

Кваліфікаційна наукова  
праця на правах рукопису

МЕЛЬНИЧЕНКО ОЛЕКСАНДР ВІКТОРОВИЧ

УДК: 004.932.2

ДИСЕРТАЦІЯ

МЕТОДИ ЗБОРУ, РОЗПІЗНАВАННЯ ТА ОБРОБКИ ЗОБРАЖЕНЬ,  
ОТРИМАНИХ ІЗ ВИКОРИСТАННЯМ БПЛА, ДЛЯ ВИЯВЛЕННЯ ЗАДАНИХ  
ОБ'ЄКТІВ

122 Комп'ютерні науки  
(шифр і назва спеціальності)

12 Інформаційні технології  
(галузь знань)

Подається на здобуття наукового ступеня доктора філософії

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

(підпис)



О. В. Мельниченко

Науковий керівник: Савенко Олег Станіславович, доктор технічних наук, професор

## АНОТАЦІЯ

*Мельниченко Олександр Вікторович.* Методи збору, розпізнавання та обробки зображень, отриманих із використанням БПЛА, для виявлення заданих об'єктів. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії з галузі знань 12 Інформаційні технології за спеціальністю 122 Комп'ютерні науки. – Хмельницький національний університет, Хмельницький. 2023.

Вирішення задачі покращення ефективності збору, точності розпізнавання та обробки зображень, отриманих із використанням безпілотних літальних апаратів (БПЛА), для виявлення заданих об'єктів, є однією із важливих наукових задач в сфері інформаційних технологій (ІТ), орієнтованих на розробку систем збору зображень групами БПЛА та подальше розпізнавання об'єктів на цих зображеннях і обробку результатів розпізнавання.

У дисертації здійснено аналіз засобів та технологій збору зображень з використанням БПЛА, існуючих систем аналогічного призначення, методів та засобів обробки зображень об'єктів та їх розпізнавання. В роботі розроблено методи збору, розпізнавання та обробки зображень отриманих з використанням БПЛА для виявлення заданих об'єктів, які покращують ефективність збору, точність розпізнавання та обробки зображень отриманих з використанням БПЛА, а також розроблено відповідні засоби, здійснено постановку експериментів і проведено з розробленими засобами експериментальні дослідження.

Об'єктом дослідження є процеси збору, розпізнавання та обробки зображень, отриманих із використанням БПЛА, для виявлення заданих об'єктів.

Предметом дослідження є методи і алгоритми забезпечення збору, розпізнавання та обробки зображень, отриманих із використанням БПЛА, для виявлення заданих об'єктів.

Метою дисертаційного дослідження є покращення ефективності збору, точності розпізнавання та обробки зображень, отриманих із використанням БПЛА, для виявлення заданих об'єктів.

Наукова новизна одержаних результатів полягає в наступному:

1) розроблено новий метод побудови маршрутів БПЛА згідно технологій самонавчання, який на відміну від відомих, полягає в позиціюванні апаратних пристроїв у тривимірному координатному просторі в реальному часі за рахунок самонавчання при побудові маршрутів їх руху, що дає змогу покращити переміщення та синхронізацію між групою БПЛА або одного БПЛА в межах робочого сегменту і за рахунок цього збільшення кількості опрацьованих даних.

2) розроблено новий метод динамічного отримання зображень заданих структурних об'єктів в тривимірному просторі за допомогою декількох БПЛА, який на відміну від відомих відрізняється тим, що забезпечує в процесі активації кожного БПЛА генерацію відеоряду в центральну систему, прийняття рішень про подальшу роботу групи БПЛА, підтвердження цілісності програмної місії та визначення рівня критичності для продовження виконання роботи групи БПЛА за рахунок визначення станів модулів та комплексному врахуванні вимог розподілення, багаторівневості та автоматизованості, що покращує узгодження між різними БПЛА та досягнення переміщення всієї групи БПЛА від заданих початкових до кінцевих точок програмної місії автономно.

3) розроблено новий метод синхронізації відеопотоків в режимі реального часу, який відрізняється від відомих тим, що забезпечує накопичення даних про попередні програмні місії БПЛА, у випадку критичних збоїв, які спотворюють цілісність структур даних із джерел отримання відеопотоків, зберігає такі структури в спеціальному журналі помилок та не відправляє їх як вхідні параметри в наступну обробку, що дає змогу виконувати порівняння отриманих поточних результатів із минулими в режимі реального часу і це забезпечує оперативне отримання результатів та здійснення виявлення структурних об'єктів, які були пропущені в процесі минулих програмних місій.

4) вдосконалено метод виявлення заданих структурних об'єктів на зображеннях, який на відміну від оригінальної архітектури YOLOv5, полягає в тому, що модифіковано модуль фокусування нейронної мережі, видалено згортковий шар поєднуючий вхідну карту ознак з операцією конкатенції,

актуалізовано механізм візуальної уваги для вилучення ознак, об'єднання шарів 4 і 15, 6 і 11, 10 і 21 оригінальної архітектури замінено на об'єднання шарів 5 і 18, 8 і 14, 13 і 24 вдосконаленої архітектури, вихідні карти ознак 14-го та 21-го шарів вдосконаленої архітектури об'єднано між собою, що дало змогу покращити точність виявлення і зменшити час навчання нейронної мережі.

Практичне значення отриманих результатів. За результатами виконаних досліджень здобувачем розроблено методи, спосіб, алгоритми та засоби покращення ефективності збору, точності розпізнавання та обробки зображень, отриманих із використанням БПЛА, для виявлення заданих об'єктів. Це дало змогу створювати автоматизовані системи з використанням групи БПЛА для збору, отримання та розпізнавання заданих об'єктів на отриманих зображеннях. Заданими об'єктами вибрано яблука на деревах в фруктовому саду. Розроблена автоматизована система для виявлення та обчислення кількості яблук у фруктовому саду в режимі реального часу. Перевагою розробленої системи над аналогами є отримання нею множини відеокадрів у режимі реального часу з камер кількох БПЛА та синхронізація цих відеокадрів між собою в одну інформаційну структуру даних, що надалі трансформується в суцільне зображення. Крім того, використання функцій оптимізації якості зображення дає змогу максимально ефективно виявляти задані структурні об'єкти під час виконання робочих місій БПЛА в робочому середовищі. Використання такої системи дало змогу отримувати суцільний потік даних до всіх наступних програмних компонентів автоматизованої системи. Так, оцінка синхронізації відеопотоків за індексом SSIM коливається від 0,79 до 0,92, із середнім значенням 0,87, а за індексом PSNR – від 22 до 39, що свідчить про покращення ефективності роботи розробленої системи із відеопотоками та належну якість отриманих об'єднаних зображень. У результаті проведених експериментальних досліджень було доведено покращення ефективності роботи розробленої автоматизованої системи та точності виявлення об'єктів на зображеннях, що підтверджується високим середнім значенням у 82,69% показника достовірності виявлення та обчислення кількості фруктових плодів та низьким середнім рівнем помилок I (14,67%) та II (18,33%) роду.

Теоретичні та практичні результати дослідження впроваджені в ТОВ «ЮКС++» (м. Хмельницький), Державному підприємстві «Новатор» (м. Хмельницький), ПП «НОЛТ ТЕХНОЛОДЖИС» (м. Хмельницький), ТОВ «Агротех сервіс» (с. Богданівці), а також, в освітньому процесі Хмельницького національного університету при викладанні дисциплін на кафедрі комп'ютерної інженерії та інформаційних систем для спеціальності 126 Інформаційні системи та технології, 123 Комп'ютерна інженерія та кафедрі комп'ютерних наук для спеціальності 122 Комп'ютерні науки, зокрема в курсах «Методи та системи штучного інтелекту», «Комп'ютерні та кіберфізичні системи», «Методи, засоби та алгоритми в задачах обчислювального інтелекту та комп'ютерного зору», «Теорія, проектування та моделювання спеціалізованих комп'ютерних систем» та «Технології проектування інформаційних систем».

У вступі представлено обґрунтування актуальності наукової задачі із забезпечення покращення ефективності збору, точності розпізнавання та обробки зображень, отриманих із використанням БПЛА, для виявлення заданих об'єктів. Також, представлено зв'язок тематики дослідження з напрямками наукових досліджень відомих дослідників цієї проблеми в світі та відображено основні наукові результати роботи та її практичне значення.

У першому розділі здійснено аналіз предметної області дослідження, відомих методів збору, розпізнавання та обробки зображень отриманих з використанням БПЛА для виявлення заданих об'єктів, зокрема актуальності їх застосування до польових задач садівництва, розпізнавання фруктових плодів в реальних умовах фруктового саду, а також, методи виявлення та відстеження заданих структурних об'єктів за відеопотоком, методи ідентифікації та обчислення кількості заданих структурних об'єктів, огляд технічних пристроїв та апаратного забезпечення для використання безпілотних літальних апаратів, керування робочими місіями групи безпілотних літальних апаратів, використання групи безпілотних літальних апаратів для генерації фото- та відеоматеріалів. Також, підведено підсумки проведеного аналізу та здійснено постановку задачі дослідження.

У другому розділі представлено розробку методу побудови маршрутів БПЛА на основі технологій самонавчання, архітектуру автоматизованої системи динамічного отримання зображень заданих об'єктів в тривимірному просторі, проектування робочої місії БПЛА в тривимірному просторі, інформаційні потоки автоматизованої системи під час виконання програмної місії, процес генерації зображень заданих структурних об'єктів в тривимірному просторі, створення програмної місії в системі, програмний інтерфейс автоматизованої системи та розробку методу динамічного отримання зображень заданих структурних об'єктів в тривимірному просторі за допомогою декількох БПЛА, а також підведено підсумки з отриманих результатів.

У третьому розділі представлено розроблений метод синхронізації відеопотоків в режимі реального часу, вдосконалений метод виявлення заданих структурних об'єктів, спосіб обчислення кількості заданих структурних об'єктів, створення детектора у задачі розпізнавання образів у режимі реального часу, постановку експерименту, підготовку робочого середовища, підготовку навчальних наборів даних, створення детектора, тестування та результати створеного детектора, а також підведено підсумки з отриманих результатів.

У четвертому розділі представлено програмну реалізацію автоматизованої системи виявлення та обчислення кількості заданих структурних об'єктів, опис експериментальної установки, постановку і проведення експериментальних досліджень із застосування розробленої автоматизованої системи, опис експериментального середовища, оцінювання ефективності автоматизованої системи, а також підведено підсумки з отриманих результатів.

У висновках подано отримані наукові та практичні результати дослідження.

У додатках подано наукові публікації, в яких відображено основні наукові результати роботи, відомості про апробацію результатів дисертації, акти про впровадження результатів роботи, аналіз комерційних платформ цифрового садівництва та лістинг програмного забезпечення.

Ключові слова: маршрут, самонавчання, БПЛА, відеопотік, розпізнавання образів, штучна нейронна мережа, автоматизована система.

## ANNOTATION

*Oleksandr Viktorovych Melnychenko*. Methods of collecting, recognizing and processing images obtained using UAVs to detect given objects. – Qualifying scientific work on manuscript rights.

Dissertation for obtaining the scientific degree of Doctor of Philosophy in the field of knowledge 12 Information technologies in the specialty 122 Computer science. – Khmelnytskyi National University, Khmelnytskyi. 2023.

Solving the task of improving the efficiency of collection, recognition accuracy and processing of images obtained using UAVs to detect given objects is one of the important scientific tasks in the field of information technology (IT), focused on the development of image collection systems by groups of UAVs and further recognition of objects on these images and processing the recognition results.

The dissertation analyzed means and technologies of image collection using UAVs, existing systems of a similar purpose, methods and means of image processing of objects and their recognition. In the work, methods of collecting, recognizing and processing images obtained using UAVs for the detection of specified objects have been developed, which improve the efficiency of collection, accuracy of recognition and processing of images obtained using UAVs, as well as appropriate tools have been developed, experiments have been set up and carried out with the developed tools experimental studies.

The object of the study is the process of collecting, recognizing and processing images obtained using UAVs to detect given objects.

The subject of the study is methods and algorithms for ensuring the collection, recognition and processing of images obtained using UAVs for the detection of given objects.

The aim of the dissertation research is to improve the collection efficiency, accuracy of recognition and processing of images obtained with the use of UAVs for the detection of given objects.

The scientific novelty of the obtained results is as follows:

1) a new method of constructing UAVs routes according to self-learning technologies has been developed, which, unlike the known ones, consists in positioning hardware devices in three-dimensional coordinate space in real time due to self-learning when constructing their movement routes, which makes it possible to improve movement and synchronization between a group of drones or one drone within the working segment and due to this increase in the amount of processed data;

2) a new method of dynamically obtaining images of given structural objects in three-dimensional space using several drones has been developed, which, unlike the known ones, differs in that it provides, in the process of activating each drones, the generation of a video sequence to the central system, making decisions about the further work of the UAVs group, confirming the integrity of the program mission and determining the level of criticality for the continuation of the work of the drones group by determining the states of the modules and comprehensively taking into account the requirements of distribution, multi-level and automation, which improves the coordination between different drones and achieving the movement of the entire group of drones from the given starting points to the end points of the program mission autonomously;

3) a new method of synchronizing video streams in real time has been developed, which differs from the known ones in that it ensures the accumulation of data on previous drones program missions, in case of critical failures that distort the integrity of data structures from the sources of receiving video streams, saves such structures in a special error log and does not send them as input parameters to the subsequent processing, which makes it possible to perform a comparison of the received current results with the past ones in real time, and this ensures the prompt obtaining of results and the detection of structural objects that were missed in the process of past program missions;

4) the method of detecting given structural objects in images has been improved, which, unlike the original YOLOv5 architecture, consists in the fact that the neural network focusing module has been modified, the convolutional layer combining the input feature map with the concatenation operation has been removed, the visual attention mechanism for feature extraction has been updated, the union of layers 4 and 15, 6 and



11, 10 and 21 of the original architecture is replaced by the union of layers 5 and 18, 8 and 14, 13 and 24 of the improved architecture, the output feature maps of the 14th and 21st layers of the improved architectures are combined, which made it possible to improve the accuracy of detection and reduce the training time of the neural network.

The practical significance of the results obtained. The developed automated system is designed to detect and count the number of apples in an orchard in real time. The advantage of the developed system over analogues is that it receives multiple video frames in real time from the cameras of several UAVs and synchronizes these video frames with each other into one informational data structure, which will later be transformed into a continuous image. In addition, the use of image quality optimization functions allows for the most efficient detection of structural during UAV operational missions in the operational environment. The use of such a transformation tool enabled the system to receive a continuous flow of data to all subsequent software components of the automated system. Thus, the evaluation of the synchronization of video streams according to the SSIM index varies from 0.79 to 0.92, with an average value of 0.87, and according to the PSNR index – from 22 to 39, which indicates the high efficiency of the developed system with video streams and the good quality of the received information. combined images.

As a result of the experimental studies, the effectiveness of the developed automated system was proven, which is confirmed by a high average value of 82.69% of the reliability indicator of detecting and calculating the number of fruit fruits and a low average level of errors I (14.67%) and II (18.33%) genus.

The theoretical and practical results of the research were implemented in LLC “UKS++” (Khmelnyskyi), SE “NOVATOR” (Khmelnyskyi), PE “NOLT TECHNOLOGY” (Khmelnyskyi), LLC “AGROTECHSERVIS” (Bohdanivtsi), as well as, in the educational process of Khmelnyskyi National University when teaching disciplines at the Department of Computer Engineering and Information Systems for the specialty 126 Information Systems and Technologies, 123 Computer Engineering and the Department of Computer Sciences for the specialty 122 Computer Science, in particular in the courses “Methods and systems of artificial intelligence”, “Computer and

cyberphysical systems”, “Methods, tools and algorithms in tasks of computational intelligence and computer vision”, “Theory, design and modeling of specialized computer systems” and “Technologies of information systems design”.

The introduction presents the justification of the relevance of the scientific task of ensuring the improvement of the efficiency of collection, recognition accuracy and processing of images obtained using UAVs for the detection of given objects. Also, the connection of the research topic with the directions of scientific research of famous researchers of this problem in the world is presented and the main scientific results of the work and its practical significance.

In the first section, an analysis of the subject area of the study, known methods of collecting, recognizing and processing images obtained using UAVs for detecting given objects, in particular, the relevance of their application to field tasks of horticulture, recognition of fruit fruits in real conditions of an orchard, as well as methods detection and tracking of specified structural objects by video stream, methods of identification and calculation of the number of specified structural objects, overview of technical devices and hardware for the use of unmanned aerial vehicles, management of operational missions of a group of unmanned aerial vehicles, use of a group of unmanned aerial vehicles for photo generation – and video material. Also, the results of the conducted analysis are summarized and the research task is formulated.

The second chapter presents the development of a method for constructing UAV routes based on self-learning technologies, the architecture of an automated system for dynamic acquisition of images of specified objects in three-dimensional space, the design of a UAV operational mission in three-dimensional space, information flows of an automated system during the execution of a program mission, the process of generating images of specified structural objects in three-dimensional space, creation of a program mission in the system, program interface of the automated system and development of a method of dynamic acquisition of images of given structural objects in three-dimensional space with the help of several UAVs, as well as summing up the results.

The third section presents the developed method of synchronizing video streams in real time, the developed method of detecting given structural objects, the method of

calculating the number of given structural objects, creating a detector in the task of pattern recognition in real time, setting up the experiment, preparing the working environment, preparing training data sets, creation of the detector, testing and results of the created detector, as well as summaries of the obtained results.

The fourth chapter presents the software implementation of the automated system for detecting and calculating the number of given structural objects, the description of the experimental setup, the setting up and conducting of experimental studies using the developed automated system, the description of the experimental environment, the evaluation of the efficiency of the automated system, and the results obtained are summarized.

The conclusions present the obtained scientific and practical results of the study.

In the appendices include scientific publications, which reflect the main scientific results of the work, information about the approval of the results of the dissertation, acts on the implementation of the results of the work, analysis of commercial digital gardening platforms and software listing.

