., Ahmed K. M. Small-Object Detection at the Edge: A Pareto-Efficient Benchmark of Lightweight YOLO Models on UAV and Overhead Datasets. *IEEE Access*, 2026, Vol. 14, pp. 1–15.
17. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv*, 2015, pp. 1–14.
18. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition. *CVPR*, 2016, pp. 770–778.
19. Huang G., Liu Z., Van Der Maaten L., Weinberger K. Q. Densely Connected Convolutional Networks. *CVPR*, 2017, pp. 470–478.
20. Dosovitskiy A., Beyer L., Kolesnikov A. et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv*, 2020, pp. 1–22.
21. Liu Z., Mao H., Wu C. Y. et al. ConvNeXt: Rethinking Convolutional Networks. *CVPR*, 2022, pp. 128–238.
22. Tan M., Le Q. V. EfficientNet: Rethinking Model Scaling for CNN. *ICML*, 2019, pp. 615–614.
23. Radford A., Kim J. W., Hallacy C., Ramesh A., Goh G. et al. Learning Transferable Visual Models from Natural Language Supervision (CLIP). *arXiv*, 2021, pp. 1–48. DOI: 10.48550/arXiv.2103.00020
24. Andriulo F.C., Fiore M., Mongiello M., Traversa E., Zizzo V. Edge Computing and Cloud Computing for Internet of Things: A Review. *Informatics*, 2024, Vol. 11, Art. 71, pp. 1–25. URL: <https://doi.org/10.3390/informatics11040071>
25. Optimizing Edge AI: A Comprehensive Survey on Data, Model, and System Strategies. *arXiv*, 2025, pp. 1–30. URL: <https://arxiv.org/html/2501.03265v1>
26. A Comprehensive Review of UAV-UGV Collaboration: Advancements and Challenges. *MDPI*, 2024, pp. 1–28. URL: <https://www.mdpi.com/2224-2708/13/6/81>
27. UAV Data Acquisition and Processing Assisted by UGV-Enabled Mobile Edge Computing. *ResearchGate*, 2024, pp. 1–12. URL:



[https://www.researchgate.net/publication/388815571\\_UAV\\_Data\\_Acquisition\\_and\\_Processing\\_Assisted\\_by\\_UGV-Enabled\\_Mobile\\_Edge\\_Computing](https://www.researchgate.net/publication/388815571_UAV_Data_Acquisition_and_Processing_Assisted_by_UGV-Enabled_Mobile_Edge_Computing)

28. Exploring Autonomous Load-Carrying Mobile Robots in Indoor Settings. IEEE Xplore, 2024, pp. 1–8. URL: <https://ieeexplore.ieee.org/iel8/6287639/6514899/10614599.pdf>

29. Streamlined Acquisition of Large Sensor Data for Autonomous Mobile Robots to Enable Efficient Creation and Analysis of Datasets. IEEE Xplore, 2024, pp. 1–10. URL: <https://ieeexplore.ieee.org/document/10611096/>

30. Yue J. Robotic AI Using JaguarLite Embedded Vector Database. Medium, 2024, pp. 1–5. URL: <https://jaguardb.medium.com/robotic-ai-using-jaguarlite-embedded-vector-database-85f255c9fba0>

31. Silvano C. et al. A Survey on Deep Learning Hardware Accelerators for Edge Computing. ACM Computing Surveys, 2025, Vol. 57, No. 3, pp. 1–35. URL: <https://doi.org/10.1145/3729215>

32. Tahir N., Parasuraman R. Edge Computing and Its Application in Robotics: A Survey. Journal of Sensor and Actuator Networks, 2025, Vol. 14, No. 4, Art. 65, pp. 1–24. DOI: 10.3390/jsan14040065

33. Роботько С.П. Багаторівневий аналіз відео зображення на базі Raspberry Pi та ChatGPT 4o Vision. Scientific and Technical Journal «Methods and devices of quality control», 2024, № 1(54), pp. 78–88. DOI: 10.31471/1993-9981-2025-1(54)-78-88

34. Dong C., Wang J., Robotko S.P., Susak O.M., Topalov A.M., Kolomiets V.V. Features of human-machine interaction in the system of wireless. Міжвузівський збірник «Наукові нотатки», Луцьк, 2023, № 76, pp. 1–7. DOI: 10.36910/775.24153966.2023.76.23

35. Роботько С., Луцак Д. Багаторівневий підхід виявлення та розпізнавання небезпечних предметів за допомогою БПЛА із застосуванням комп'ютерного зору. Вісник Харківського національного автомобільно-дорожнього університету, 2025, № 111, pp. 1–10. DOI: 10.30977/BUL.2219-5548.2025.111.0