Keywords: route, self-learning, drone, video stream, pattern recognition, artificial neural network, automated system.

#### Список публікацій здобувача за темою дисертації

*Наукові праці, в яких опубліковані основні наукові результати дисертації*

1. Мельниченко О. В. Архітектура автоматизованої системи розпізнавання сукупності структурних об'єктів однієї природи в тривимірному просторі. *Вимірвальна та обчислювальна техніка в технологічних процесах*. 2022. № 4(2022). С. 128–133. DOI: <https://doi.org/10.31891/2219-9365-2022-72-4-18>

2. Мельниченко О. В. Автоматизована система самоорганізації для керування безпілотними літальними апаратами з метою виявлення об'єктів. *Вимірвальна та обчислювальна техніка в технологічних процесах*. 2023. № 1(2023). С. 116–122. DOI: <https://doi.org/10.31891/2219-9365-2023-73-1-16>

3. Мельниченко О. В. Методи збору та обробки зображень отриманих з використанням БПЛА для виявлення заданих об'єктів. *Вісник Хмельницького національного університету. Технічні науки*. 2022. № 6, Т. 1(315). С. 131–137. DOI: <https://doi.org/10.31891/2307-5732-2022-315-131-138>

4. Melnychenko O. Method of real-time video stream synchronization in the working environment of an apple orchard. *Computer systems and information technologies*. 2023. No. 1(10). P. 91–97. DOI: <https://doi.org/10.31891/csit-2023-1-12>

5. Мельниченко О. Методи розпізнавання та обробки зображень за модифікованою YOLOv5-v1. *Information Technology: Computer Science, Software Engineering and Cyber Security*. 2023. Вип. 1. С. 74–84. DOI: <https://doi.org/10.32782/IT/2023-1-10>

6. Мельниченко О. В. Метод обчислення кількості структурних об'єктів у фруктовому саду в реальному часі. *Вісник Хмельницького національного університету. Технічні науки*. 2023. № 2(319), Т. 1. С. 213–219. DOI: <https://doi.org/10.31891/2307-5732-2023-319-1-213-214>

*Праці, які засвідчують апробацію матеріалів дисертації*

7. Melnychenko O., Savenko O. A self-organized automated system to control unmanned aerial vehicles for object detection. *The 4th International Workshop on Intelligent Information Technologies & Systems of Information Security (IntelITSIS-2023)* : CEUR-Workshop Proceedings. Vol. 3373. (Khmelnyskyi, Ukraine, 22–24 March 2023). Aachen : CEUR-WS.org, 2023. P. 589–600. URL: <https://ceur-ws.org/Vol-3373/paper40.pdf> (Scopus)

8. Мельниченко О. В. Метод та підсистема самовідновлення після критичних збоїв. *Актуальні проблеми комп'ютерних наук АПКН-2022* : Матеріали XIV всеукр. наук.-практ. конф., м. Хмельницький, 18–19 лист. 2022 р. Хмельницький : ХНУ, 2022. С. 202–204. URL: [https://kn.khmnmu.edu.ua/wp-content/uploads/sites/18/apkn2022\\_corpuspaper.pdf](https://kn.khmnmu.edu.ua/wp-content/uploads/sites/18/apkn2022_corpuspaper.pdf)

9. Мельниченко О. В. Самоорганізована система управління декількома БПЛА для динамічного отримання зображень в тривимірному просторі. *Математичне та програмне забезпечення інтелектуальних систем (МПЗІС-2022)*

: Тези доповідей XX міжнар. наук.-практ. конф., м. Дніпро, 23–25 лист. 2022 р. / під заг. ред. О.М. Кісельової. Дніпро : ДНУ, 2022. С. 139–140. URL: <http://mpzis.dnu.dp.ua/wp-content/uploads/2022/12/MPZIS-2022-1.pdf>

10. Мельниченко О. В. Метод керування та самоорганізована система планування маршрутів БПЛА. *2022 International Conference on Innovative Solutions in Software Engineering (ICISSE-2022)* : Conference Proceedings, Ivano-Frankivsk, Ukraine, 29–30 November 2022 / Kuz M., Kozenko M. eds. Ivano-Frankivsk : VSPNU, 2022. P. 26–30. URL: <https://shorturl.at/nyIMO>

11. Мельниченко О. В. Метод обчислення кількості розпізнаних структурних об'єктів певного класу. *Комп'ютерні ігри та мультимедіа як інноваційний підхід до комунікації* : Матеріали II Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів, м. Одеса, 29–30 верес. 2022 р. Одеса : Видавництво ОНТУ, 2022 р. С. 124–126. <https://card-file.ontu.edu.ua/handle/123456789/23485>

*Публікації, які додатково відображають наукові результати дисертації*

12. Мельниченко О. В. А. С. 116949, Україна. Комп'ютерна програма «Автоматизована система розпізнавання та обчислення кількості структурних об'єктів однієї природи». Дата реєстрації 09.03.2023.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	17
ВСТУП .....	18
РОЗДІЛ 1. АНАЛІЗ СУЧАСНИХ МЕТОДІВ ТА ПІДХОДІВ ДО РОЗПІЗНАВАННЯ ОБ’ЄКТІВ В РЕАЛЬНИХ УМОВАХ .....	26
1.1. Актуальність застосування інформаційних технологій до польових задач садівництва .....	26
1.2. Методи та засоби розпізнавання фруктових плодів в реальних умовах фруктового саду .....	30
1.2.1 Модель фруктового плоду у вигляді структурного об’єкта.....	31
1.2.2 Методи виявлення заданих структурних об’єктів за відеопотоком .....	34
1.2.3 Методи відстеження та ідентифікації заданих структурних об’єктів....	36
1.3. Огляд технічних пристроїв та апаратного забезпечення для задач садівництва .....	41
1.3.1 Актуальність використання безпілотних літальних апаратів в галузі садівництва .....	41
1.3.2 Керування робочими місіями групи безпілотних літальних апаратів ...	44
1.3.3 Використання групи безпілотних літальних апаратів для генерації фото- та відеоматеріалів.....	47
1.4. Постановка задачі .....	49
1.5. Висновки до першого розділу .....	50
РОЗДІЛ 2. МЕТОДИ ТА СИСТЕМА ДИНАМІЧНОГО ОТРИМАННЯ ЗОБРАЖЕНЬ ЗАДАНИХ ОБ’ЄКТІВ В ТРИВИМІРНОМУ ПРОСТОРИ.....	52
2.1. Автоматизована система динамічного отримання зображень заданих об’єктів в тривимірному просторі.....	53
2.2. Проєктування робочої місії БПЛА в тривимірному просторі .....	62
2.2.1 Метод побудови маршрутів БПЛА за технологією самонавчання.....	67
2.2.2 Інформаційні потоки автоматизованої системи під час виконання програмної місії.....	74

2.3. Генерація зображень заданих структурних об'єктів в тривимірному просторі робочого середовища.....	79
2.3.1 Створення програмної місії в системі .....	79
2.3.2 Програмний інтерфейс автоматизованої системи.....	83
2.3.3 Метод динамічного отримання зображень заданих структурних об'єктів в тривимірному просторі за допомогою декількох БПЛА .....	85
2.4. Висновки до другого розділу .....	91
<b>РОЗДІЛ 3. МОДЕЛІ ЗОБРАЖЕНЬ ЗАДАНИХ СТРУКТУРНИХ ОБ'ЄКТІВ В ЗАДАЧАХ РОЗПІЗНАВАННЯ ОБРАЗІВ .....</b>	<b>93</b>
3.1. Метод синхронізації відеопотоків в режимі реального часу .....	94
3.2. Метод та засоби виявлення, відстеження та обчислення кількості заданих структурних об'єктів .....	101
3.2.1 Метод виявлення заданих структурних об'єктів.....	101
3.2.2 Обчислення кількості заданих структурних об'єктів .....	110
3.3. Створення детектора у задачі відстеження.....	117
3.3.1 Підготовка робочого середовища .....	118
3.3.2 Підготовка навчальних наборів даних.....	118
3.3.3 Створення детектора.....	121
3.4. Тестування та результати створеного детектора .....	124
3.5. Висновки до третього розділу .....	129
<b>РОЗДІЛ 4. СТРУКТУРА, ОЦІНКА ДОСТОВІРНОСТІ ТА ЕФЕКТИВНОСТІ АВТОМАТИЗОВАНОЇ СИСТЕМИ ОБЧИСЛЕННЯ КІЛЬКОСТІ ЗАДАНИХ СТРУКТУРНИХ ОБ'ЄКТІВ ЗА ДОПОМОГОЮ ГРУПИ БПЛА .....</b>	<b>131</b>
4.1. Програмна реалізація автоматизованої системи виявлення та обчислення кількості заданих структурних об'єктів .....	131
4.1.1 Опис експериментальної установки .....	131
4.1.2 Засоби налаштування автоматизованої системи .....	134
4.2. Постановка і проведення експериментальних досліджень застосування розробленої автоматизованої системи.....	139
4.2.1 Опис експериментального середовища .....	139

4.2.2 Оцінювання ефективності автоматизованої системи.....	142
4.3. Висновки до четвертого розділу .....	150
ВИСНОВКИ .....	152
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	155
ДОДАТОК А. СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА .....	169
ДОДАТОК Б. ВІДОМОСТІ ПРО АПРОБАЦІЮ РЕЗУЛЬТАТІВ ЗА ТЕМОЮ ДИСЕРТАЦІЇ.....	172
ДОДАТОК В. АКТИ ВПРОВАДЖЕННЯ .....	174
ДОДАТОК Г. КОМЕРЦІЙНІ ПЛАТФОРМИ ЦИФРОВОГО САДІВНИЦТВА ..	184
ДОДАТОК Д. ЛІСТИНГ ПРОГРАМНОГО КОДУ .....	187



## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

AdaBoost	– Adaptive Boosting (мета-алгоритм статистичної класифікації)
CNN	– Convolutional Neural Network (згорткова нейронна мережа)
GPS	– Global Positioning System (система глобального позиціонування)
GT	– GroundTruth (валідаційна вибірка образів для тестування детектора)
IOU	– Intersection over Union (перетин двох множин)
KLT	– Kanade–Lucas–Tomasi (алгоритм Канаде–Лукаса–Томазі).
LF	– Loss Function (функція втрат)
OD	– Object Detection (пошук об'єкта)
PR	– Pattern recognition (розпізнавання образів)
RGB	– Red, Green, Blue (відтінки червоного, зеленого та синього)
PSNR	– індекс пікового співвідношення візуального сигналу до шуму
ReLU	– Rectified Linear Unit (випрямлений лінійний вузол)
ROI	– Region of Interest (область інтересу)
RTK	– Real Time Kinematic (позиціонування в режимі реального часу)
SE	– Squeeze and Networks
SSD	– Single Shot Detector (однопрохідна нейронно-мережева архітектура)
SSIM	– індекс структурної подібності
WC	– Weak Classifiers (слабкі класифікатори)
YOLO	– You Only Look Once (однопрохідна нейронно-мережева архітектура)
БПЛА	– безпілотний літальний апарат
БНТЗ	– безпілотні наземні транспортні засоби
ГЗНМ	– Глибока згорткова нейронна мережа
ГН	– Глибоке навчання
ЗНМ	– Згорткова нейронна мережа
ІТ	– Інформаційна технологія
ІС	– Інформаційна система
МН	– Машинне навчання
ШІ	– Штучний інтелект

## ВСТУП

**Актуальність роботи.** Використання безпілотних літальних апаратів (БПЛА) для виконання різних завдань набуло широко поширення. Можливість доповнення БПЛА відеокамерами для збору та отримання зображень покращила ефективність виконання завдань у важкодоступних місцях та суттєво автоматизувала таку роботу. Важливим напрямом їх використання є галузь садівництва, де необхідна автоматизація процесів обліку врожаю, яку можна виконати за допомогою безпілотних літальних апаратів. Така задача зводиться до застосування методів збору, отримання, обробки та аналізу інформації про задані об'єкти. Використання цих методів, а також стратегій, підходів, способів та технологій має важливе значення в контексті автоматизації процесів. Однак, суттєвими проблемами, які виникають в процесі використання БПЛА та збору зображень в реальному часі та в реальних умовах є зовнішні природні впливи, які впливають на ефективність функціонування автоматизованих систем з використанням БПЛА та, відповідно, збір зображень. Крім того, у випадку використання декількох БПЛА, крім позиціонування та узгодження їх роботи між собою із збору зображень, виникають проблеми з вибору маршруту та досягнення ними руху за заданими координатами в просторі.

Сучасні автоматизовані системи з використанням БПЛА відрізняються між собою, в залежності від функційних можливостей здійснювати збір, отримання, розпізнавання зображень заданих об'єктів, обчислення кількості заданих об'єктів, налаштуваннями, ефективністю роботи та точністю отриманих результатів. Залишаються актуальними питання забезпечення ефективності функціонування таких систем із збору та отримання зображень в реальних умовах і точність розпізнавання для подальшого обчислення кількості. Тому, вирішення задачі розробки методів збору, розпізнавання та обробки зображень отриманих з використанням БПЛА для виявлення заданих об'єктів, є однією із важливих наукових задач в сфері інформаційних технологій (ІТ), орієнтованих на побудову та подальшу експлуатацію автоматизованих систем з використанням БПЛА.

Незважаючи на обсяг виконаних в цьому напрямку наукових досліджень і, відповідно, отриманих наукових результатів та розробок, на сьогодні, надзвичайно актуальною, залишається задача покращення ефективності збору, точності розпізнавання, обробки зображень отриманих з використанням БПЛА для виявлення заданих об'єктів та вибір і узгодження маршрутів для групи БПЛА.

Значний внесок в розробку теорії розпізнавання образів зробили українські та іноземні вчені: В. Харченко [119], J. Pandian [29], Н. Liu [30], Р. Chu [31], S. Kamal [28], Н. Altaheri [42], Н. Sun [53], Х. Jiang [54], А. Саченко [119], А. Fernández [110].

Використанню БПЛА та їх груп, гарантуванню за будь-яких реальних умов їх функціонування, у тому числі в умовах природніх впливів і для вирішення різноманітних завдань, присвячені роботи І. Журавська [118], С. Лавренко [12], А. Boursianis [13], А. Rejeb [14], Р. Rallo [24], Р. Chen [26], Х. Dong [38], А. Саченко [120], С. Zhang [39].

Незважаючи на обсяг виконаних в цьому напрямку наукових досліджень і, відповідно, отриманих наукових результатів та розробок, на сьогодні, надзвичайно актуальною, залишається задача покращення ефективності збору, точності розпізнавання та обробки зображень отриманих з використанням БПЛА для виявлення заданих об'єктів.

Тому, актуальною науковою задачею є розробка методів збору, розпізнавання та обробки зображень, отриманих із використанням БПЛА, для виявлення заданих об'єктів, які б покращували ефективність збору, точність розпізнавання та обробки зображень, які отримані з використанням БПЛА, для виявлення заданих об'єктів.

### **Зв'язок роботи з науковими програмами, планами, темами.**

Дисертаційне дослідження виконувалось у рамках науково-дослідної тематики Хмельницького національного університету: держбюджетної науково-дослідної теми «Розроблення інформаційної технології прийняття контрольованих людиною критично-безпекових рішень за ментально-

формальними моделями машинного навчання» №2Б-2021 (№ держреєстрації 0121U112025), в якій автор дисертації був виконавцем.

**Мета і завдання дослідження.**

*Об'єкт дослідження* – процеси збору, розпізнавання та обробки зображень, отриманих із використанням БПЛА, для виявлення заданих об'єктів.

*Предмет дослідження* – методи і алгоритми забезпечення збору, розпізнавання та обробки зображень, отриманих із використанням БПЛА, для виявлення заданих об'єктів.

*Метою* дисертаційного дослідження є покращення ефективності збору, точності розпізнавання та обробки зображень, отриманих із використанням БПЛА, для виявлення заданих об'єктів.

**Задачі дослідження** формулюються в роботі наступним чином:

1. Провести аналіз методів, засобів та систем збору, розпізнавання та обробки зображень, отриманих із використанням БПЛА, для виявлення заданих об'єктів.

2. Розробити новий метод побудови маршрутів БПЛА згідно технологій самонавчання, щоб покращити переміщення та синхронізацію між групою БПЛА або одного БПЛА в межах робочого сегменту і за рахунок цього збільшити кількість опрацьованих даних.

3. Розробити новий метод динамічного отримання зображень заданих структурних об'єктів в тривимірному просторі за допомогою декількох БПЛА для узгодження між різними БПЛА та досягнення переміщення всієї групи БПЛА від заданих початкових до кінцевих точок програмної місії автономно.

4. Розробити новий метод синхронізації відеопотоків в режимі реального часу для забезпечення оперативного отримання результатів та здійснення виявлення структурних об'єктів, які були пропущені в процесі минулих програмних місій.

5. Вдосконалити метод виявлення заданих структурних об'єктів на зображеннях для покращення точності виявлення і зменшення часу навчання нейронної мережі.

6. Розробити новий спосіб отримання кількості заданих структурних об'єктів із отриманих зображень за допомогою групи БПЛА.

7. Розробити автоматизовану систему з групою БПЛА з використанням алгоритмів комп'ютерного зору, машинного навчання та оптимізації, здійснити постановку експерименту та провести з нею експериментальні дослідження щодо встановлення покращення точності виявлення заданих об'єктів на отриманих зображеннях та впровадити її у виробництво.

**Методи дослідження.** Для розв'язання поставлених задач використовуються основні положення абстрактної алгебри, теорія комп'ютерних мереж, теоретичні основи ІТ, методи захисту інформації в комп'ютерних системах, методи проектування інформаційних систем.

**Наукова новизна одержаних результатів** полягає в наступному:

1) розроблено новий метод побудови маршрутів БПЛА згідно технологій самонавчання, який на відміну від відомих, полягає в позиціюванні апаратних пристроїв у тривимірному координатному просторі в реальному часі за рахунок самонавчання при побудові маршрутів їх руху, що дає змогу покращити переміщення та синхронізацію між групою БПЛА або одного БПЛА в межах робочого сегменту і за рахунок цього збільшення кількості опрацьованих даних.

2) розроблено новий метод динамічного отримання зображень заданих структурних об'єктів в тривимірному просторі за допомогою декількох БПЛА, який на відміну від відомих відрізняється тим, що забезпечує в процесі активації кожного БПЛА генерацію відеоряду в центральну систему, прийняття рішень про подальшу роботу групи БПЛА, підтвердження цілісності програмної місії та визначення рівня критичності для продовження виконання роботи групи БПЛА за рахунок визначення станів модулів та комплексному врахуванні вимог розподілення, багаторівневості та автоматизованості, що покращує узгодження між різними БПЛА та досягнення переміщення всієї групи БПЛА від заданих початкових до кінцевих точок програмної місії автономно.

3) розроблено новий метод синхронізації відеопотоків в режимі реального часу, який відрізняється від відомих тим, що забезпечує накопичення даних про

попередні програмні місії БПЛА, у випадку критичних збоїв, які спотворюють цілісність структур даних із джерел отримання відеопотоків, зберігає такі структури в спеціальному журналі помилок та не відправляє їх як вхідні параметри в наступну обробку, що дає змогу виконувати порівняння отриманих поточних результатів із минулими в режимі реального часу і це забезпечує оперативне отримання результатів та здійснення виявлення структурних об'єктів, які були пропущені в процесі минулих програмних місій.

4) вдосконалено метод виявлення заданих структурних об'єктів на зображеннях, який на відміну від оригінальної архітектури YOLOv5, полягає в тому, що модифіковано модуль фокусування нейронної мережі, видалено згортковий шар поєднуючий вхідну карту ознак з операцією конкатенації, актуалізовано механізм візуальної уваги для вилучення ознак, об'єднання шарів 4 і 15, 6 і 11, 10 і 21 оригінальної архітектури замінено на об'єднання шарів 5 і 18, 8 і 14, 13 і 24 вдосконаленої архітектури, вихідні карти ознак 14-го та 21-го шарів вдосконаленої архітектури об'єднано між собою, що дало змогу покращити точність виявлення і зменшити час навчання нейронної мережі.

**Обґрунтованість і достовірність наукових положень, висновків і рекомендацій.** Наукові положення, висновки і рекомендації дисертації обґрунтовані коректним та доцільним використанням математичного апарату, алгоритмами побудови маршрутів, узгодження роботи різних БПЛА в групі, збору і отримання зображень в реальних умовах, розпізнавання заданих об'єктів та обчислення їх кількості, успішною реалізацією розробленої автоматизованої системи, ефективним практичним впровадженням результатів дисертаційного дослідження на підприємствах, що використовують такі автоматизовані системи, яке продемонструвало відповідність теоретичних досліджень з реальними результатами застосування.

**Практичне значення отриманих результатів.** Розроблена автоматизована система призначена для виявлення та обрахування кількості яблук у фруктовому саду в режимі реального часу. Перевагою розробленої системи над аналогами є отримання нею множини відеокадрів у режимі реального часу з камер кількох

БПЛА та синхронізація цих відеокадрів між собою в одну інформаційну структуру даних, що надалі трансформується в суцільне зображення. Крім того, використання функцій оптимізації якості зображення дає змогу максимально ефективно виявляти структурні під час виконання робочих місій БПЛА в робочому середовищі. Використання такого засобу трансформації дало змогу системі отримувати суцільний потік даних до всіх наступних програмних компонентів автоматизованої системи. Так, оцінка синхронізації відеопотоків за індексом SSIM коливається від 0,79 до 0,92, із середнім значенням 0,87, а за індексом PSNR – від 22 до 39, що свідчить про високу ефективність роботи розробленої системи із відеопотоками та належну якість отриманих об'єднаних зображень.

У результаті проведених експериментальних досліджень було доведено ефективність роботи розробленої автоматизованої системи, що підтверджується високим середнім значенням у 82,69% показника достовірності виявлення та обчислення кількості фруктових плодів та низьким середнім рівнем помилок I (14,67%) та II (18,33%) роду.

Теоретичні та практичні результати дослідження впроваджені в ТОВ «ЮКС++» (м. Хмельницький), Державному підприємстві «Новатор» (м. Хмельницький), ПП «НОЛТ ТЕХНОЛОДЖИС» (м. Хмельницький), ТОВ «Агротех сервіс», а також, в освітньому процесі Хмельницького національного університету при викладанні дисциплін на кафедрі комп'ютерної інженерії та інформаційних систем для спеціальності 126 Інформаційні системи та технології, 123 Комп'ютерна інженерія та кафедрі комп'ютерних наук для спеціальності 122 Комп'ютерні науки, зокрема в курсах «Методи та системи штучного інтелекту», «Комп'ютерні та кіберфізичні системи», «Методи, засоби та алгоритми в задачах обчислювального інтелекту та комп'ютерного зору», «Теорія, проєктування та моделювання спеціалізованих комп'ютерних систем» та «Технології проєктування інформаційних систем». Акти про впровадження результатів дисертаційної роботи наведено у додатку В.

**Особистий внесок здобувача.** Всі основні результати дисертаційного дослідження, які представлені до захисту, одержані автором особисто. В роботах,

опублікованих одноосібно автором, отримано наступні результати: [121, 122, 128, 129, 130] – розроблено метод побудови маршрутів БПЛА згідно технологій самонавчання; [123, 129, 130] – розроблено метод динамічного отримання зображень заданих структурних об'єктів в тривимірному просторі за допомогою декількох БПЛА для узгодження між різними БПЛА та досягнення переміщення всієї групи БПЛА; [124] – розроблено метод синхронізації відеопотоків в режимі реального часу; [123, 125, 131] – вдосконалено метод виявлення заданих структурних об'єктів на зображеннях; [125, 126, 131] – розроблено спосіб отримання кількості заданих структурних об'єктів із отриманих зображень за допомогою групи БПЛА; [122, 131, 140] – розроблено автоматизовану систему з групою БПЛА з використанням алгоритмів комп'ютерного зору, машинного навчання та оптимізації. У роботі, опублікованій у співавторстві, автору належать основні ідеї, теоретична та практична розробка положень, відображених у характеристиці наукової новизни отриманих результатів, а саме: [127] – розроблено автоматизовану систему з групою БПЛА та метод побудови маршрутів БПЛА для збору зображень об'єктів.

Список наукових публікацій за результатами дисертаційної роботи наведено у додатку А.

**Апробація результатів дисертації.** Апробацію основних положень, ідей, висновків дисертаційної роботи проведено на науковому семінарі кафедри комп'ютерної інженерії та інформаційних систем у Хмельницькому національному університеті. Наукові результати роботи доповідалися, також, на: II Всеукраїнській науково-технічній конференції молодих вчених, аспірантів та студентів «Комп'ютерні ігри та мультимедіа як інноваційний підхід до комунікації» (м. Одеса, 29-30 вересня 2022); XIV всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2022» (м. Хмельницький, 18-19 листопада 2022 р.); XX міжнародній науково-практичній конференції «Математичне та програмне забезпечення інтелектуальних систем» (м. Дніпро, 23-25 лист. 2022); International Conference on Innovative Solutions in Software Engineering (Ivano-Frankivsk, Ukraine, November



29-30, 2022); 4th International Workshop on Intelligent Information Technologies & Systems of Information Security (Khmelnyskyi, Ukraine, March 22-24, 2023).

Відомості про апробацію результатів дисертаційної роботи наведено в додатку Б.

**Публікації.** За результатами проведених досліджень основні наукові результати опубліковано у 6 наукових статтях у фахових наукових журналах України [121-126]. Апробація засвідчена публікаціями 5 праць в матеріалах міжнародних та всеукраїнських конференцій [127-131], з яких одна праця індексована у наукометричній базі Scopus [127]. Опубліковано 1 свідоцтво про реєстрацію авторського права на твір (програму) [140].

**Структура та обсяг дисертації.** Дисертаційна робота складається з анотації, змісту, переліку умовних скорочень, вступу, чотирьох розділів, висновку, списку використаних джерел та п'яти додатків. Повний обсяг роботи містить 204 сторінки друкованого тексту, з них анотація – на 12 стор., зміст – на 3 стор., перелік умовних скорочень – на 1 стор., основний текст – на 137 стор., список із 140 використаних джерел – на 14 стор., додатки – на 36 стор. Дисертація містить 42 рисунки та 12 таблиць.

## РОЗДІЛ 1.

### АНАЛІЗ СУЧАСНИХ МЕТОДІВ ТА ПІДХОДІВ ДО РОЗПІЗНАВАННЯ ОБ'ЄКТІВ В РЕАЛЬНИХ УМОВАХ

Багато наукових задач, які спрямовані на розв'язання завдань в різних галузях промисловості, потребують застосування методів отримання, обробки та аналізу інформації про об'єкти з різних предметних областей. Ці методи, а також стратегії, підходи, способи та технології, мають важливе значення в контексті автоматизації процесів. Розглянемо їх застосування до задач з розпізнавання об'єктів. Завдяки швидкому розвитку технологій, застосуванню різноманітних інноваційних методів у садівництві, підвищується ефективність вирощування фруктів та знижуються витрати ресурсів. Одним з ключових напрямків розвитку цієї сфери є автоматизація процесів обліку врожаю за допомогою безпілотних літальних апаратів. Однак, проблеми розпізнавання фруктів для обліку технічними системами у реальних умовах залишаються актуальними і вимагають розробки нових методів та підходів до їх вирішення. Здійснимо аналіз відомих методів та підходів до розпізнавання фруктових плодів в реальних умовах фруктового саду, реалізація та впровадження яких може сприяти покращенню ефективності обліку врожаїв в садівництві.

#### 1.1. Актуальність застосування інформаційних технологій до польових задач садівництва

На розвиток сільського господарства у всьому світі впливають проблеми, що включають зміну клімату, скорочення запасів води для зрошення, збільшення витрат на виробництво та загальне зменшення кількості робочої сили протягом останніх десятиліть. Крім того, в Україні найактуальнішою проблемою сьогодення та негативним викликом для галузі сільськогосподарського виробництва є війна та окупація територій [1], що призвела до руйнування систем виробництва та постачання продуктів харчування [2]. Ці чинники загрожують екологічній та

економічній стійкості поточних та майбутніх систем продовольчого забезпечення [3, 4]. Хоча сільське господарство постійно розвивається, необхідні значні інновації, щоб пристосуватися до постійних змін клімату [5-7]. Проблема полягає в тому, як забезпечити стає виробництво продуктів харчування для швидко зростаючого населення світу, зокрема, з огляду на економічні проблеми в Україні, що пов'язані з війною [2].

Дослідники в галузі сільського господарства завжди використовували передові технології та постійно шукали способи їх інтеграції в сільськогосподарські системи [8]. Динамічні імітаційні моделі різних сільськогосподарських культур були важливими інструментами для інтеграції різноманітних компонентів сільського господарства, включаючи садівництво та дозволяли вивчати, як ці компоненти функціонують у системі [9, 10].

На сьогодні системам та засобам штучного інтелекту (ШІ) надають значну увагу для їх застосування в галузі сільського господарства через їх потенціал в обробці великих даних, які тепер стають доступними завдяки безпілотним літальним апаратам (БПЛА) [11, 12]. БПЛА надає суттєву можливість використовувати розширену аналітику для управління інформаційними системами в садівництві, покращуючи стійкість та ефективність виробничих систем [13].

За останні роки значно зросла кількість наукових та науково-прикладних джерел, що стосуються використання БПЛА в сільському господарстві. Це свідчить про збільшення інтересу до такого наукового та інженерного напрямку. Кількість дослідницьких статей, в яких розглядають використання БПЛА в сільському господарстві, зросла з 544 у 2018 р. [14] до 1593 у 2022 р. [15, 16]. Зокрема, найбільше зростання відбулося у таких сферах, як дистанційне зондування [17], візуальна навігація [19], сталий розвиток довкілля [18], дика природа та садівництво [20] тощо. Крім того, дослідники використовували БПЛА для оцінки водного стресу, моніторингу захворювань культур, картографування бур'янів та оцінювання біомаси і врожайності фруктових культур [21, 22]. Інші роботи показали, що інформаційні системи із БПЛА можуть бути успішно використані для оцінки параметрів культур садівництва таких як висота пологів,

покриття пологів та індекси рослинності [23], вибору генотипів [24] та прогнозування врожайності [25].

Попри те, що деякі спеціалізовані інформаційні системи розпочали застосовувати БПЛА, перед тим як цільові користувачі таких систем зможуть повністю прийняти їх для використання, необхідно вирішити значні довгострокові проблеми, пов'язані зі збором та обробкою даних, а також інтерпретацією оброблених даних. Оскільки необроблені дані проходять через процес розробки програми, то їх цілісність і якість є вирішальними для забезпечення точності прогнозних моделей. Подібний процес зображено на рис. 1.1 [21].

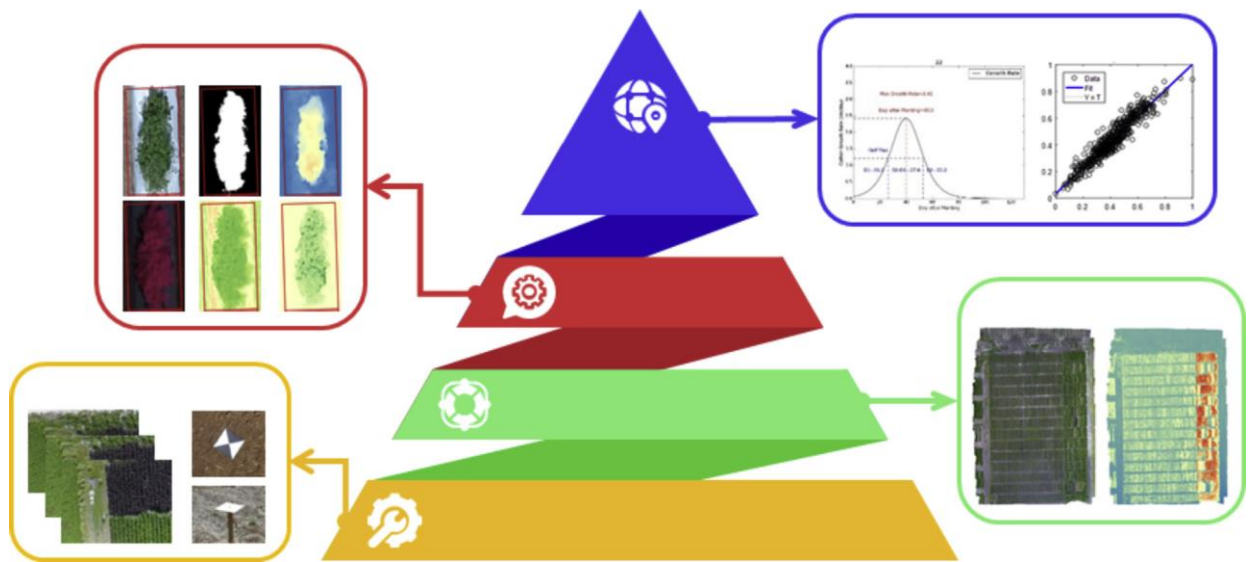


Рисунок 1.1 – Процес проєктування та розроблення інформаційних систем в галузі сільського господарства [21]

За дослідженнями, які були проведені, обсяг навчальних даних є важливим чинником для розробки надійних моделей ШІ в садівництві. Навіть якщо до набору даних залучені дані з шумом [26], то продуктивність моделей залишається високою. Таким чином, окрім якості необроблених даних, великий обсяг даних для навчання важливий для створення надійних моделей. У садівництві активно використовуються методи та засоби ШІ. Короткий аналіз найбільш популярних підходів до застосування ШІ для задач сільського господарства подано в табл. 1.1.

Таблиця 1.1

## Методи ШІ, які використовуються для задач сільського господарства

Застосування	Урожай	Тип даних	Метод	Точність, %
Прогнозування врожайності	Пшениця	Геодані з БПЛА	CNN [27]	77.01
Виявлення бур'янів	Рис	RGB з БПЛА	Fully-CNN [28]	94.00
Виявлення хвороб	Банан	RGB з поля	DCNN [29]	91.00
Виявлення фруктів	Цитрусові	RGB з поля	R-CNN [30]	90%
Виявлення фруктів	Яблука	RGB з поля	R-CNN [31]	86%

За останні десятиліття на ринку з'явилося чимало комерційних рішень, що сприяють веденню цифрового садівництва та надають послуги розпізнавання плодів різних культур з використанням БПЛА. У табл. Г.1 додатку Г подано результати аналізу подібних комерційних продуктів [32-36].

Відповідно до аналізу комерційних платформ, результати якого подано в табл. В.1, автоматизоване розпізнавання фруктових плодів за допомогою БПЛА має низку переваг, як от, висока точність виявлення плодів [37], моніторинг у реальному часі [38] та інтеграція з іншими технологіями цифрового садівництва [39]. Проте, значними недоліками [40] таких платформ є доступність малої кількості типів культур для розпізнавання, обмежена географічна доступність і висока плата за підписку. Отже, з проведеного аналізу встановлено, що єдиного універсального рішення для автоматизованого розпізнавання фруктових плодів на ринку немає, що зумовлює проведення додаткового дослідження та розроблення нових інформаційних систем.

У підсумку, застосування БПЛА в садівництві може стати ефективним засобом для збільшення врожайності та оптимізації використання ресурсів. Разом з тим, галузь цифрового сільського господарства загалом та садівництва зокрема стикається із різними актуальними проблемами [41], які потребують вирішення.