36. Stamate M.-A., Pupăză C., Nicolescu F.-A., Moldoveanu C.-E. Improvement of Hexacopter UAVs Attitude Parameters Employing Control and Decision Support Systems. *Sensors*, 2023, Vol. 23, Art. 1446, pp. 1–19. URL: <https://doi.org/10.3390/s23031446>
37. Xiao H., Krisshnakumar P., Pothuri J. P. K. V. et al. An Open-source Hardware/Software Architecture and Supporting Simulation Environment to Perform Human FPV Flight Demonstrations for Unmanned Aerial Vehicle Autonomy. *arXiv*, 2024, pp. 1–14. URL: <https://arxiv.org/html/2407.06320v2>
38. Pixhawk 6C Ports. Official Documentation, 2024, pp. 1–4. URL: <https://docs.holybro.com/autopilot/pixhawk-6c/pixhawk-6c-ports>
39. Pastor E., Lopez J., Royo P. A. Hardware/Software architecture for UAV payload and mission control. *Proceedings of the 2006 IEEE/AIAA 25TH Digital Avionics Systems Conference*, 2006, pp. 1–8.
40. Saleh T., Aboukassem K. Study the Equations of Motion for a Hexacopter Aerial Vehicle based on The Newton-Euler Method. *International Journal of Computer Science Trends and Technology (IJCTST)*, 2018, Vol. 6, Issue 4, pp. 1–6. URL: <https://www.ijctstjournal.org/volume-6/issue-4/IJCTST-V6I4P4.pdf>
41. Larionova S., Marques L., de Almeida A. Multi-Stage Sensor Fusion for Landmine Detection. *IROS*, 2006, pp. 343–348. DOI: 10.1109/IROS.2006.282148
42. Liu P., Yuan W., Fu J., Jiang Z., Hayashi H., Neubig G. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Computing Surveys*, 2023, Vol. 55, No. 9, pp. 1–35.
43. Robotko S. AI-Driven OCR in Practice: Assessing ChatGPT-4o and Google Vision as Alternatives to Manual Optical Character Recognition. *2025 15th International Conference on Advanced Computer Information Technologies (ACIT)*, 2025, pp. 14–65. DOI: 10.1109/ACIT65614.2025.11185827
44. Robotko S., Zivenko O. Effectiveness of vision–language models (VLMs) for ground-object recognition in a multi-level Edge–cloud UAV architecture. *Науковий вісник Херсонської державної морської академії*, 2025, Том 2, № 31, pp. 19–29. DOI: 10.33815/2313-4763.2025.2.31.019-029

45. Robotko S., Topalov A., Nechai I., Vasylenko R. Multimodal data fusion for security screening with weighted confidence aggregation and escalation in Edge-cloud systems. *Proceedings of the Bulgarian Academy of Sciences*, 2026, Vol. 79, No. 1, pp. 13–50. DOI: 10.7546/CRABS.2026.01.09
46. Tracker PI-2 metal detector. Simple Metal Detector, 2024, pp. 1–6. URL: <https://simplemetaldetector.com/pulse-induction-metal-detectors/tracker-pi-2-metal-detector/?amp=1>
47. Lekhak S., Ientilucci E.J., Brinkley A.W. Viability of Substituting Handheld Metal Detectors with an Airborne Metal Detection System for Landmine and Unexploded Ordnance Detection. *Remote Sensing*, 2024, Vol. 16, No. 24, Art. 4732, pp. 1–17. DOI: 10.3390/rs16244732
48. Šipoš D., Gleich D. A Lightweight and Low-Power UAV-Borne Ground Penetrating Radar Design for Landmine Detection. *Sensors*, 2020, Vol. 20, No. 8, Art. 2234, pp. 1–15. DOI: 10.3390/s20082234
49. Official Raspberry Pi documentation. 2024, pp. 1–12. URL: <https://www.raspberrypi.com/documentation/>
50. Official Arducam 64MP camera documentation. 2024, pp. 1–5. URL: <https://docs.arducam.com/Raspberry-Pi-Camera/Native-camera/64MP-OV64A40/>
51. Sapkota R., Meng Z. Comprehensive performance evaluation of YOLO11, YOLOv10, YOLOv9 and YOLOv8 on detecting and counting fruitlet in complex orchard environments. *arXiv preprint*, 2024, No. 2407.12040, pp. 1–18. URL: <https://arxiv.org/abs/2407.12040>
52. Li Z., Wu X. A Survey of State of the Art Large Vision-Language Models. *CVPR*, 2025, No. 1, pp. 1–25. URL: <https://github.com/zli12321/Vision-Language-Models-Overview>
53. Qian H., Wang M., Zhu M., Wang H. A Review of Multi-Sensor Fusion in Autonomous Driving. *Sensors*, 2025, Vol. 25, No. 19, Art. 6033, pp. 1–22. DOI: 10.3390/s25196033

54. Lalinia M., Sahafi A. Colorectal polyp detection in colonoscopy images using YOLO-V8 network. *Signal, Image and Video Processing*, 2023, Vol. 18, pp. 1–12. DOI: 10.1007/s11760-023-02835-1

55. Kadri A. A comprehensive software framework for cloud-enabled autonomous drone navigation. *PeerJ Computer Science*, 2025, No. 11, pp. 1–20. DOI: 10.7717/peerj-cs.3238

56. Роботько С.П., Топалов А.М. Багаторівнева архітектура системи автоматичного керування БПЛА для здійснення пошукових місій за відеоаналізом та металодетекцією. *Оптико-електронні інформаційно-енергетичні технології*, 2025, Том 50, № 2, pp. 1–14. DOI: 10.31649/1681-7893-2025-50-2

57. Langelaan J. W., Alley N., Neidhoefer J. Wind Field Estimation for Small Unmanned Aerial Vehicles. *AIAA Guidance, Navigation, and Control Conference*, 2010, AIAA 2010-8177, pp. 1–16. DOI: 10.2514/6.2010-8177

58. Zheng Y., Robotko S.P., Dong C., Wang J., Ravliuk V.V., Topalov A.M. Cloud-based GPS signal processing of the drone using Raspberry Pi and Pixhawk 6c flight controller. *Вчені записки Таврійського національного університету імені В.І. Вернадського*, 2023, Том 34 (73), № 6, pp. 1–8. DOI: 10.32782/2663-5941/2023.6/24

59. Johansen T. A. *Guidance, Navigation and Control of Fixed-Wing Unmanned Aerial Vehicles*. NTNU, 2007, pp. 1–45. URL: [https://torarnj.folk.ntnu.no/fixedwing\\_uav.pdf](https://torarnj.folk.ntnu.no/fixedwing_uav.pdf)

60. Official Label Studio documentation. 2024, pp. 1–10. URL: <https://labelstud.io/guide/>

61. Li C., Zuo Z., Tong X., Huang H., Yuan S., Dang Z. CPROS: A Multimodal Decision-Level Fusion Detection Method Based on Category Probability Sets. *Remote Sensing*, 2024, Vol. 16, No. 15, Art. 2745, pp. 1–20. DOI: 10.3390/rs16152745

62. Роботько С. П. Багаторівневий аналіз відеозображення на базі Raspberry Pi та ChatGPT-4 Vision. *Методи та прилади контролю технологічних*

параметрів, 2025, № 1(54), pp. 78–88. DOI: 10.31471/1993-9981-2025-1(54)-78-88

63. Johansen T. A. Guidance, Navigation and Control of Fixed-Wing Unmanned Aerial Vehicles. NTNU, 2007, pp. 1–45. URL: [https://torarnj.folk.ntnu.no/fixedwing\\_uav.pdf](https://torarnj.folk.ntnu.no/fixedwing_uav.pdf)

64. Hamza M., Mohsin M., Khalil M., Kazmi S. M. K. A. MAVLink Protocol: A Survey of Security Threats and Countermeasures. 2024 International Conference on Digital Technologies (ICoDT), 2024, pp. 1–8.

65. Official Radiomaster documentation. 2024, pp. 1–35. URL: [https://cdn.shopify.com/s/files/1/0609/8324/7079/files/TX12MKII\\_User\\_Manual.pdf?v=1712910668](https://cdn.shopify.com/s/files/1/0609/8324/7079/files/TX12MKII_User_Manual.pdf?v=1712910668)

66. Li C., Zuo Z., Tong X., Huang H., Yuan S., Dang Z. CPROS: A Multimodal Decision-Level Fusion Detection Method Based on Category Probability Sets. Remote Sensing, 2024, Vol. 16, No. 15, Art. 2745, pp. 1–20. DOI: 10.3390/rs16152745

67. Malizia M., Hasselmann K., Miuccio A., Haelterman R., Tsiogkas N., Demeester E. PFM-1 Landmine Detection in Vegetation Using Thermal Imaging with Limited Training Data. 2025 25th International Conference on Control, Automation and Systems (ICCAS), 2025, pp. 1864–1869.

68. Koilia N., Kachris C. Hardware Acceleration of LLMs: A comprehensive survey and comparison. arXiv, 2024, pp. 1–25. URL: <https://arxiv.org/html/2409.03384v1>

69. Gonzalez R. C., Woods R. E. Digital Image Processing. 4-th ed. Pearson, 2018, pp. 90–154.

70. Artale V., Collotta M., Pau G., Ricciardello A. Hexacopter Trajectory Control using a Neural Network. AIP Conference Proceedings, 2013, Vol. 1558, pp. 216–219. DOI: 10.1063/1.4825729

## ДОДАТОК А

Програмний код реалізації хмарної обробки кадрів за допомогою візуально-мовних моделей

```
import os
from pathlib import Path
from dotenv import load_dotenv

# Load environment variables
load_dotenv()

class Config:
    """Configuration class for API keys and project settings."""

    # API Keys
    GPT_API_KEY = os.getenv("GPT_API_KEY")
    GEMINI_API_KEY = os.getenv("GEMINI_API_KEY")

    # Project paths
    PROJECT_ROOT = Path(__file__).parent
    DATASET_PATH = PROJECT_ROOT / "dataset"
    RESULTS_PATH = PROJECT_ROOT / "results"

    # Model settings
    GPT_MODEL = "gpt-4.1" # GPT-5 with vision capabilities
    GEMINI_MODEL = "gemini-2.5-flash" # Gemini 2.5 Flash model

    DETECTION_PROMPT = (
        "You are an image safety flagger for a humanitarian demining dataset. Your
job "
```

```

        "is only to flag if an image likely contains an ordnance/UXO/mine-like object.
    "
    "Respond with: YES (suspicious ordnance-like object visible) or NO (none
    visible). "
    "Shortly describe type if you can."
)

```

```

# Image settings

```

```

MAX_IMAGE_SIZE = (1024, 1024) # Max dimensions for image processing

```

```

@classmethod

```

```

def validate(cls):

```

```

    """Validate that all required configurations are set."""