Одна з головних проблем – це точність даних, зібраних БПЛА. Для досягнення позитивних результатів, потрібно, щоб БПЛА, наприклад, мали точну карту саду, а також точну інформацію про різні типи дерев та плодів. Важливо також мати точну інформацію про кліматичні умови та ґрунт. Інша проблема полягає в тому, що БПЛА можуть завдати шкоди рослинам під час польоту, наприклад, коли їх тіла зіткнуться з деревами або гілками. Також, БПЛА можуть розкидати пестициди або здійснювати знімання фото- або відеоматеріалів нерівномірно, що може привести до незадовільних бізнес-результатів.

Одним з можливих шляхів вирішення проблем з БПЛА в садівництві є підвищення точності картографування та розроблення додаткових датчиків, що зменшать можливість пошкодження рослин. Також, потребує розробити програмне забезпечення, яке допоможе аналізувати дані, що зібрані БПЛА та розробляти стратегії для їх ефективного використання. Крім того, розробка нових технологій для збору та аналізу даних таких як сенсори, машинне навчання (МН) та ШІ також може допомогти вирішити ці проблеми та зробити застосування БПЛА більш ефективними в садівництві.

## 1.2. Методи та засоби розпізнавання фруктових плодів в реальних умовах фруктового саду

Правильний підхід до пошуку потрібних сортів та правильного вибору фруктів полягає в усвідомленні природи проблеми. Класифікація фруктів, через велику кількість сортів, є відносно складною проблемою [42]. Види та різновиди мають значні варіації зовнішнього вигляду, включаючи неправильні форми, кольори та текстури. Крім того, зображення вузько покривають блискавичний пейзаж камери, відстань та кут, що сприяє розмитості зображень. Часткова або повна закритість об'єкта є ще однією проблемою. Цей недолік призвів до відсутності в реальних реалізаціях мультикласових автоматизованих систем класифікації фруктів [43]. За останні роки проведено десятки досліджень, щоб ідентифікувати та класифікувати фрукти з різними цілями та застосуванням [44].

Попередня обробка та сегментація є важливими етапами під час виявлення та класифікації об'єктів будь-якого типу. Оскільки фрукти відрізняються за формою, розміром, кольором і текстурою, попереднє оброблення є першою і найважливішою дією для виявлення та класифікації фруктів. Під час етапу попередньої обробки отримані зображення обробляються для видалення фонового шуму, тим самим виділяючи зображення фруктів. Після отримання набору даних, зображення плодів трансформують в сіре зображення з кольорового зображення RGB перед перетворенням у бінарне зображення. З моменту появи глибокого навчання (ГН) вилучення ознак стало широко використовуваним етапом попередньої обробки. Для цього використовуються методи МН та ГН, що дозволяють вилучити елементи фруктів такі як форма, кольорова гама та розмір та перетворити їх у векторні елементи.

Об'єкти, що є предметом інтересу на зображенні, часто сегментуються за допомогою таких методів, як видалення фону. Віднімання фону можна розглядати як спосіб розділення двох об'єктів на зображенні. Методи сегментації ґрунтуються на OTSU, ураження, ROILS або краях, щоб виділити область інтересу (ROI) від фону [45]. Після цього виконується процес злиття, щоб інтегрувати вектори ознак у кінцевий вектор. Оскільки сегментація є обов'язковою для виявлення об'єктів, використовувана архітектура може бути розділена на дві: повністю згорткову нейронну мережу (ЗНМ, з англ. convolutional neural network, CNN) [46] та Mask-RCNN [47]. Повнозв'язна ЗНМ під назвою Fully-CNN [48] для задач садівництва була запропонована у роботі [49], де використовується звичайний FCN, U-Net та SigNet. Спочатку ROI витягується з набору даних про фрукти, після чого видаляється фон сегментованого зображення. Далі до сегментованого зображення застосовується функція класифікації для досягнення мети виявлення.

### 1.2.1 Модель фруктового плоду у вигляді структурного об'єкта

Результатом розв'язання задачі виявлення фруктового плоду є множина обмежувальних рамок разом із розпізнаними класами плодів. У разі застосування

архітектури ЗНМ до розв'язання задачі виявлення, магістральний шар мережі зменшує розмірність вхідного зображення на коефіцієнт, що має назву крок мережі [50, 51]. Далі ознаки, виділені на згорткових шарах ЗНМ, передаються на останній шар класифікації, що обчислює прогнозні значення координат розташування об'єкта в реальному часі. Тобто вихідні прогнозні значення ЗНМ утворюють матриці мап ознак із першими двома вимірами попереднього шару.

Виходом згорткового шару  $1 \times 1$  в ЗНМ, що призначена для виявлення об'єктів, є вектор ознак певного розміру (зазвичай, 255), який поділяють на  $B$  секцій. Кожна секція вектору відповідає за обчислення прогнозних значень координат певної обмежувальної рамки. Водночас кожна обмежувальна рамка характеризується координатами центру  $t_x$ ,  $t_y$ , ширини  $t_w$  та висоти  $t_h$ . Крім того, кожна рамка супроводжується оцінкою достовірності  $t_o$ , яка відображає впевненість мережі в тому, що рамка дійсно містить об'єкт, і наскільки точно вона вважає, що він має правильний розмір. Показник достовірності  $t_o$  формалізовано за формулою

$$t_o = Pr(Object) \times IOU_{pred}^{truth}, \quad (1.1)$$

де  $IOU_{pred}^{truth}$  – одиниця перетину істинної та прогнозованої рамок.

Кожна секція вектору ознак із множини секцій  $B$  містить також ймовірності умовного класу  $C$ ,  $Pr(Class_i | Object)$ , що нормалізовані в ЗНМ функцією активації.

У підсумку, для кожного потенційного на зображенні об'єкта згортковий шар  $1 \times 1$  створює масив елементів  $(B \cdot (5 + C))$ , де  $C$  – це кількість виявлених класів, а  $B$  – кількість обмежувальних рамок, що створюються мережею (зазвичай,  $B = 3$ ). Важливо зазначити, що мережа не прогнозує координати безпосередньо, втім уточнює обрані вручну пріоритети, відомі як якоря.

На рис. 1.2 зображено приклад подання цільового об'єкта у вигляді яблука [50], що описується координатною сіткою розміром  $13 \times 13$ .



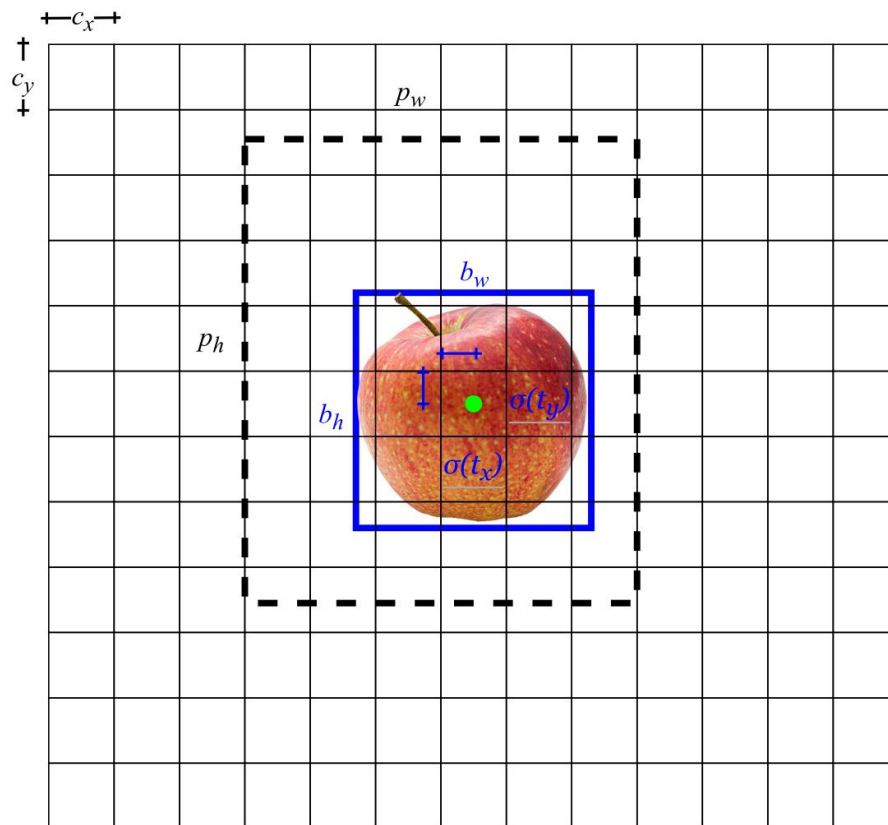


Рисунок 1.2 – Схема обчислення прогнозних значень координат обмежувальних рамок нейромережевою моделлю, як зміщення до попередньо ініціалізованих вручну значень координат прив'язувальних рамок [50]

Відповідно до рис. 1.2, центр цільового об'єкта потрапляє в окрему комірку сітки  $13 \times 13$ . Ця комірка призначена для визначення наявності об'єкта на координатній площині. Під час свого навчання мережа використовує ініціалізовану прив'язувальну рамку із вищим значенням IOU (Intersection over Union). Відповідно мережа генеруватиме нові значення  $t_x$ ,  $t_y$ ,  $t_w$  та  $t_h$  для уточнення значень координат прив'язувальної рамки, що присутня у комірці із центром цільового об'єкта. Тоді, до характеристик  $t_x$ ,  $t_y$ ,  $t_w$  та  $t_h$  обмежувальної рамки застосовуються наступні перетворення:

$$\begin{aligned} b_x &= \sigma(t_x + c_x); \\ b_y &= \sigma(t_y + c_y), \end{aligned} \quad (1.2)$$

$$\begin{aligned} b_w &= p_w e^{t_w}; \\ b_h &= p_h e^{t_h}, \end{aligned} \quad (1.3)$$

де  $b_x$  та  $b_y$  – обчислені координати центру  $x$ ,  $y$  обмежувальної рамки,  $b_w$  та  $b_h$  – обчислені ширина та висота обмежувальної рамки,  $c_x$  та  $c_y$  – верхні ліві координати сітки, що описує цільовий об’єкт,  $p_w$  та  $p_h$  – ініціалізовані заздалегідь координати прив’язувальної рамки.

У підсумку, нейромережева модель виводить матрицю розміром  $N \times N \times (3 \cdot (5 + C))$ .

Таким чином, для кожної координатної сітки розміром з  $N \times N$  мережа виводить три обмежувальні рамки (1.2-1.3) з показником достовірності (1.1) та ймовірностями  $p_c$  класу. Далі оцінка достовірності  $\sigma(t_o)$  використовуються для видалення кількох виявлень одного й того ж цільового об’єкта і отримання фінальних прогнозних значень координат обмежувальної рамки.

### 1.2.2 Методи виявлення заданих структурних об’єктів за відеопотоком

Для точного обчислення кількості фруктових плодів в реальному середовищі фруктового саду потрібна система ідентифікації фруктів, яка автоматизує процес виявлення та маркування плодів. Хоча дослідники розглядали питання виявлення фруктів, як це показано в [52], проблема розробки швидкої та надійної системи виявлення фруктів все ще залишається актуальною, як зазначено в [53]. Це пов’язано з тим, що зовнішній вигляд фруктів у польових умовах, включаючи колір, тип, масштаб, текстуру та властивості відображення, дуже різноманітний.

Нещодавно глибокі згорткові нейронні мережі (ГЗНМ) досягли значних успіхів у класифікації та ідентифікації об’єктів. У сучасній системі виявлення передбачено два етапи [54]. На першому етапі конвеєра використовується регіональний запропонований метод для виділення цікавих областей із зображення, наприклад вибірковий пошук та межовий блок, а потім передача їх до класифікації з використанням ГЗНМ [55]. Попри значні досягнення в галузі оброблення зображень, використання цієї системи в реальному часі є обчислювально витратною. Цю проблему можна вирішити, інтегруючи архітектуру ГЗНМ для класифікації з мережею пропозицій об’єктів, щоб система

могла одночасно прогнозувати та класифікувати межі об'єктів у кожному місці. Параметри двох мереж є спільними, що призводить до значного зменшення обчислювальних витрат і таким чином, оптимізується для застосування у робототехнічних цілях.

У табл. 1.2 наведено опис різних моделей ГН, які застосовуються для вирішення проблем виявлення фруктів.

Таблиця 1.2

Порівняння моделей ГН, що використовуються для виявлення фруктових плодів

Посилання та рік	Модель ГН	Набір даних	Точність виявлення	Недоліки
[56], 2020	Faster R-CNN	Яблуневий сад	< 83%	Залежність від попереднього набору даних для передавального навчання
[31], 2021	Mask R-CNN	Яблука	> 88%	Потреба у великій кількості даних
[57], 2022	RetinaNet-FPN	Полуниця	91%	Велика ймовірність перенавчання моделі
[58], 2022	YOLOv5	Яблука	>90%	Високий показник хибно позитивних виявлень
[59], 2023	SSD+ YOLOv5	Цитрусові та овочі	~90%	Низька точність виявлення малих об'єктів

З табл. 1.2 можемо виділити кілька моделей ГН, які використовують для виявлення фруктових плодів у саду. Зазначимо, що вибір моделі залежить від конкретної програми, типу використовуваних зображень або відео та доступних обчислювальних ресурсів. Нижче подамо короткий опис моделей ГН, які часто використовуються для виявлення яблук.

1. Faster R-CNN [56] – модель ГН, що широко використовується для різних виявлення об'єктів на зображеннях. Він використовує мережу пропозицій регіонів для створення множини так званих регіонів, які, ймовірно, містять об'єкти, а потім класифікує кожен регіон як цільовий плід або фон.

2. Mask R-CNN [31] –розширення Faster R-CNN, яка також передбачає маску сегментації для кожного виявленого об'єкта на зображенні. Такий підхід може бути корисним для точної локалізації плода на зображенні.

3. RetinaNet [57] – модель розроблена для вирішення проблеми дисбалансу класів у виявленні об'єктів, коли фонових областей набагато більше, ніж областей об'єктів на зображенні. Вона використовує функцію фокусних втрат, щоб зменшити втрату для простих прикладів і збільшити втрату для складних прикладів, що допомагає покращити виявлення малих об'єктів на зображенні, таких як фруктові плоди, зокрема, яблука.

4. YOLO (You Only Look Once) [58] – популярний тип моделі ГН для виявлення об'єктів на зображеннях та відео в режимі реального часу. Вона ділить зображення на сітку комірок і передбачає обмежувальні рамки та ймовірності класу для кожної комірки.

5. SSD (Single Shot Detector) [59] – модель схожа на YOLO тим, що вона передбачає координати обмежувальних рамок та ймовірності класу для кожної комірки сітки, але використовує багатомасштабну карту функцій для підвищення точності.

З проведеного аналізу встановлено, що архітектура ЗНМ під назвою YOLO є відмінним рішенням для розв'язання задачі виявлення яблук у фруктовому саду. Модель YOLO може слугувати компонентом системи для швидкої й точної ідентифікації яблук на зображенні чи відео.

### 1.2.3 Методи відстеження та ідентифікації заданих структурних об'єктів

Поширеною проблемою в садівництві є точне оцінювання врожайності. Вирішення цієї проблеми є суттєвим, оскільки дає можливість профільним підприємствам оптимізувати врожайність, краще планувати врожай на наступний сезон і ефективно управляти своїми ресурсами [60, 61]. Традиційні підходи до оцінювання врожайності, такі як ручне обчислення кількості плодів, переважно займають багато часу та призводять до багатьох помилок. У зв'язку із цим, у

цифровому садівництві віднедавна почали використовувати автоматизовані системи підрахунку плодів у садах, що ґрунтуються на математичних та обчислювальних методах.

Для розв'язання задачі обчислення кількості фруктів часто використовуються два методи: фільтр Калмана [62] та угорський алгоритм [63]. Фільтр Калмана – це математичний алгоритм, який оцінює стан системи на основі вимірювань шуму. Водночас угорський алгоритм, також відомий як алгоритм Мункреса, є комбінаторним алгоритмом оптимізації, який вирішує проблему призначення у дводольних графах [64]. Розглянемо принципи застосування цих методів до розв'язання задачі обчислення кількості фруктових плодів.

Фільтр Калмана – це математичний метод, який здійснює багаторазові вимірювання протягом певного часу для оцінки стану певної системи. Під час обчислення кількості плодів фільтр Калмана оцінює кількість плодів у саду, вимірюючи їхню щільність. Алгоритм Калмана складається з двох етапів: етапу передбачення та етапу корекції.

1. Етап передбачення. Алгоритм оцінює стан системи на наступному часовому кроці на основі попереднього стану та динаміки системи. Динаміка системи моделюється як лінійна функція вектору стану та вхідного вектору керування. Етап передбачення формалізується такими рівняннями:

$$\begin{aligned}\hat{x}_{k+1|k} &= F_k \hat{x}_{k|k} + B_k u_k; \\ P_{k+1|k} &= F_k P_{k|k} F_k^T + Q_k,\end{aligned}\tag{1.4}$$

де  $\hat{x}$  – прогнозований стан у момент часу  $k + 1$  за даними вимірювань до часу  $k$ ,  $F_k$  – матриця переходів станів, яка моделює динаміку системи,  $B_k$  – вхідна матриця керування,  $u_k$  – вхідний сигнал керування в момент часу  $k$ ,  $P_{k+1|k}$  – прогнозована коваріаційна матриця,  $Q_k$  – коваріаційна матриця шуму процесу.

2. Етап корекції. Алгоритм оновлює оцінку стану на основі вимірювань у момент часу  $k + 1$ . Вимірювання моделюються як лінійна функція вектору стану та

шумового члена спостереження. Етап корекції подано такими математичними співвідношеннями:

$$\begin{aligned} K_{k+1} &= P_{k+1|k} H_{k+1}^T (H_{k+1} P_{k+1|k} H_{k+1}^T + R_{k+1})^{-1}; \\ \hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + K_{k+1} (z_{k+1} - H_{k+1} \hat{x}_{k+1|k}); \\ P_{k+1|k+1} &= (I - K_{k+1} H_{k+1}) P_{k+1|k}, \end{aligned} \quad (1.5)$$

де  $K_{k+1}$  – підсилення Калмана,  $H_{k+1}$  – матриця вимірювання, яка відображає вектор стану в простір вимірювання,  $z_{k+1}$  – вимірювання в момент часу  $k + 1$ ,  $R_{k+1}$  – коваріаційна матриця шуму вимірювання,  $I$  – матриця пікселей, що відображає вхідне зображення.

Два етапи (1.4)-(1.5) фільтру Калмана забезпечує оптимальний результат у тому сенсі, що він мінімізує середню квадратичну помилку між оціненим станом і реальним станом. Заразом припускається, що динаміка системи та модель вимірювання є лінійними, а шум – Гаусовським [65]. Подібні вимоги до застосування фільтру Калмана не завжди можуть бути виконані в реальних умовах.

Угорський алгоритм – це комбінаторний алгоритм оптимізації, який призначений, насамперед, для розв’язання задачі про призначення. Ця задача є фундаментальною в комп’ютерних науках, дослідженні операцій та оптимізації, і спрямована на оптимальний розподіл ресурсів поміж завдань. У випадку обчислення кількості плодів задача про призначення полягає в тому, щоб зіставити передбачуване положення фруктів із фактичними фруктами на зображенні. Угорський алгоритм розв’язує цю задачу, знаходячи оптимальне призначення, яке мінімізує загальну відстань між оціненими позиціями та фактичними плодами. Розглянемо ідею застосування Угорського алгоритму до обчислення кількості плодів.

Вхідними даними для Угорського алгоритму є матриця відстаней, де кожен елемент відображає відстань між розрахунковою позицією плоду та фактичним розміщенням плоду на зображенні. Зазвичай за функцію відстані приймають

відстань Махаланобіса [66] між двома положеннями на зображенні. Алгоритм полягає у виконанні наступних дій.

1. Віднімається мінімальне значення в кожному рядку з усіх елементів поточного рядка.

2. Віднімається мінімальне значення в кожному стовпці з усіх елементів поточного стовпця.

3. Розраховується мінімальна кількість рядків, які можна провести, щоби покрити всі нулі в матриці. Якщо кількість таких ліній дорівнює кількості рядків, оптимальне рішення знайдено, й алгоритм завершує роботу. В іншому випадку переходять кроку 4.

4. Розраховується мінімальний непокритий елемент у матриці. Він віднімається від усіх непокритих елементів і додається до всіх елементів, які покриті двома лініями. Переходять до кроку 3.

5. Будується оптимальне призначення з матриці. Кожен рядок і стовпець з одним нулем позначає плід, якому було призначено оціночне місце.

Для точного обчислення кількості плодів Угорському алгоритму потрібно подати на вхід повну матрицю відстаней, що може бути обчислювально складним завданням під час обчислення великої кількості плодів на одному зображенні. Крім того, алгоритм передбачає, що кількість передбачуваних позицій дорівнює кількості фактичних фруктів на зображенні, що не завжди може бути істинним. Водночас Угорський алгоритм можна розширити для обробки випадку, коли кількість оцінених позицій відрізняється від кількості фактичних плодів, через введення фіктивних позицій. У цьому випадку матриця відстаней доповнюється для включення фіктивних позицій, що відповідають уявним плодам, і алгоритм продовжує роботу, як і раніше.

Отже, фільтр Калмана та Угорський алгоритм є ефективними методами обчислення кількості плодів, які можна використовувати в поєднанні для отримання точних і надійних оцінок кількості плодів у фруктовому саду. Фільтр Калмана дає змогу відстежувати переміщення фруктів у часі та оцінювати їхнє розташування, тоді як Угорський алгоритм зіставляє передбачуване положення

плодів із фактичними фруктами на зображенні, ідентифікуючи в такий спосіб унікальні плоди. Впровадження цих двох методів у єдину систему комп'ютерного зору може слугувати засобом для точного, надійного та масштабованого обчислення кількості плодів у фруктовому саду в режимі реального часу.

Введемо математичну формалізацію статистичних показників [67], що використовуються для оцінювання моделей ідентифікації фруктових плодів. Для цього позначимо через  $P$  кількість істинно позитивних випадків виявлення плодів, а через  $N$  – істинно негативних. Тоді, результати ідентифікації розділимо на істинно позитивні (TP), істинно негативні (TN), хибно позитивні (FP) та хибно негативні (FN) випадки. Нижче опишемо використані статистичні критерії через відомі математичні співвідношення.

Точність (Accuracy) класифікації визначається так

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1.6)$$

Влучність (Precision) класифікації позначається так

$$\text{Precision} = \frac{TP}{TP + FP} \quad (1.7)$$

Повнота (Recall) класифікації формалізована у такий спосіб

$$\text{Accuracy} = \frac{TP}{TP + FN} \quad (1.8)$$

Показник  $F_1$ -норми визначено так

$$F_1 = \frac{2TP}{2TP + FP + FN} \quad (1.9)$$



Помилка I роду виражається так

$$FPR = \frac{FP}{FP + TN}, \quad (1.10)$$

Помилку II роду визначено так

$$FNR = \frac{FN}{TP + FN}, \quad (1.11)$$

Крім критеріїв (1.6-1.11), модель ідентифікації оцінюється за часом навчання нейронної мережі, що виражений в годинах роботи обчислювального процесора.

Таким чином, здійснено аналіз відомих методів ідентифікації та обчислення кількості заданих структурних об'єктів, а також критеріїв оцінювання отриманого результату, що є необхідним для розпізнавання та обробки зображень отриманих з використанням БПЛА при виявленні заданих об'єктів.

### 1.3. Огляд технічних пристроїв та апаратного забезпечення для задач садівництва

#### 1.3.1 Актуальність використання безпілотних літальних апаратів в галузі садівництва

Сьогодні на ринку існує низка комерційних рішень для ефективного ведення садівництва, серед яких можна виділити супутникові, наземні та повітряні технології. Попри те, що безпілотні наземні транспортні засоби (БНТЗ) мають багато переваг таких, як можливість працювати тривалий час та перевозити корисне навантаження, вони мають і свої обмеження [68, 69]. По-перше, БНТЗ можуть бачити лише об'єкти, які знаходяться неподалік від землі або з точки зору, близької до землі, що зменшує кількість зібраної інформації. По-друге, БНТЗ не можуть пересуватись по складному рельєфу, що часто зустрічається на реальних

сільськогосподарських ділянках. Нарешті, БНТЗ не можуть швидко оглядати великі сільськогосподарські поля.

Останнім часом спостерігається значний прогрес у розробці БПЛА вагою менше, ніж 5 кг [70]. У цифровому садівництві найбільш поширеними є БПЛА малих розмірів, оскільки саме вони здатні здійснювати польоти між деревами та під кронами [41]. Такі БПЛА можуть швидко рухатися у тривимірному просторі та застосовуються в цифровому садівництві для визначення врожайності, внесення добрив та моніторингу посівів. Однак, більшість програм зосереджена на польотах у відкритому просторі над головою, що обмежує можливості вимірювання. Наприклад, неможливо оцінити розміри окремих плодів і стан здоров'я дерев або отримати вимірювання діаметрів лише на основі накладних даних.

Проблему можна вирішити застосуванням БПЛА, які можуть літати нависом над робочим середовищем. Такі БПЛА можуть досягти оптимального балансу між рівнем покриття та роздільною здатністю датчика, залишаючись при цьому економічно вигідними [71, 72]. Однак, розробка автономної системи БПЛА, яка може працювати при великому масштабі та на кількох висотах, між рядами дерев або навіть занурюватися під крону дерев, все ще є складною. По-перше, глобальна система позиціонування (GPS) не завжди є надійною через оклюзію кроною дерев [73]. По-друге, навколишнє середовище дуже неструктуроване та динамічне [74], що створює значні проблеми для роботизованих однометричних систем, які покладаються на статичні колекції геометричних характеристик, але переважно з людьми-пілотами. На рис. 1.3 Наведено приклади автономних польотів над різними робочими середовищами.

Мета робочої місії та характеристики середовища такі як масштаб, структурованість, ступінь захаращеності та можливість доступу до GPS [75], суттєво впливають на конструкцію кожного модуля БПЛА, що безпосередньо впливає на сенсорні та обчислювальні платформи. Стандартним обмеженням для систем з БПЛА є неможливість виконання місій на великій відстані (кілька кілометрів) [11, 69, 75] у складних сільськогосподарських середовищах, як показано на рис. 1.3.



Рисунок 1.3 – Приклад виконання автономних польотів БПЛА в робочому середовищі фруктовому саду [41]

Крім того, вони не можуть із заводським налаштуванням виявляти та ідентифікувати складні об'єкти такі як фрукти або дерева, у реальному часі та в масштабі, що є важливим для керування БПЛА планування більш інформативних траєкторій [76].

Датчики навігаційних супутникових систем GNSS [77] такі як GPS, дозволяють дронам отримувати геопросторове позиціонування [78]. Ці датчики є недорогими, а їх точність може досягати 1 метра. Однак, якщо між приймачем і супутником є перешкоди, точність GNSS знижується, що характерно для середовищ між деревами або під кронами. Удосконалення GNSS за допомогою наземних станцій таких як диференціальний GPS (DGPS) [79] і кінематика в реальному часі (RTK) [80], дозволяють підвищити точність до сантиметрового діапазону. При використанні поправок необхідна наявність радіозв'язку між БПЛА і базовою станцією. Методи постоброблення такі як постоброблена кінематика [81], можуть допомогти отримати точні вимірювання, коли немає доступного зв'язку між базовою станцією та дроном [82], але їх не можна використовувати для керування в реальному часі.

Менші БПЛА є більш безпечними, оскільки можуть пролітати через вузькі проміжки [83], мають вищу маневреність та легше монтувати. Попри наявність на сучасному ринку різноманітних невеликих та легких БПЛА, їхні технічні характеристики щодо автономного польоту, обчислювальної потужності та зберігання даних для виконання великомасштабних автономних місій в умовах відсутності GPS є обмеженими.

У табл. Г.2 додатку Г наведено короткий опис моделей БПЛА малого розміру та відповідні платформи [84-88], що зазвичай використовуються в помірно захищеному середовищі без присутності людини.

З огляду на результати аналізу технічних характеристик п'яти популярних моделей БПЛА, які наведені в табл. В.2, видно, що кожен із п'яти проаналізованих комерційних дронів має свої сильні та слабкі сторони для використання в фруктових садах. Дрони DJI Phantom 4 RTK та senseFly eBee<sup>x</sup> пропонують високоякісні зображення та картографічні можливості, тоді як Parrot Bluegrass Fields пропонує багатоспектральне зображення та 3D-картографування. DJI Agras T20 і Yamaha RMAX спеціалізуються на обприскуванні пестицидів і добрив, але мають обмежені функції за межами цієї програми. Вартість дрона також є важливим чинником, який варто враховувати в разі вибору рішення в саду, водночас деякі моделі суттєво дорожчі за інші.

Компанія DJI також забезпечує свої дрони технологією DJI GS PRO [88], яка призначена для подання вихідних даних робочого середовища фруктового саду у вигляді тривимірних GPS координат. В результаті проведеного аналізу встановлено, що БПЛА моделі DJI Phantom 4 RTK разом із технологією DJI GS PRO найкраще слугує для задачі виявлення та обчислення кількості плодів у фруктовому саді.

Насамкінець, для полегшення доступу до складних систем БПЛА у сільському господарстві потрібний інтуїтивно зрозумілий інтерфейс користувача, і абстрагування цієї складності для користувача – це складне завдання. Під час роботи у новому середовищі БПЛА, як правило, потрібні налаштування та конфігурування, що можуть бути здійснені досвідченими користувачами.

### 1.3.2 Керування робочими місіями групи безпілотних літальних апаратів

Використання інформаційних систем, які дістають та аналізують дані із БПЛА, стало на сьогодні стандартом для ведення успішного сільського господарства. Цифрове управління садами може бути вкрай успішним із

використанням БПЛА [89], оскільки вони надають детальну інформацію про стан та кількість плодів у садах [90], даючи змогу фермерам приймати більш обґрунтовані рішення щодо обрізки, удобрення та збору врожаю [91, 92].

Наразі дослідники та практики використовують різні алгоритми планування маршруту [93-95], які можуть бути застосовані до групи БПЛА у фруктовому саду. На сьогодні популярними алгоритмами побудови маршрутів БПЛА є алгоритм на основі сітки [96, 97], алгоритм на основі діаграми Вороного [98, 99] та алгоритм  $Q$ -навчання ( $Q$ -алгоритм) [100, 101].

Вирішення проблеми ідентифікації та обчислення фруктових плодів у режимі реального часу залежить від того, наскільки повно та швидко група БПЛА виконуватиме обліт усіх дерев фруктового саду. Водночас кожен фруктовий сад може мати різне планування з різним розташуванням дерев, різною висотою дерев та заздалегідь невідомим розташуванням інших фізичних перешкод. Як наслідок, є потреба у гнучкості до визначення траєкторії руху групи БПЛА. Тобто маршрут, за яким рухатиметься група, має бути унікальним для кожного фруктового саду. Тому, з огляду на вище вказані вимоги до вирішення проблеми, що розглядається,  $Q$ -алгоритм видається найкращим рішенням для побудови маршруту групи БПЛА в режимі реального часу.

$Q$ -алгоритм – це тип алгоритму навчання з підкріпленням [102], який використовується для вирішення проблем оптимізації, зокрема для побудови квазіоптимальних маршрутів для груп БПЛА [103]. Основна ідея  $Q$ -алгоритму полягає в тому, щоб вивчити кроки дій, послідовне виконання яких максимізує очікувану накопичувальну винагороду. У контексті планування маршруту БПЛА  $Q$ -алгоритм працює таким чином.

1. Визначається простір станів, що відповідає робочому середовищу, у якому працює група БПЛА. Простір станів містить інформацію про поточне положення БПЛА, положення цілі, положення перешкод у середовищі тощо.

2. Визначається простір дій, що відповідає набору можливих операцій, які БПЛА можуть виконувати, перебуваючи в кожному стані. У разі планування маршруту діями можуть бути рух у певному напрямку або зміна висоти.

3. Визначається функція винагороди, що слугує критерієм якості побудованого маршруту в поточний момент виконання алгоритму. У випадку планування маршруту БПЛА функція винагороди може бути формалізована так, щоб заохочувати ефективні й безпечні маршрути, та одночасно штрафувати зіткнення, відхилення від бажаного шляху або інші небажані результати.

4. Ініціалізується  $Q$ -таблиця, що подається у вигляді матриці, яка зберігає очікувану винагороду для кожної пари стан-дія. На початку виконання алгоритму  $Q$ -таблиця заповнюється випадковими значеннями.

5. Наступні кроки виконуються до настання зближення:

5.1. Оцінюється попередній стан довкілля.

5.2. Обирається дія на основі попереднього стану та значень у  $Q$ -таблиці, використовуючи  $\varepsilon$ -greedy [104] або softmax [105].

5.3. Виконується обрана дія та оцінюється отриманий стан і винагорода.

5.4. Оновлюється  $Q$ -таблиця на основі отриманих значень стану  $S_{t-1}$  та дії  $A_{t-1}$ , за формулою:

$$Q(S_t, A_t) = Q(S_{t-1}, A_{t-1}) + \alpha \left( r_{t-1} + \gamma \cdot \max_A \{Q(S_t, A)\} - Q(S_{t-1}, A_{t-1}) \right), \quad (1.12)$$

де  $Q(S_{t-1}, A_{t-1})$ ,  $1, \dots, T$ , – кількісне вираження винагороди, яку отримав інтелектуальний агент  $Q$ -алгоритму за досягнення попереднього стану  $S_{t-1}$ ,  $T$  – загальний час роботи інтелектуального агента,  $\alpha$  – коефіцієнт швидкості навчання моделі,  $0 < \alpha \leq 1$ ,  $r_t$  – рівень винагороди, яку отримує інтелектуальний агент у разі переходу зі стану  $S_{t-1}$  до стану  $S_t$ ,  $\gamma$  – коефіцієнт знецінювання, який визначає важливість майбутніх винагород, здобутих інтелектуальним агентом,  $0 \leq \gamma \leq 1$ ,  $\max_A \{Q(S_t, A)\}$  – оціночне кількісне значення майбутньої винагороди у разі виконання дії  $A$ , перебуваючи в стані  $S_t$ .

5.5. Попередній стан переводиться в поточний.

6. Розраховані значення за формулою (1.12) записуються у  $Q$ -таблицю та використовуються для планування маршрутів для БПЛА.

Завдяки цьому ітераційному процесу оновлення  $Q$ -таблиці  $Q$ -алгоритм оптимізує очікувану накопичувальну винагороду за послідовністю дій, якою є пошук квазіоптимального маршруту для групи БПЛА.

### 1.3.3 Використання групи безпілотних літальних апаратів для генерації фото- та відеоматеріалів

Виявлення об'єктів та семантична сегментація є критичними для цифрового садівництва, оскільки корисна інформація зазвичай пов'язана з семантичними характеристиками такими як фрукти, дерева чи дані фенотипу.

Для сільськогосподарських застосувань таких як класифікація зображень, виявлення аномалій та оцінка врожайності, використовуються різні методи отримання зображень, зокрема RGB [93], мультиспектральні [107] та гіперспектральні [108], теплові зображення [109] та зображення в ближньому інфрачервоному діапазоні [110].

Подання фруктового саду у вигляді робочого середовища в тривимірному просторі використовується для виявлення об'єктів та включає воксельну сітку [111-113], хмару точок, мультиракурс та сферичне зображення. Об'ємні ЗНМ подають дані як тривимірну сітку зайнятості та безпосередньо застосовують тривимірну згортку до цього подання [114]. Підходи, засновані на мультипроекції, проєктують тривимірні дані з різних кутів огляду на двовимірне зображення [115], яке потім можна передати в ГЗНМ на основі зображень. Однак процес проєктування чутливий до шумів або неповного введення, а процес вокселізації викликає втрату інформації. З цієї причини деякі стверджують, що краще використовувати хмари точок.