```

```

    if not cls.GPT_API_KEY:

```

```

        raise ValueError("GPT_API_KEY not found in environment variables")

```

```

    if not cls.GEMINI_API_KEY:

```

```

        raise ValueError("GEMINI_API_KEY not found in environment variables")

```

```

    if not cls.DATASET_PATH.exists():

```

```

        raise ValueError(f"Dataset path does not exist: {cls.DATASET_PATH}")

```

```

# Create results directory if it doesn't exist

```

```

cls.RESULTS_PATH.mkdir(exist_ok=True)

```

---

```

def run_frame_analyzing(self, limit_per_subdir: int = None, output_path:
Optional[Path] = None) -> pd.DataFrame:

```

```

    images_by_subdir = self.dataset_manager.get_all_images()

```

```

# Load already processed images if output file exists

```

```

if output_path:

```

```

self.output_file = output_path
self.processed_images = self._load_processed_images(output_path)

total_to_process = 0
for subdir, images in images_by_subdir.items():
    if limit_per_subdir:
        images = images[:limit_per_subdir]
        print(f" - {subdir}: {len(images)} images")
        total_to_process += len(images)

if self.processed_images:
    # Process each image with each detector
    for subdir_name, images in images_by_subdir.items():
        if limit_per_subdir:
            images = images[:limit_per_subdir]
            print(f"\nProcessing subdirectory: {subdir_name}")
            for image_path in tqdm(images, desc=f"{subdir_name}"):
                # Check if image already processed
                img_id = f"{subdir_name}/{image_path.name}"
                if img_id in self.processed_images:
                    continue # Skip already processed images

            # Collect results from all detectors for this image
            image_results = {}
            image_results['subdirectory'] = subdir_name
            image_results['image_name'] = image_path.name

            for detector in self.detectors:
                result = self._process_single_image(detector, image_path,
subdir_name)

```



```

self.results.append(result)

# Add to image_results for immediate CSV writing
model_name = detector.get_model_name()
if 'gpt' in model_name.lower():
    image_results['gpt_detected'] = result['detected']
    image_results['gpt_response'] = result['response'] if result['response']
else result['error']

    elif 'gemini' in model_name.lower():
        image_results['gemini_detected'] = result['detected']
        image_results['gemini_response'] = result['response'] if
result['response'] else result['error']

# Save row immediately after both models respond
if self.output_file:
    self._append_row_to_csv(image_results, self.output_file)
    self.processed_images.add(img_id)

# Convert results to DataFrame
df = pd.DataFrame(self.results)
return df

```

## ДОДАТОК Б

Програмний код для розгортання інтерфейсу для роботи оператора

```
version: '3.8'
```

```
services:
```

```
  web:
```

```
    build: .
```

```
    ports:
```

```
      - "5001:5000"
```

```
    volumes:
```

```
      - ../app
```

```
    environment:
```

```
      FLASK_APP: app
```

```
      FLASK_ENV: development
```

```
      FLASK_RUN_PORT: 5000
```

```
      MYSQL_HOST: host.docker.internal
```

```
    extra_hosts:
```

```
      - "host.docker.internal:host-gateway"
```

```
volumes:
```

```
  db_data: # Add this volume
```

---

```
import mysql.connector
```

```
from flask import current_app
```

```
def get_drone_data():
```

```

# server = '49.12.213.10' # Replace with your MySQL server IP or hostname
# database = 'da_vinci'
# username = 'serhii' # Replace with your MySQL username
# password = 'pbGb#bxV64Uo89' # Replace with your MySQL password

try:
    conn = mysql.connector.connect(
        host="host.docker.internal", # <- host Mac from inside Docker
        port=3306,
        database="da_vinci",
        user="serhii", # don't use root remotely
        password="pbGb#bxV64Uo89",
        connection_timeout=3,
    )
    if conn.is_connected():
        cursor = conn.cursor(dictionary=True)
        cursor.execute(
            'SELECT created, gps_latitude, gps_longitude, metal_detection_signal
FROM drone_data'
        )
        data = cursor.fetchall()
        return data
    except Exception as e:
        current_app.logger.error(f'Error: {e}')
        return []
finally:
    if conn.is_connected():
        conn.close()

```

## ДОДАТОК В

## Частини програмного коду для навчання нейронної мережі yolov8

```
import os
import sys
from pathlib import Path
import warnings

# Suppress warnings
warnings.filterwarnings('ignore')

def create_local_dataset_config():
    """Create dataset configuration for local training"""
    dataset_config = """
# Landmine Detection Dataset Configuration for Local Training
path: data # dataset root dir (relative to current directory)
train: train/images # train images (relative to 'path')
val: val/images # val images (relative to 'path')
test: test/images # test images (optional)

# Classes
names:
  0: mine

# Number of classes
nc: 1
"""

    config_path = 'landmine_dataset_local.yaml'
    with open(config_path, 'w') as f:
```

```

        f.write(dataset_config)
    return config_path

def verify_dataset():
    """Verify dataset structure and count files"""
    data_root = Path('data')

    if not data_root.exists():
        return False

    splits = ['train', 'val']
    total_images = 0
    total_labels = 0

    for split in splits:
        images_dir = data_root / split / 'images'
        labels_dir = data_root / split / 'labels'

        if images_dir.exists():
            images = list(images_dir.glob('*.*jpg')) + list(
                images_dir.glob('*.*png'))
            total_images += len(images)
        else:
            return False

        if labels_dir.exists():
            labels = list(labels_dir.glob('*.*txt'))
            total_labels += len(labels)
        else:

```

```

    return False

print(f'Total images: {total_images}, labels: {total_labels}')

if total_images == 0:
    print("No training images found!")
    return False

return True

def train_landmine_detector():
    """Train YOLOv8 model from scratch on landmine dataset"""

    # Verify dataset
    if not verify_dataset():
        print("Dataset verification failed!")
        return None

    # Create local dataset config
    dataset_config = create_local_dataset_config()

    # Create fresh model
    print("Creating YOLOv8n model from scratch...")
    model = YOLO("yolov8n.yaml") # Create from scratch, no pretrained weights
    print("Model created")

    # Device detection
    if torch.backends.mps.is_available():
        device = 'mps'

```

```

    print("Using Apple Silicon MPS acceleration")
elif torch.cuda.is_available():
    device = 'cuda'
    print("Using CUDA GPU acceleration")
else:
    device = 'cpu'
    print("Using CPU")

# Training parameters optimized for landmine detection
train_params = {
    'data': dataset_config,
    'epochs': 50, # Sufficient epochs for good training
    'imgsz': 640,
    'batch': 8 if device != 'cpu' else 4,
    'workers': 4 if device != 'cpu' else 2,
    'device': device,
    'project': 'landmine_training_runs',
    'name': 'landmine_detector_v1',
    'exist_ok': True,
    'verbose': True,
    'save': True,
    'plots': True,
    'val': True,
    'patience': 20, # Early stopping patience
    'save_period': 10, # Save every 10 epochs

    # Learning parameters
    'lr0': 0.01,
    'lrf': 0.01,
    'momentum': 0.937,

```

```
'weight_decay': 0.0005,  
'warmup_epochs': 3.0,  
'warmup_momentum': 0.8,  
'warmup_bias_lr': 0.1,  
  
# Loss function weights  
'box': 7.5,  
'cls': 0.5,  
'dfl': 1.5,  
  
# Data augmentation for aerial imagery  
'hsv_h': 0.015,  
'hsv_s': 0.7,  
'hsv_v': 0.4,  
'degrees': 0.0, # No rotation for aerial images  
'translate': 0.1,  
'scale': 0.5,  
'shear': 0.0,  
'perspective': 0.0,  
'flipud': 0.0,  
'fliplr': 0.5, # Horizontal flip is ok for landmines  
'mosaic': 1.0,  
'mixup': 0.0,  
'copy_paste': 0.0,  
  
# Optimization  
'optimizer': 'SGD',  
'close_mosaic': 10,  
'amp': False, # Disable mixed precision for stability  
'single_cls': True, # Single class detection
```



```

'rect': False,
'cos_lr': True,
'seed': 42,
}

for key, value in train_params.items():
    print(f' {key}: {value}')

print(f'\nStarting training for {train_params['epochs']} epochs...')

try:
    # Start training
    results = model.train(**train_params)
    print("\nTraining completed successfully!")

    # Check for saved models
    model_dir = Path("landmine_training_runs/landmine_detector_v1/weights")
    if model_dir.exists():
        best_model = model_dir / "best.pt"
        last_model = model_dir / "last.pt"

        if best_model.exists():
            print(f'Best model saved: {best_model}')
        if last_model.exists():
            print(f'Last model saved: {last_model}')

        return best_model if best_model.exists() else last_model

    return None

```

```
except Exception as e:
    error_msg = str(e)
    print(f"\nTraining error: {error_msg}")

    # Check if model was saved despite error
    model_dir = Path("landmine_training_runs/landmine_detector_v1/weights")
    if model_dir.exists():
        models = list(model_dir.glob("*.pt"))
        if models:
            print(f"Found saved models despite error: {models}")
            return models[0]

    raise e
```

## ДОДАТОК Г

Частини програмного коду для використання нейронної мережі yolov8

```
import os
import cv2
import numpy as np
from pathlib import Path
from ultralytics import YOLO
import time

class VideoLandmineDetector:

    def __init__(
        self,
        model_path="landmine_training_runs/landmine_detector_v1/weights/best.pt"
    ):
        """Initialize the video landmine detector"""

        if not os.path.exists(model_path):
            raise FileNotFoundError(f"Trained model not found: {model_path}")

        print(f"Loading trained model: {model_path}")
        self.model = YOLO(model_path)
        self.detection_count = 0
        self.total_frames = 0

    def process_video(self,
        input_video_path,
        output_video_path=None,
        conf_threshold=0.3,
```

```

        save_frames=False):
    """
    Process video file and create annotated output

    Args:
        input_video_path: Path to input video file
        output_video_path: Path for output annotated video (optional)
        conf_threshold: Confidence threshold for detections
        save_frames: Whether to save individual annotated frames
    """

    if not os.path.exists(input_video_path):
        print(f"Video file not found: {input_video_path}")
        return

    # Generate output video path if not provided
    if output_video_path is None:
        video_name = Path(input_video_path).stem
        output_video_path = f"ANNOTATED_{video_name}.mp4"

    print(f"\n Processing video: {input_video_path}")
    print(f"Output will be saved as: {output_video_path}")
    print(f"Confidence threshold: {conf_threshold}")

    # Open input video
    cap = cv2.VideoCapture(input_video_path)

    if not cap.isOpened():
        print(f"Cannot open video file: {input_video_path}")
        return