Проблема оклюзії серйозно ускладнює оцінку врожайності та семантичне відображення у садах [116]. Для врахування різниці між оціненою та фактичною врожайністю можна використовувати прості статистичні інструменти такі як

лінійна регресія. Однак ці процедури калібрування обмежені, оскільки вимагають наявності даних про врожайність попереднього врожаю. Щодо рішень для боротьби зі сильними оклюзіями, можна використовувати агрегацію інформації з кількох видів і кількох датчиків або застосовувати БПЛА для ефективного виявлення закритих об'єктів.

Якість згенерованих фотоматеріалів зазвичай оцінюють за індексом структурної подібності (SSIM) та індексом пікового співвідношення візуального сигналу до шуму (PSNR) [117].

Індекс SSIM формалізовано так

$$SSIM = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \quad (1.13)$$

де  $\mu_x$  та  $\mu_y$  – середні значення пікселів у вхідному та об'єднаному зображеннях, відповідно,  $\sigma_x^2$  та  $\sigma_y^2$  – стандартні відхилення пікселів у вхідному та об'єднаному зображеннях, відповідно,  $\sigma_{xy}$  – коваріація між пікселями в обох зображеннях, а  $C_1$  та  $C_2$  – константні значення, які дозволяють стабілізувати результуюче значення формули.

Індекс PSNR формалізовано за формулою

$$PSNR = 10 \log_{10} \left( \frac{MAX^2}{MSE} \right), \quad (1.14)$$

де  $MAX$  – максимальне значення пікселя в зображенні, а  $MSE$  – середньоквадратична похибка між вихідним зображенням та об'єднаним зображенням.

Для проведення збору зображень з використанням БПЛА для виявлення заданих об'єктів можуть, також, використовуватись групи БПЛА. Поєднання декількох БПЛА в групи і організація їх взаємодії представлені в роботі [118] та доведено ефективність такого рішення. Функціонування БПЛА в складі систем може



використовуватись в різних галузях та сферах [119]. Подальше розпізнавання та обробка зображень, отриманих з використанням БПЛА, для виявлення заданих об'єктів можуть розв'язуватись як окрема задача з використанням методів ШІ [120].

З проведеного огляду встановлено, що БПЛА стають все більш популярними в садівництві, де вони можуть бути використані для розв'язання різних задач таких як моніторинг рослин, картографування саду та розпилювання пестицидів. В цьому огляді було розглянуто різні аспекти використання БПЛА в садівництві, зокрема технічні можливості БПЛА, типи датчиків, що можуть використовуватися, а також програмне забезпечення для управління БПЛА.

Отже, для підвищення точності визначення кількості яблук, можна використовувати методи машинного навчання, які дозволяють БПЛА самостійно вчитися розрізняти яблука від інших об'єктів та розраховувати їх кількість на знімку. Застосування цих методів може значно збільшити ефективність використання БПЛА для виявлення яблук у фруктовому саду. Отже, описані вище проблеми можуть бути вирішені через розроблення та впровадження нових методів та ефективних технологій безпілотної авіації в робочому середовищі фруктового саду.

#### 1.4. Постановка задачі

Вирішення завдань збору, розпізнавання та обробки зображень для виявлення заданих об'єктів в просторі потребують здійснення розробки методів збору, розпізнавання та обробки зображень, отриманих з використанням БПЛА, і розв'язання таких наукових задач:

1. Провести аналіз та порівняння відомих методів збору, розпізнавання та обробки зображень, які отримані з використанням БПЛА, а також можливості створення інформаційних систем такого призначення, результатом роботи яких був би облік заданих об'єктів.

1. Провести аналіз методів, засобів та систем збору, розпізнавання та обробки зображень отриманих з використанням БПЛА для виявлення заданих об'єктів.

2. Розробити новий метод побудови маршрутів БПЛА згідно технологій самонавчання, щоб покращити переміщення та синхронізацію між групою БПЛА або одного БПЛА в межах робочого сегменту і за рахунок цього збільшити кількість опрацьованих даних.

3. Розробити новий метод динамічного отримання зображень заданих структурних об'єктів в тривимірному просторі за допомогою декількох БПЛА для узгодження між різними БПЛА та досягнення переміщення всієї групи БПЛА від заданих початкових до кінцевих точок програмної місії автономно.

4. Розробити новий метод синхронізації відеопотоків в режимі реального часу для забезпечення оперативного отримання результатів та здійснення виявлення структурних об'єктів, які були пропущені в процесі минулих програмних місій.

5. Вдосконалити метод виявлення заданих структурних об'єктів на зображеннях для покращення точності виявлення і зменшення часу навчання нейронної мережі.

6. Розробити новий спосіб отримання кількості заданих структурних об'єктів із отриманих зображень за допомогою групи БПЛА.

7. Розробити автоматизовану систему з групою БПЛА з використанням алгоритмів комп'ютерного зору, машинного навчання та оптимізації, здійснити постановку експерименту та провести з нею експериментальні дослідження щодо встановлення покращення точності виявлення заданих об'єктів на отриманих зображеннях та впровадити її у виробництво.

### 1.5. Висновки до першого розділу

Дослідження методів збору, розпізнавання та обробки зображень, отриманих з використанням БПЛА, показало наступні результати:

1. В сучасному світі застосування безпілотних літальних апаратів (БПЛА) для розв'язання різних задач є все більш популярним. Однією з таких задач, в якій можуть бути використані БПЛА, є застосування їх в складі інформаційної системи для виявлення, збору, розпізнавання та обчислення кількості яблук у фруктовому

саду. Цей процес може бути пов'язаний з високим ризиком втрати часу та ресурсів, особливо у великих садах. Використання БПЛА може допомогти автоматизувати цей процес та покращити ефективність роботи.

2. Однією з найбільших проблем, пов'язаних з використанням БПЛА для розпізнавання яблук, є складність визначення оптимальних параметрів польоту та налаштування обладнання. Це може вимагати багато часу та зусиль від операторів, що може призвести до зниження продуктивності та збільшення витрат. Також, іншою важливою проблемою є обмежена продуктивність та ефективність БПЛА у реальних умовах фруктових садів. Наприклад, використання традиційних камер з високою роздільною здатністю може призвести до складнощів з обробкою великої кількості даних, які потрібно збирати та аналізувати.

3. Щоб вирішити ці проблеми, можна застосувати різні підходи та технології. Наприклад, можна використовувати спеціальні алгоритми машинного навчання для підвищення точності розпізнавання яблук, що дозволить зменшити кількість помилок та скоротити час, необхідний для обробки даних. Також, можна використовувати новітні технології обробки даних такі як штучний інтелект та глибоке навчання, щоб автоматизувати процеси збору та аналізу даних, що зменшить навантаження на операторів та збільшить ефективність.

4. Перспективним напрямом згідно проведеного дослідження предметної області є розробка методів та інформаційної системи збору, розпізнавання та обробки зображень, отриманих з використанням БПЛА. В складі інформаційної системи може бути декілька БПЛА. Крім того, для отримання результату з обчислення кількості яблук потрібно забезпечити розпізнавання зображень системою з використанням методів ШІ та ГН.

Основні результати розділу опубліковані у працях [121-127].

## РОЗДІЛ 2.

МЕТОДИ ТА СИСТЕМА ДИНАМІЧНОГО ОТРИМАННЯ ЗОБРАЖЕНЬ  
ЗАДАНИХ ОБ'ЄКТІВ В ТРИВИМІРНОМУ ПРОСТОРИ

Отримання зображення в тривимірному просторі в динамічному режимі з подальшою його обробкою в контексті розпізнавання заданих структурних об'єктів є актуальним завданням, оскільки потребує забезпечення не тільки певної точності результату розпізнавання, але першочергово забезпечення коректного повного визначення зображення. Тому, для розв'язування такого наукового завдання необхідним є розробка всього процесу та першочергово методів динамічного отримання зображень структурних об'єктів однієї природи в тривимірному просторі. Враховуючи обсяг поставлених завдань щодо зображень необхідним є визначення траєкторій руху залучених засобів їх збору та отримання. Автоматизація таких завдань покращить економічний ефект від її впровадження.

Враховуючи потребу поєднання різних технічних засобів та реалізацію різноспрямованих методів і алгоритмів для забезпечення отримання результату, необхідно створити систему, в якій би поєдналися ці засоби, реалізовані методи та алгоритми. Оскільки така система буде розподіленою в просторі, бо збір інформації про об'єкти та центр прийняття рішень можуть бути суттєво віддалені, а також враховуючи, що така система може керуватись не фаховим спеціалістом, а звичайним користувачем, то бажано, щоб вона була автоматизована в частині виконання основного завдання.

Тому, перспективним напрямом для розв'язання такого наукового завдання є розробка методів та засобів, які ґрунтуються на автоматизації та які потрібно впроваджувати в автоматизовану систему. В такій системі здійснюватиметься керування одним або декількома БПЛА, які будуть отримувати зображення об'єктів, враховуючи один об'єкт одноразово.

## 2.1. Автоматизована система динамічного отримання зображень заданих об'єктів в тривимірному просторі

Поставлені завдання щодо отримання зображень структурних об'єктів в тривимірному просторі, враховуючи плановано використовувані засоби, потребують розробки системи, яка б поєднала різні за функційним призначенням засоби. Специфіка завдань потребуватиме використання та керування засобами, які будуть територіально-розподіленими. Підтримка комунікації здійснюватиметься із залученням відповідних інформаційно-комунікаційних засобів. Також, важливим елементом, який потрібно розробити в системі, є забезпечення керування всіма компонентами через здійснення автоматизації. Така потреба зумовлена орієнтацією на використання системи побутовими користувачами. Як наслідок, виникає задачу спроектувати систему  $S$  так, щоб вона максимально автоматизувала роботу.

Зображення компонентів проєктованої автоматизованої системи  $S$  в загальній архітектурі та в розрізі різнотипності її компонентів подано основними складовими частинами системи на рис. 2.1. Наявність різнотипних компонентів системи та їх розподілення суттєво впливатимуть на функціонування, тому важливими завданнями при проєктуванні системи є забезпечення підтримки її цілісності в процесі функціонування та досягнення ефективності при використанні.

Для проєктування автоматизованої системи  $S$  попередньо визначимо також, об'єкти на які вона буде спрямована. Першочергово такими об'єктами є об'ємні області в досліджуваній частині тривимірного простору, де будуть отримуватись зображення. Задамо усю область досліджуваного простору, в якому будуть отримуватись структурні об'єкти однієї природи, координатами його початкової точки та трьома векторами, що відображатимуть сторони паралелепіпеда. Досліджувану область простору позначимо і формалізовано задамо так:

$$V = \langle P(x_1, x_2, x_3), V_1(v_{1,1}, v_{1,2}, v_{1,3}), V_1(v_{2,1}, v_{2,2}, v_{2,3}), V_1(v_{3,1}, v_{3,2}, v_{3,3}) \rangle, \quad (2.1)$$

де  $P(x_1, x_2, x_3)$  – початкова точка досліджуваної ділянки простору з координатами  $(x_1, x_2, x_3)$ ,  $V_i(v_{i,1}, v_{i,2}, v_{i,3})$  –  $i$  – вектор в просторі,  $i = 1, 2, 3$ .

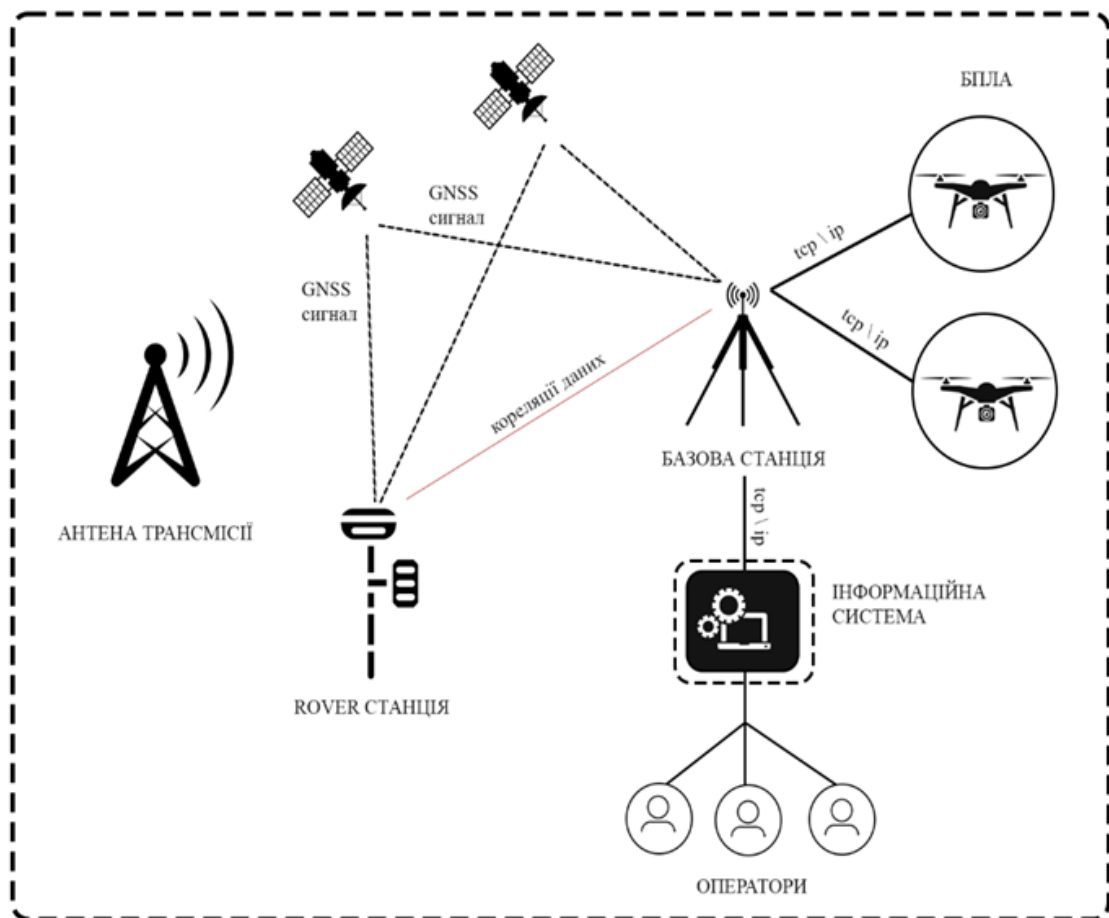


Рисунок 2.1 – Загальна архітектура системи  $S$  згідно її компонентів

Таким чином, вектор задаватиме в проєктованій системі  $S$  область досліджуваного простору (2.1) і визначимо його так:

$$v = (x_1, x_2, x_3, v_{1,1}, v_{1,2}, v_{1,3}, v_{2,1}, v_{2,2}, v_{2,3}, v_{3,1}, v_{3,2}, v_{3,3}). \quad (2.2)$$

Оскільки, крім загальної області, що подана формулою (2.1), на яку будуть спрямовані засоби отримання зображень, є підобласті, в яких будуть зосереджені підмножини структурних об'єктів безпосередньо, то такі підобласті теж визначатимуться наборами координат, які будуть в межах загальної області. Визначення початкових координат та координат векторів підобластей

задаватимемо лінійною матрицею векторів, яку можна задати також, прямокутною матрицею. Поділ області, що заданий формулою (2.1) необхідний, тому що підмножини структурних об'єктів не матимуть перетинів і, тому, їх можна досліджувати окремо кожен. Крім того, це надає змогу масштабувати результати роботи. А також, в кожній підмножині структурних об'єктів є їх близькість та позиціонування. Таким чином, задамо лінійною матрицею векторів підмножини структурних об'єктів в розглядуваній області простору так:

$$M_v = (v_1, v_2, \dots, v_{N_v}), \quad (2.3)$$

де  $M_v$  – лінійна матриця векторів підмножини структурних об'єктів,  $v_i$  – вектор  $i$ -підмножини структурних об'єктів,  $i = 1, \dots, N_v$ ,  $N_v$  – кількість підмножин в розглядуваній області простору.

При здійсненні деталізації значень векторів за формулою (2.2) матрицю координат векторів підмножини структурних об'єктів в розглядуваній області простору (2.1) задамо так:

$$M_{k_v} = \begin{bmatrix} v_{1,x_1} & \cdots & v_{N_v,x_1} \\ \vdots & \ddots & \vdots \\ v_{1,v_{3,3}} & \cdots & v_{N_v,v_{3,3}} \end{bmatrix}, \quad (2.4)$$

де  $v_{1,x_1} - v_{1,v_{3,3}}$  – 12 координат першого вектору  $v_1$ , згідно формули (2.2), аналогічно для решти  $N_v-1$  векторів.

Набір координат з формули (2.4) поступатиме на вхід системи  $S$ .

Проектована система  $S$  буде розподіленою, оскільки потребуватиме збору інформації в певній області простору. Зображення її архітектури задано на рис. 2.2. В ній виділено зовнішній та внутрішній рівні. До зовнішнього рівня віднесено рівень користувацького інтерфейсу та рівень центральної архітектури системи. До внутрішнього рівня віднесено програмні модулі та інтерфейси між ними.