```

```

# Get video properties
fps = int(cap.get(cv2.CAP_PROP_FPS))
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
total_frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
duration = total_frames / fps

# Setup video writer
fourcc = cv2.VideoWriter_fourcc(*'mp4v')
out = cv2.VideoWriter(output_video_path, fourcc, fps, (width, height))

# Create frames directory if saving frames
frames_dir = None
if save_frames:
    frames_dir = Path(
        f'annotated_frames_{Path(input_video_path).stem}')
    frames_dir.mkdir(exist_ok=True)
    print(f'Saving frames to: {frames_dir}')

# Process video frame by frame
frame_number = 0
total_detections = 0
detection_log = []

print(f'\nStarting video processing...')
start_time = time.time()

while True:
    ret, frame = cap.read()

```

```

if not ret:
    break

frame_number += 1

# Run landmine detection on frame
results = self.model(frame, conf=conf_threshold, verbose=False)

frame_detections = 0
detections_info = []

# Process detection results
for result in results:
    if result.bboxes is not None:
        frame_detections = len(result.bboxes)

        for i, box in enumerate(result.bboxes):
            # Get bounding box coordinates
            x1, y1, x2, y2 = box.xyxy[0].cpu().numpy().astype(int)
            conf = box.conf[0].cpu().numpy()

            # Draw red bounding box
            cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 255),
                           3)

            # Create label with timestamp
            timestamp = frame_number / fps
            label = f"LANDMINE #{i+1}: {conf:.1%}"
            time_label = f"Time: {timestamp:.1f}s"

```

```

# Draw label background
(label_w, label_h), _ = cv2.getTextSize(
    label, cv2.FONT_HERSHEY_SIMPLEX, 0.7, 2)
(time_w,
time_h), _ = cv2.getTextSize(time_label,
                             cv2.FONT_HERSHEY_SIMPLEX,
                             0.5, 1)

# Main label
cv2.rectangle(frame, (x1, y1 - label_h - 15),
              (x1 + label_w, y1), (0, 0, 255), -1)
cv2.putText(frame, label, (x1, y1 - 5),
            cv2.FONT_HERSHEY_SIMPLEX, 0.7,
            (255, 255, 255), 2)

# Timestamp label
cv2.rectangle(frame, (x1, y2),
              (x1 + time_w, y2 + time_h + 10),
              (0, 0, 255), -1)
cv2.putText(frame, time_label, (x1, y2 + time_h + 5),
            cv2.FONT_HERSHEY_SIMPLEX, 0.5,
            (255, 255, 255), 1)

# Store detection info
detections_info.append({
    'frame': frame_number,
    'time': timestamp,
    'confidence': conf,
    'bbox': [x1, y1, x2, y2]
})

```

```

# Add frame info overlay
frame_info = f'Frame: {frame_number}/{total_frames} | Detections:
{frame_detections}'

cv2.putText(frame, frame_info, (10, 30),
cv2.FONT_HERSHEY_SIMPLEX,
0.8, (255, 255, 255), 2)

cv2.putText(frame, frame_info, (10, 30),
cv2.FONT_HERSHEY_SIMPLEX,
0.8, (0, 0, 0), 1)

# Write frame to output video
out.write(frame)

# Save individual frame if requested
if save_frames and frame_detections > 0:
    frame_filename = frames_dir /
f'frame_{frame_number:06d}_detections_{frame_detections}.jpg'
    cv2.imwrite(str(frame_filename), frame)

# Log detections
if frame_detections > 0:
    total_detections += frame_detections
    detection_log.extend(detections_info)
    timestamp = frame_number / fps
    print(
        f' Frame {frame_number} ({timestamp:.1f}s): {frame_detections}
landmines detected"
    )

```



```

# Progress update every 100 frames
if frame_number % 100 == 0:
    progress = (frame_number / total_frames) * 100
    elapsed = time.time() - start_time
    eta = (elapsed / frame_number) * (total_frames - frame_number)
    print(
        f" Progress: {progress:.1f}% | ETA: {eta:.1f}s | Detections so far:
{total_detections}"
    )

# Cleanup
cap.release()
out.release()
cv2.destroyAllWindows()

# Processing complete
processing_time = time.time() - start_time

# Save detection report
self._save_detection_report(detection_log, input_video_path,
                             output_video_path)

return detection_log

```

## ДОДАТОК І

### Акт впровадження

«ЗАТВЕРДЖУЮ»  
Директор ТОВ «Хладотехника»  
А. І. Бузник

« 19 » грудня 2025 р.

АКТ  
про впровадження результатів дисертаційної роботи  
Роботько Сергія Павловича  
«Багаторівнева автоматизована система керування роботизованою  
платформою для задач просторового моніторингу на основі  
комп'ютерного зору»

Цей акт підтверджує, що результати дисертаційної роботи аспіранта кафедри Комп'ютеризованих систем управління Миколаївського Національного університету кораблебудування імені адмірала Макарова Роботько С.П. впроваджено у виробничу діяльність ТОВ «Хладотехника», Україна, Миколаїв, 54005, вул. Артилерійська, 19/2, тел. +38 (0512) 470969.

Впровадження здійснено в межах підготовчих робіт перед встановленням промислових холодильних установок на об'єктах замовників. Компанія застосувала напрацювання дисертації як регламентований етап попереднього моніторингу території об'єкта перед початком робіт, що включає дистанційний огляд майданчика роботизованою платформою та формування матеріалів для інженерної оцінки і планування підготовки ділянки.

У процесі впровадження використано підхід, що передбачає виконання просторового моніторингу з отриманням оптичних даних, їх прив'язкою до координат місцевості, фіксацією виявлених аномалій та потенційно небезпечних об'єктів з подальшим формуванням звіту для відповідальних осіб. Застосування зазначених результатів дозволило на етапі підготовки майданчика завчасно виявляти об'єкти та фактори ризику, які можуть становити загрозу для робітників і техніки під час під'їзду, розвантаження, переміщення обладнання та виконання земляних або монтажних робіт. До таких факторів віднесено підозрілі предмети на поверхні, металеві залишки та уламки, небезпечні перешкоди, локальні зони з обмеженою прохідністю, а також ділянки, що потребують додаткової перевірки перед допуском персоналу і механізмів.

За результатами застосування напрацьовань дисертаційної роботи ТОВ «Хладотехника» отримало можливість підвищити рівень промислової безпеки на підготовчому етапі, скоротити час первинного огляду території та підвищити обґрунтованість рішень щодо організації робіт на майданчику. Підхід використовується як допоміжний інструмент інженерної підготовки об'єкта та планування безпечного виконання робіт із встановлення промислових холодильних установок.

Цей акт не є підставою для фінансових розрахунків.

Директор ТОВ «Хладотехника»



А. І. Бузник

# ДОДАТОК Д

## Акт впровадження



«ЗАТВЕРДЖУЮ»

Ректор

Національного університету

кораблебудування ім. адм. Макарова

проф. Є.І. Трушляков

2026 р.

### АКТ

впровадження результатів дисертаційної роботи  
Роботька Сергія Павловича в області автоматизації мобільних  
роботизованих платформ та систем комп'ютерного зору

Комісія у складі:

д.т.н., професора Павлова Г.В. – проректора з наукової роботи (НР), к.т.н.,  
доцента О.С. Герасіна – доцента кафедри комп'ютеризованих систем управління  
(КСУ), Савенкова О.В. – начальника науково-дослідної частини (НДЧ) НУК,  
підписала дійсний акт про те, що основні теоретичні та практичні результати  
наукових досліджень здобувача кафедри КСУ Роботька С.П., проведених в рамках  
дисертації на здобуття наукового ступеня доктора філософії, а саме:

- програмно-алгоритмічні засоби побудови систем комп'ютерного зору та семантичної верифікації для задач автономного просторового моніторингу;
- алгоритми ідентифікації та класифікації об'єктів (зокрема, сторонніх предметів) на складних фонах на основі комплексування даних (sensor fusion) візуального каналу та індукційних сенсорів;
- методи організації відмовостійких багаторівневих систем керування роботизованими платформами та передачі телеметрії у режимі реального часу;

впроваджені в Національному університеті кораблебудування ім. адм. Макарова при виконанні (в рамках наукових досліджень на замовлення Міністерства освіти і науки України) держбюджетної теми № 2398 "Новітні системи дистанційного керування мобільними роботами для відновлення суден і портової інфраструктури", номер державної реєстрації № 0126U001721 (2026-2028 рр.).

Проректор з НР, д.т.н., професор

Г.В. Павлов

Доцент кафедри КСУ, к.т.н., доцент

О.С. Герасін

Начальник науково-дослідної частини НУК

О.І. Савенков



## ДОДАТОК Е

### Акт впровадження

#### МАЛЕ ПІДПРИЄМСТВО «ОРБІТА»

Україна, 54028, Миколаївська обл., місто Миколаїв, вул. Кірова, буд. 238, офіс 121, [elmerbud.ua@gmail.com](mailto:elmerbud.ua@gmail.com)

#### АКТ ВПРОВАДЖЕННЯ

результатів дисертаційної роботи Роботька Сергія Павловича на тему «Багаторівнева автоматизована система керування роботизованою платформою для задач просторового моніторингу на основі комп'ютерного зору», поданої на здобуття наукового ступеня доктора філософії за спеціальністю 151 – Автоматизація та комп'ютерно-інтегровані технології.

Цим актом засвідчується, що теоретичні та практичні результати дисертаційної роботи Роботька Сергія Павловича були використані у науково-виробничому процесі МП «ОРБІТА» при розробці та модернізації систем дистанційного моніторингу територій і об'єктів інфраструктури за допомогою мобільних робототехнічних комплексів (БПЛА).

Зокрема, у практичну діяльність підприємства впроваджено наступні результати досліджень:

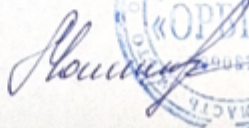
1. **Структуру багаторівневої системи автоматичного керування роботизованою платформою**, що дозволило оптимізувати процеси виконання автономних місій просторового моніторингу зі збереженням стійкості до зовнішніх збурень.
2. **Асинхронну конвеєрну архітектуру обробки візуальних даних (на базі алгоритмів YOLO)**, яка була застосована як візуальна підсистема для виявлення та класифікації цільових об'єктів на земній поверхні в умовах складного фону.
3. **Методи та алгоритми комплексування сенсорних даних (sensor fusion)**, які забезпечили ефективне об'єднання даних оптичних каналів сприйняття з іншими бортовими датчиками, що суттєво підвищило надійність розпізнавання об'єктів під час обстеження цільових територій.