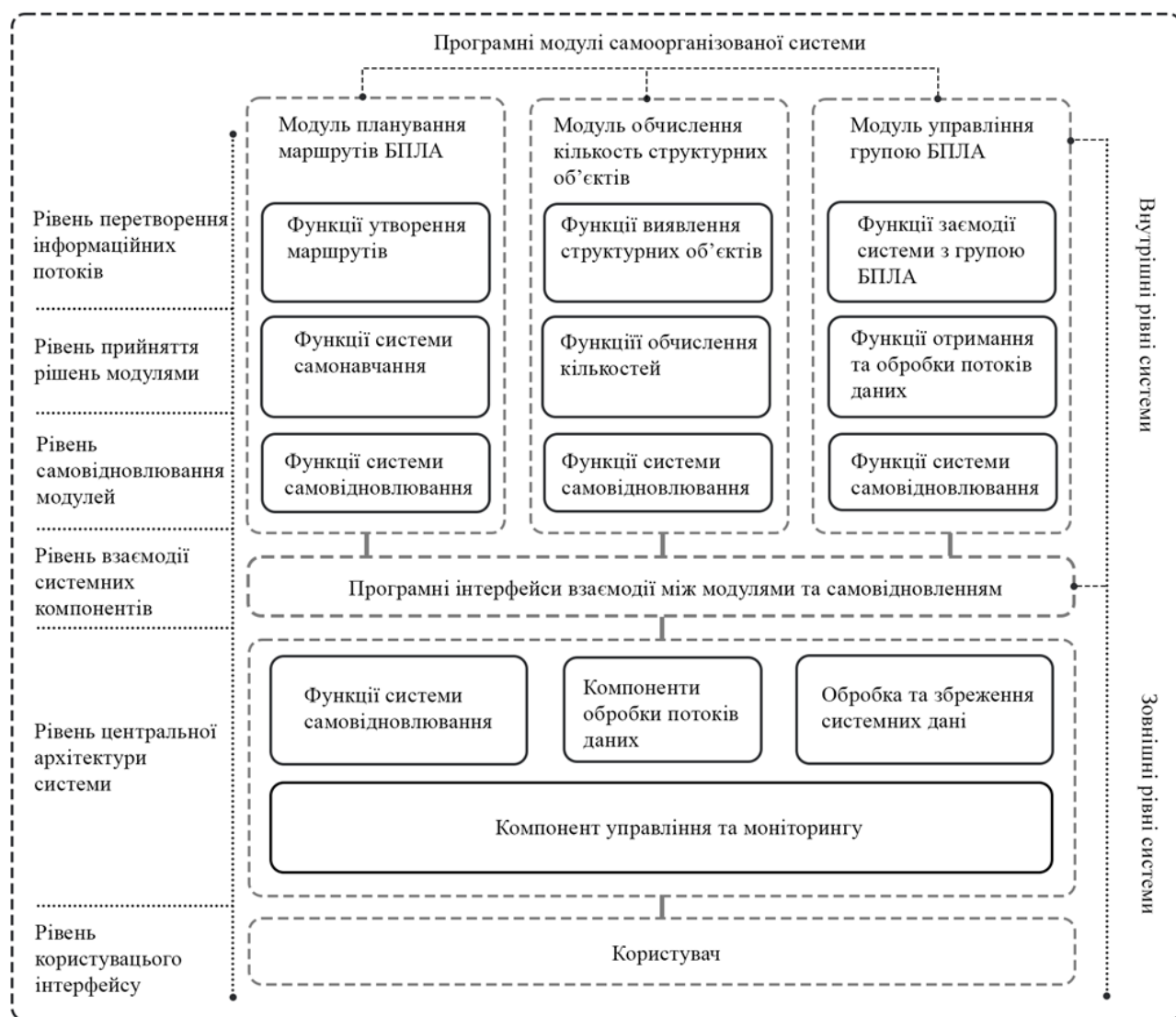


Рисунок 2.2 – Архітектура автоматизованої системи S

Компонентами системи будуть БПЛА, комп'ютер, різнофункційні датчики та засоби обслуговування БПЛА в автоматизованому або в автоматичному режимі. Різнофункційні датчики можуть поділятися на декілька підмножин в залежності від їх функційного призначення та використання.

Структурні компоненти автоматизованої системи можуть перебувати у різних станах. На основі матриць станів, що відображають стани компонентів автоматизованої системи, розроблено підсистему для активного моніторингу системних подій та узгодженої взаємодії компонентів системи при прийнятті нею рішень. Такий підхід дає можливість оператору системи втручатися у корегування поведінки окремих компонентів.



Система  $S$  повинна мати певні чітко визначені функційні можливості, які потребують виділення в контексті її розподілення. Задамо функційні можливості проєктованої системи елементами множини  $M_{f,S} = \{m_{f,S,1}, m_{f,S,2}, \dots, m_{f,S,n_{f,S}}\}$ , де  $n_{f,S}$  – кількість функційних можливостей проєктованої автоматизованої системи. Кожен елемент множини  $M_{f,S}$  забезпечується відповідним модулем. Міжмодульну взаємодію в системі задамо графом  $G_{f,S}$ , тоді автоматизована система  $S$  буде однозначно визначатись її елементами та заданими зв'язками між ними і задамо її так:

$$S_v = \langle M_{f,S}, G_{f,S} \rangle, \quad (2.5)$$

де  $v$  в позначенні  $S_v$  для задання системи  $S$  означає її задання на рівні архітектури згідно рис. 2.2.

Автоматизована система  $S$  повинна мати наступні функційні можливості:

1. Формування групи БПЛА.
2. Утворювати тривимірний програмний простір та встановлювати дозволені та заборонені сегменти.
3. Встановлювати та корегувати початкові точки обльоту.
4. Розпочинати обліт в робочому середовищі досліджуваної області досліджуваного простору  $V$ .
5. Управління вихідними даними від групи БПЛА.
6. Змінювати стани групи та окремого БПЛА.

Таке задання системи  $S$  згідно формули (2.5) відображає елементи та процеси між ними в середині системи. При цьому врахування різнотипних компонентів автоматизованої системи  $S$  та вхідної інформації згідно формули (2.4) не враховується. Для врахування цих складових частин задамо проєктовану систему  $S$  так:

$$S_z = \langle M_{z,S,1}, M_{z,S,2}, \dots, M_{z,S,n_{f,S}} \rangle, \quad (2.6)$$

де  $n_{f,S}$  – кількість множин, які задають різнотипні компоненти системи,  $M_{z,S,i}$  – множина, елементами якої є однотипні компоненти, що формують систему  $S$ ,  $i = 1, 2, \dots, n_{f,S}$ ,  $z$  в позначенні  $S_z$  для задання системи  $S$  означає її задання на рівні архітектури, яка сформована з фізичних компонентів.

Результатом роботи автоматизованої системи  $S$  (2.6) буде зібрана інформація зовнішніми давачами та оброблена згідно неї цільова інформація. Враховуючи, що згідно формули (2.4), робоча область буде поділена на підобласті, то інформацію зібрану зовнішніми давачами задамо матрицею так:

$$M_{D_k} = \begin{bmatrix} d_{1,d_1} & \cdots & d_{N_d,d_1} \\ \vdots & \ddots & \vdots \\ d_{1,d_n} & \cdots & d_{N_d,d_n} \end{bmatrix}, \quad (2.7)$$

де  $d_{i,d_j}$  – елемент матриці  $M_{D_k}$ , який є вектором і містить інформацію зібрану давачами про  $(i, d_j)$  підобласть,  $i = 1, 2, \dots, N_d$ ,  $N_d$  та  $d_n$  – два дискретні значення сторін прямокутника, яким визначається досліджувана частина простору (робоча область),  $j = 1, 2, \dots, n$ ,  $n$  – кількість  $d_n$ .

Інформація, яку отримано із зовнішніх давачів, потрібно опрацювати у відповідному модулі системи і результати цього опрацювання будуть використані безпосередньо для отримання цільового результату. Тому, задамо функцію  $F_{D_k}$ , яка згідно кожного з елементів матриці  $M_{D_k}$  (2.7) буде видавати результат опрацювання інформації з певної підобласті. Тоді, цільову інформацію обчислюватимемо так:

$$R_{D_k} = \sum_{i=1}^{N_d} \sum_{j=1}^{d_n} F_{D_k}(d_{i,d_j}), \quad (2.8)$$

де  $F_{D_k}$  – функція, яка видає результат опрацювання інформації з певної підобласті  $(i, d_j)$ ,  $i = 1, 2, \dots, N_d$ ,  $N_d$  та  $d_n$  – два дискретні значення сторін прямокутника, яким

визначається досліджувана частина простору (робоча область),  $j = 1, 2, \dots, n$ ,  $n$  – кількість  $d_n$ .

Таким чином, автоматизована система  $S$  буде визначатись згідно формул (2.5) та (2.6) в повному обсязі на різних рівнях її архітектури. Необхідність такого задання системи  $S$  вимагалось для повного опису, як складових компонентів так і для опису взаємодій між модулями системи. Система  $S$  буде розподіленою, тому її модулі описані за формулою (2.5) будуть в фізичних компонентах, які задані формулою (2.6). Архітектурно елементи системи, які задані формулою (2.5) будуть здійснювати керування компонентами, які задані формулою (2.6).

Згідно формул (2.4-2.8), які необхідні для однозначного задання автоматизованої системи  $S$ , враховуючи різноманітність компонент системи та зв'язків між ними і між модулями, розроблена архітектура системи на рівні модулів задає наявність певних її модулів в компонентах системи, які описані в архітектурі системи на рівні компонент. Тобто, в компонентах системи будуть перебувати модулі системи. Таким чином, це поєднання дозволяє задати архітектуру такої системи з врахуванням всіх можливих особливостей, включаючи варіанти з масштабуванням кількості компонент за рахунок багатократного використання модулів. Формулами (2.4-2.8) задано архітектуру системи, формат вхідної інформації для неї і результуючу інформацію. Подальші кроки в проектуванні системи  $S$  потребуватимуть деталізації складових частин, які задані формулами (2.4-2.8).

Архітектуру автоматизованої системи  $S$  на рівні модулів та взаємодій між ним зображено на рис. 2.3. У цій схемі архітектури наявні всі складові частини формул (2.4-2.8) включно з формуванням цільового результату. Архітектура автоматизованої системи містить наступні складові.

1. Функції підготовки системи до роботи, а саме: формування групи БПЛА, приєднання всіх пристроїв до комп'ютерної мережі та переміщення пристроїв до початкової точки робочого середовища.

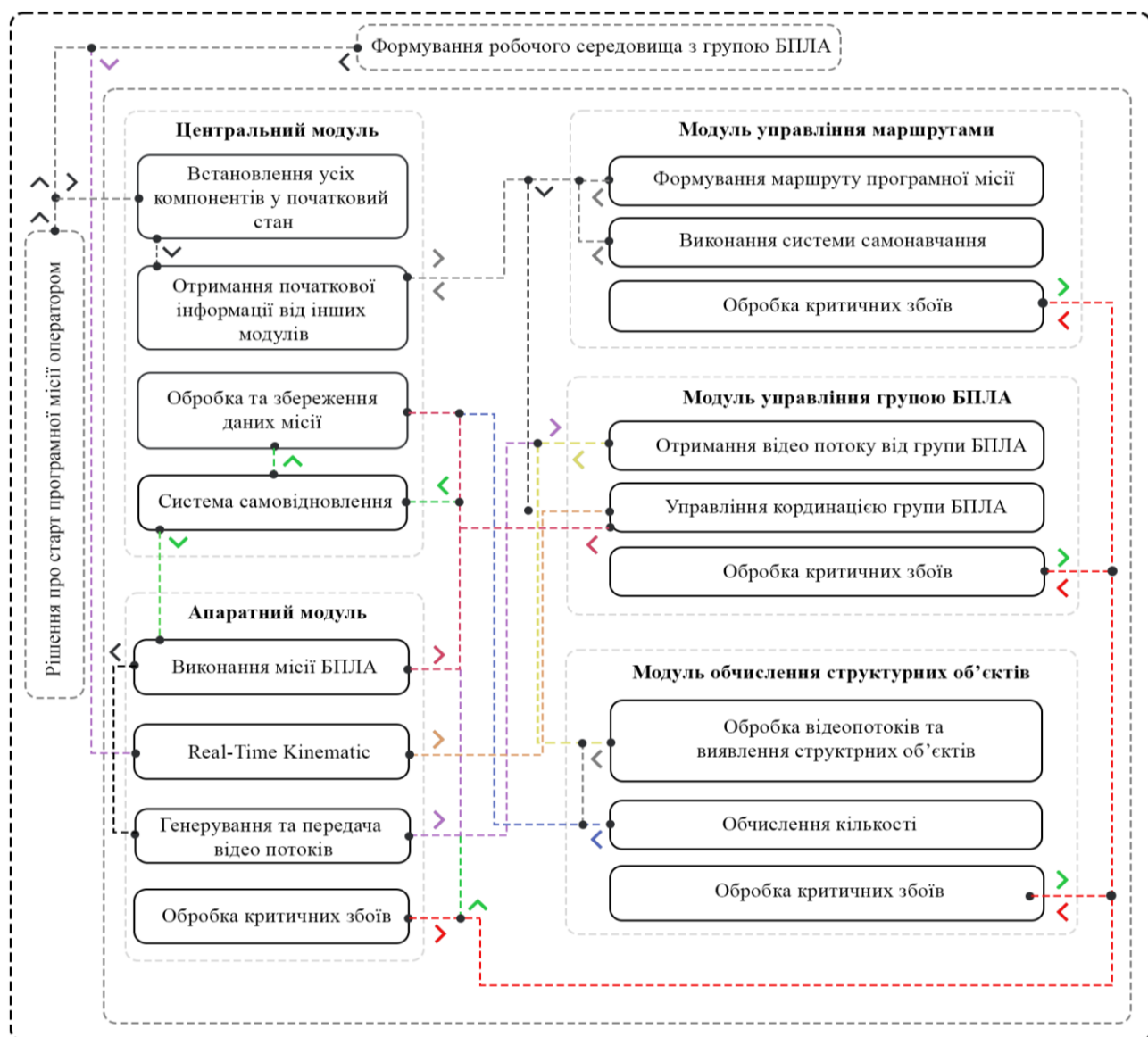


Рисунок 2.3 – Архітектура автоматизованої системи  $S$  на рівні модулів та взаємодій

2. Центральний модуль, що є основою архітектури автоматизованої системи та забезпечує перехід між різними визначеними рівнями архітектури на основі інформації, отриманої з інших рівнів автоматизованої системи. Основними функціями центрального модуля є:

- встановлення всіх компонентів у початковий стан;
- отримання початкової інформації від інших модулів;
- обробка та збереження даних місії;
- самовідновлення системи.

3. Модуль управління маршрутами, що слугує для автоматичного формування та коригування маршруту групи БПЛА під час виконання ними робочої місії. Основними функціями модуля управління маршрутами є:

- формування маршруту програмної місії;
- виконання системи самонавчання;
- обробка критичних збоїв.

4. Модуль управління групою БПЛА, що призначений для коригування координації польоту БПЛА у робочому середовищі в режимі реального часу. Основними функціями модуля управління групою БПЛА є:

- отримання відеопотоку від групи БПЛА;
- управління координацією групи БПЛА;
- обробка критичних збоїв;

5. Апаратний модуль, що виконує взаємодію між модулями автоматизованої системи та зовнішнім апаратним забезпеченням. Основними функціями апаратного модуля є:

- виконання місії БПЛА;
- отримання GPS-координат для побудови маршрутів;
- генерування та передача відеопотоків;
- обробка критичних збоїв.

6. Модуль обчислення структурних об'єктів, що призначений для обчислення кількості структурних об'єктів у робочому середовищі в режимі реального часу. Основними функціями модуля обчислення структурних об'єктів є:

- обробка відеопотоків та виявлення структурних об'єктів;
- обчислення кількості виявлених структурних об'єктів;
- обробка критичних збоїв.

Таким чином, спроектована архітектура автоматизованої системи задає вимоги до необхідних для її функціонування елементів та компонентів, зв'язки між ними, можливість масштабування. Для досягнення належного рівня автоматизації при визначенні подальших кроків функціонування підсистем та компонентів її складові частини та внутрішні засоби автоматизації потребують деталізації.

## 2.2. Проектування робочої місії БПЛА в тривимірному просторі

В основі використання автоматизованої системи закладено робочу місію БПЛА, що призначена для виконання визначених цілей та завдань у режимі реального часу. Успішне виконання робочої місії потребує її проектування заздалегідь. Нижче наведені основні етапи проектування робочої місії БПЛА в тривимірному просторі:

1. Визначення початкових та кінцевих координат робочої місії. Оператор автоматизованої системи визначає початкову та кінцеву точки руху БПЛА в робочому середовищі.

2. Встановлення технічних характеристик БПЛА. Вибір моделей БПЛА потребує врахування їхніх технічних характеристик, зокрема, швидкість руху, радіус дії, висота й ширина охоплення камери.

3. Визначення маршруту. Необхідно розробити маршрут польоту БПЛА, з огляду на характеристики робочого середовища. Також, потрібно врахувати можливі перешкоди на маршруті, щоб забезпечити його автоматичне прокладання надалі.

4. Ініціалізація значень параметрів технічного устаткування. Необхідно вибрати та встановити технічне устаткування для збору й передачі даних такі як сенсори, RTK модулі, мережеве обладнання та інші.

5. Перевірка та тестування. Перед запуском БПЛА з початкової точки необхідно виконати тестування всіх систем та компонентів автоматизованої системи для перевірки їхньої працездатності та правильності функціонування.

Головною особливістю функціонування всієї робочої місії є визначення та утворення маршруту руху БПЛА в робочому середовищі. Робоча місія виконується в складному робочому середовищі, де під час її протікання можуть виникати різноманітні сторонні чинники та події, що вноситимуть корективи в маршрут БПЛА. До таких чинників варто віднести зміну погодних умов, сильні пориви вітру, критичний заряд батареї, поява несподіваних перешкод тощо. Тому, з огляду на вищевказані проблеми, актуальною є потреба у розробленні нового метода

побудови маршрутів БПЛА в тривимірному координатному просторі, реалізація якого даватиме змогу автоматично відновлювати та корегувати координати руху групи БПЛА після критичних відхилень та забезпечуватиме самонавчання автоматизованої системи під час виконання програмної місії. Побудований у такий спосіб маршрут зможе використовуватись у подальших програмних місіях, що дасть змогу зменшити навантаження на оператора автоматизованої системи під час формування маршрутів.