Очікується, що подальше застосування розроблених Роботьком С.П. моделей та алгоритмів для автоматизації процесів дистанційного керування роботизованими платформами дозволить значно підвищити точність виявлення об'єктів, енергоефективність виконання місій та загальну продуктивність під час проведення моніторингових, відновлювальних або пошукових робіт, у тому числі на об'єктах судноремонтної та портової інфраструктури.

Запропоновані рішення є доцільними для використання у подальших науково-дослідних та дослідно-конструкторських роботах підприємства у галузі робототехніки та інформаційних технологій.

З повагою,

технічний директор МП «Орбіта»



Малашевський О.М.





# ДОДАТОК Є

## Акт впровадження



«ЗАТВЕРДЖУЮ»

Ректор

Національного університету

корабелобудування ім. адм. Макарова

проф. Є.І. Трушляков

2026 р.

### АКТ

впровадження результатів дисертаційної роботи на здобуття наукового ступеня доктора філософії С.П. Роботько у навчальний процес

Комісія у складі:

д.т.н., доцента Жука Д.О. – директора Навчально-наукового інституту автоматики та електротехніки (ННІАЕ), д.т.н., професора Черно О.О. – завідуючого кафедрою комп'ютеризованих систем управління (КСУ), к.т.н., доцента Топалова А.М. – доцента кафедри КСУ, підписала дійсний акт про те, що основні теоретичні положення наукових досліджень аспіранта кафедри комп'ютеризованих систем управління Роботька С.П., проведених у рамках дисертаційної роботи, впроваджені та використовуються в навчальному процесі НУК ім. адм. Макарова, а саме:

- математичні моделі просторового руху та функціональні схеми багаторівневих систем автоматичного керування мобільними роботизованими платформами (БПЛА) – при викладанні дисциплін «Мікропроцесорні технології та системи», «Теорія проектування цифрових систем»;
- програмно-алгоритмічні засоби обробки візуальних даних (на базі архітектур YOLO), методи комплексування сенсорних даних (sensor fusion) та семантичної верифікації – при викладанні дисциплін «Теорія систем та системний аналіз», «Програмні засоби систем управління».

Результати дисертаційної роботи використовуються, зокрема, в лекційному матеріалі та лабораторних роботах, а також для дипломного проектування при підготовці фахівців (бакалаврів та магістрів) за спеціальністю 151 – «Автоматизація та комп'ютерно-інтегровані технології».

Директор ННІАЕ,  
д.т.н., доцент

Д.О. Жук

Завідуючий кафедрою КСУ,  
д.т.н., професор

О.О. Черно

Доцент кафедри КСУ,  
к.т.н., доцент

Топалов А.М.

ДОДАТОК Ж  
Патент



УКРАЇНА

(19) **UA** (11) **157368** (13) **U**  
(51) МПК (2024.01)  
**G01B 7/06** (2006.01)  
**H01F 13/00**

НАЦІОНАЛЬНИЙ ОРГАН  
ІНТЕЛЕКТУАЛЬНОЇ ВЛАСНОСТІ  
ДЕРЖАВНА ОРГАНІЗАЦІЯ  
"УКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ  
ОФІС ІНТЕЛЕКТУАЛЬНОЇ  
ВЛАСНОСТІ ТА ІННОВАЦІЙ"

(12) ОПИС ДО ПАТЕНТУ НА КОРИСНУ МОДЕЛЬ

|                                                                                 |                                                                                                                                                                                                                                                                                                                                       |
|---------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (21) Номер заявки: <b>u 2024 00179</b>                                          | (72) Винахідник(и):<br><b>Тростинський Михайло Миколайович (UA),<br/>Топалов Андрій Миколайович (UA),<br/>Карпеченко Антон Анатолійович (UA),<br/>Герасін Олександр Сергійович (UA),<br/>Бобров Максим Миколайович (UA),<br/>Роботько Сергій Павлович (UA),<br/>Хо́да Валентин Михайлович (UA),<br/>Недо Антон Олександрович (UA)</b> |
| (22) Дата подання заявки: <b>11.01.2024</b>                                     |                                                                                                                                                                                                                                                                                                                                       |
| (24) Дата, з якої є чинними права інтелектуальної власності: <b>10.10.2024</b>  |                                                                                                                                                                                                                                                                                                                                       |
| (46) Публікація відомостей про державну реєстрацію: <b>09.10.2024, Бюл.№ 41</b> | (73) Володілець (володільці):<br><b>НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ<br/>КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА<br/>МАКАРОВА,<br/>просп. Героїв України, 9, м. Миколаїв, 54025 (UA)</b>                                                                                                                                                              |

(54) ПРИСТРІЙ ДЛЯ ВИМІРЮВАННЯ ТОВЩИНИ СТРУМОПРОВІДНОЇ МЕТАЛЕВОЇ ПЛІВКИ В ПРОЦЕСІ НАПИЛЮВАННЯ У ВАКУУМІ

(57) Реферат:

Пристрій для вимірювання товщини струмопровідної металевої плівки в процесі напилювання у вакуумі містить котушку та блок обробки вихідного сигналу, що складається з генератора синусоїдального сигналу, підсилювача, аналого-цифрового перетворювача та комп'ютера. Причому використано котушку, ізоляція обмотки якої витримує температуру до 300 °С, яка конструктивно з'єднана з вимірювальним склом і розташована у вакуумній камері. Клеми котушки підключені до блока обробки вихідного сигналу, який додатково містить блок випрямлення сигналів та блок різниці.

UA 157368 U