У результаті аналізу типового робочого середовища фруктового саду (див. п. 1.3.1) визначено три можливих траєкторії руху групи БПЛА в робочому середовищі: (I) пріоритетна траєкторія руху, (II) допустима та (III) небажана. Їхнє схематичне подання ілюстровано на рис 2.4.

Розглянемо вибір саме пріоритетної траєкторії руху БПЛА з рис. 2.4 порівняно з рештою варіантів за визначеним критерієм.

Вибір траєкторії обльоту робочих сегментів фруктових дерев в експериментальному робочому середовищі обумовлений, насамперед, часом  $T$ , за який група БПЛА здійснює виконання робочої місії в робочому середовищі. Час на виконання робочої місії залежить від швидкості руху  $v$  кожного апаратного пристрою в групі БПЛА. Зазначимо, що  $v$  є постійною величиною для кожного БПЛА в групі за будь-якої траєкторії руху. Тоді, час що потрібен для обльоту робочого середовища, подамо через відношення відстані  $D$  і швидкості руху  $v$  за формулою:

$$T = \frac{D}{v}. \quad (2.9)$$

Задамо визначення згідно формули (2.9) для кожної з трьох визначених траєкторій. Позначимо через  $d_1$  відстань між сусідніми робочими сегментами в ряді робочого середовища. Величина  $d_1$  є сталою величиною для всіх траєкторій і вказує на те, що відстань між робочими сегментами є однаковою. Позначимо через  $d_2$  відстань між рядами робочих сегментів у експериментальному робочому

середовищі. Величина  $d_2$  є також сталою величиною і визначає однакову відстань між рядами робочих сегментів. Нехай  $n$  – кількість робочих сегментів в одному ряду,  $m$  – кількість рядів у робочому середовищі, яку потрібно облетіти під час виконання програмної місії, а  $D^I$ ,  $D^{II}$  та  $D^{III}$  – відстані в метрах, які потрібно пролетіти групі БПЛА за траєкторіями руху (I), (II) та (III), відповідно.

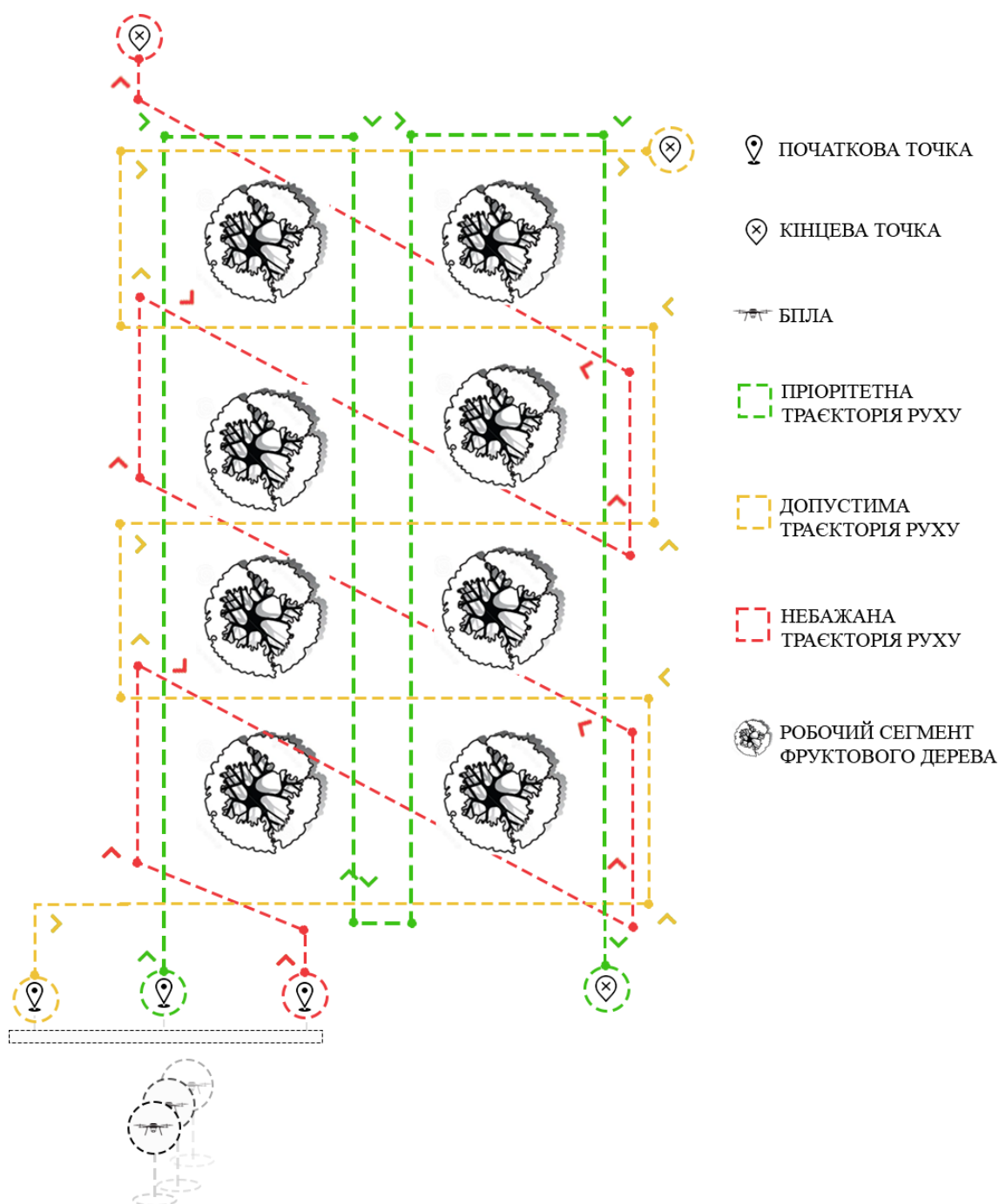


Рисунок 2.4 – Схема можливих траєкторії руху групи БПЛА у робочому середовищі



Тоді, час обльоту у визначеній пріоритетній траєкторії (I) визначимо за наступною формулою:

$$T_1 = \frac{D^I}{v} = \frac{(n \times m \times d_1 + (m - 1) \times d_1)}{v}. \quad (2.10)$$

Час обльоту у допустимій траєкторії руху (II) є таким:

$$T_2 = \frac{D^{II}}{v} = \frac{(n \times m \times (d_1 + d_2) - (m - 1) \times d_2)}{v}. \quad (2.11)$$

I, нарешті, час обльоту у небажаній траєкторії руху (III) подамо так:

$$T_3 = \frac{D^{III}}{v} = \frac{\sqrt{(d_1^2 + d_2^2)} \times n \times m}{v}. \quad (2.12)$$

Параметри  $d_1$ ,  $d_2$ ,  $n$ ,  $m$  та  $m$  використовуються у формулах (2.10-2.12) для обчислення відстані та часу для кожної можливої траєкторії руху групою БПЛА. Значення цих параметрів є сталими для кожної траєкторії руху в процесі обльоту групою БПЛА. Тому, щоб формально обґрунтувати відмінності між траєкторіями за часом обльоту, обчислимо різницю між  $T_1$  та  $T_2$ , а також різницю між  $T_2$  та  $T_3$ .

Різницю  $\Delta T_{1,2}$  між  $T_1$  та  $T_2$  визначимо так

$$\begin{aligned} \Delta T_{1,2} = T_1 - T_2 &= \frac{D^I}{v} - \frac{D^{II}}{v} \\ &= \frac{(n \times m \times d_1 + (m - 1) \times d_2)}{v} \\ &\quad - \frac{(n \times m \times (d_1 + d_2) - (m - 1) \times d_2)}{v} \\ &= \frac{(n \times m \times d_1 - n \times m \times d_2)}{v}. \end{aligned} \quad (2.13)$$

Отже,  $\Delta T_{1,2} > 0$ , якщо  $n \times m \times d_1 > n \times m \times d_2$ , тобто  $D^I > D^{II}$ . Іншими словами, обліт відбувається швидше, коли відстань між робочими сегментами в ряду більша, ніж за відстань між самими рядами.

Різницю  $\Delta T_{2,3}$  між  $T_2$  та  $T_3$  обчислимо так:

$$\begin{aligned} \Delta T_{2,3} = T_2 - T_3 &= \frac{D^{II}}{v} - \frac{D^{III}}{v} \\ &= \frac{(n \times m \times (d_1 + d_2) - (m - 1) \times d_2)}{v} \\ &\quad - \frac{\sqrt{(d_1^2 + d_2^2)} \times n \times m}{v}. \end{aligned} \tag{2.14}$$

Отже,  $\Delta T_{2,3} > 0$ , якщо  $(n \times m \times (d_1 + d_2) - (m - 1) \times d_2) > \sqrt{(d_1^2 + d_2^2)} \times n \times m$ , тобто  $d_1 + d_1 > \sqrt{(d_1^2 + d_2^2)}$ , що є завжди істинним з огляду на геометричну інтерпретацію відстані.

З проведеного аналізу за формулами (2.13-2.14) маємо, що для будь-яких значень  $d_1$  та  $d_2$  різниці  $\Delta T_{1,2} > 0$  та  $\Delta T_{2,3} > 0$ , що означає  $T_1 < T_2$  та  $T_2 < T_3$ . Тому, для мінімізації часу обльоту  $T$ , траєкторія руху  $D^I$  є найкращим вибором для виконання робочих місії групою БПЛА.

З огляду на проведений аналіз часу обльоту за різними траєкторіями руху БПЛА, встановлено, що “пріоритетна траєкторія руху”  $D^I$  подана на рис. 2.4 є найкращою для використання в подальшому дослідженні. Проведені розрахунки підтверджують, що обрана “пріоритетна траєкторія” забезпечує найменший час обльоту порівняно з рештою траєкторій, що підкреслює її цінність під час прокладання автоматичних маршрутів для групи БПЛА. Ця траєкторія, також, забезпечує найвищу швидкість обробки даних в експериментальному робочому середовищі, підвищуючи таким способом ефективність моніторингу та збору даних.

### 2.2.1 Метод побудови маршрутів БПЛА за технологією самонавчання

Переміщення БПЛА в робочому середовищі з елементами самонавчання є ключовою задачею для обчислення кількості заданих структурних об'єктів однієї природи. Необхідно розробити новий метод, який реалізує автоматичне прокладання маршруту групи БПЛА. Для цього, спершу формалізуємо наступні проблеми, що виникають у процесі протікання робочої місії.

Переміщення БПЛА в робочому середовищі з елементами самонавчання є ключовою задачею для обчислення кількості структурних об'єктів однієї природи. Необхідно розробити новий метод, який реалізує автоматичне прокладання маршруту групи БПЛА. Для цього спочатку розглянемо проблеми, які виникають у процесі протікання робочої місії:

1. Переміщення та синхронізація між множиною БПЛА в межах робочого сегменту відбувається повільно.
2. Переміщення групи БПЛА між робочими сегментами відбувається повільно.
3. Можливість прокладати маршрути з елементами самонавчання для одного, або групи БПЛА.
4. Збільшення кількості даних, яку БПЛА може опрацювати під час роботи в режимі реального часу за рахунок оптимізація ресурсів заряду батареї БПЛА.
5. Самовідновлення у випадку критичного збою.

З огляду на вищезгадані проблеми, запропонований метод має складатися з таких блоків: 1) обробка вхідних даних у вигляді матриць координат робочого середовища, 2) позиціонування апаратних пристроїв у тривимірному координатному просторі, 3) визначення прогнозних координат та 4) оброблення вихідних даних, якими є обчислена матриця квазіоптимальних координат для БПЛА в робочому середовищі.

Схему метода побудови маршрутів БПЛА згідно технологій самонавчання подано на рис. 2.5.

Блок 1. Вхідні дані для поточного методу отримуються одним із наступних способів:

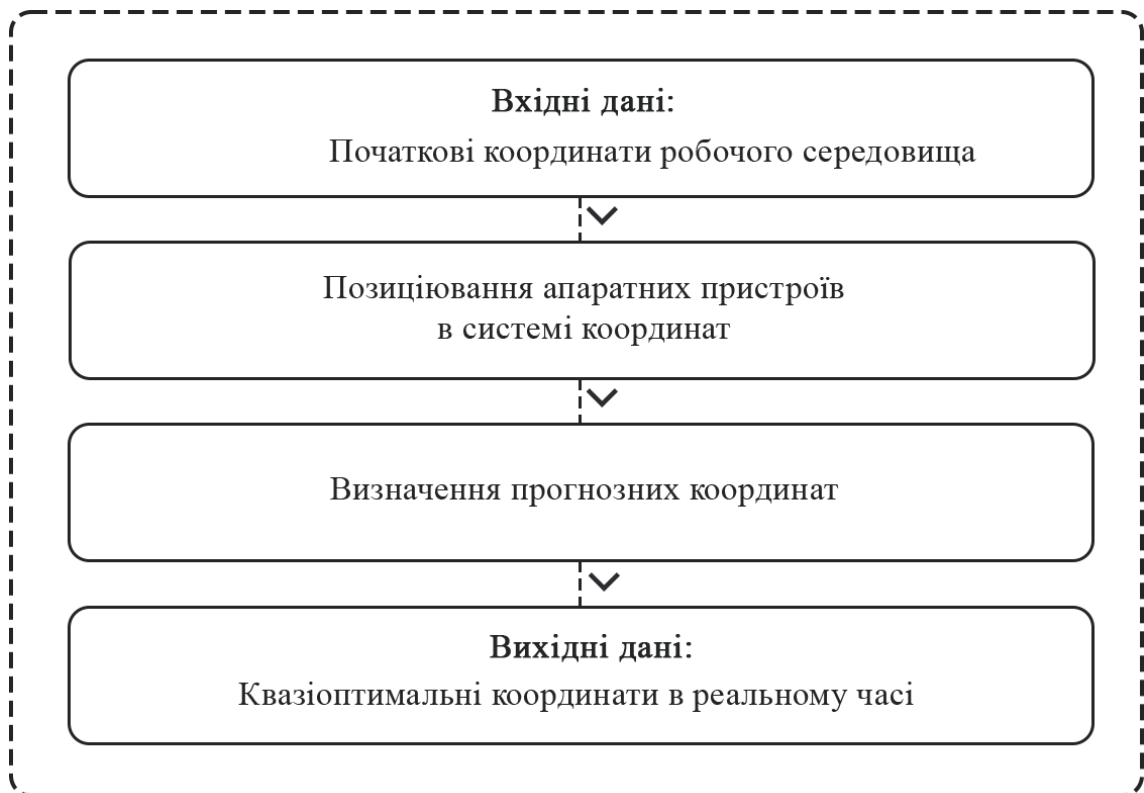


Рисунок 2.5 – Схема методу побудови маршрутів БПЛА на основі технологій самонавчання

1. За першого використання автоматизованої системи в робочому середовищі оператор оцифровує його ландшафт, з огляду всі перешкоди та особливості рельєфу місцевості. Збирається велика кількість даних про об'єкти та території, які будуть охоплюватися робочою місією, за допомогою апаратних приладів та технічного устаткування. Оператор визначає множину точок у тривимірній площині, що складається з таких елементів:

- а) координати початкової точки робочого середовища;
- б) координати кінцевої точки робочого середовища;
- в) матриця координат початкових точок робочих сегментів;
- г) матриця координат кінцевих точок робочих сегментів;
- г) матриця координат початкової траєкторії руху БПЛА.

2. При наступних запусках робочої місії в поточному робочому середовищі вхідні дані можуть бути отримані із таких подій:

- а) описаний вище сценарій першого запуску автоматизованої системи;
- б) змінені оператором вхідні дані на основі першого запуску;

в) дані, скореговані системою самонавчання, у результаті виконання першої робочої місії.

Блок 2. Позичіювання пристроїв у тривимірному просторі здійснюється за допомогою навігаційних систем GPS. Ця система здатна забезпечувати досить точне позиціювання літальних апаратних у просторі. Водночас простий набір координат, який формує система GPS, не може служити єдиним розв'язком необхідних задач. Тому, для утворення більш якісних маршрутів, що дасть змогу знизити помилки під час обчислення структурних об'єктів у режимі реального часу, у роботі використано технологію RTK (див. п. 1.3.1), як основу для побудови маршрутів БПЛА у тривимірному координатному просторі. Кінематика реального часу дає змогу синхронізувати розташування БПЛА у робочому середовищі до сантиметрової точності за допомогою супутникової системи навігації.

Обліт групою БПЛА робочих сегментів відбувається в складних умовах. У процесі польоту БПЛА можуть відхилятися від наперед заданих траєкторій внаслідок впливу різноманітних зовнішніх чинників. Внутрішніми чинниками можуть бути технічні обмеження можливостей БПЛА, зокрема їх рівень заряду акумулятора. У зв'язку з цим для розв'язання завдання щодо ефективності роботи автоматизованої системи в частині якості планування обльоту БПЛА введемо критерії якості планування обльоту БПЛА та узагальнений критерій, що визначатиме інтегрований показник якості планування обльоту БПЛА. Тому, множиною  $M_{kr,S,pl} = \{m_{kr,S,pl,1}, m_{kr,S,pl,2}, \dots, m_{kr,S,pl,n_{kr,S,pl}}\}$  задамо показники якості обльоту групою БПЛА. Введемо для кожного показника критерії, які будуть задані функціями з результатами оцінювання показників якості обльоту групою БПЛА так:

$$r_{kr,S,pl,i} = k_{kr,S,pl,i}(m_{kr,S,pl,i}), \quad (2.15)$$

де  $k_{kr,S,pl,i}$  – функція з результатом оцінювання показників якості обльоту групою БПЛА,  $n_{kr,S,pl}$  – кількість показників,  $i = 1, 2, \dots, n_{kr,S,pl}$ .

Функції результатів оцінювання (2.15) показників якості обльоту групою БПЛА задані в проміжку  $[0,1]$ . Для оцінювання якості планування польоту групи БПЛА розроблені такі критерії якості планування: 1) виконання завдання за найменший час, 2) відстань, яку пролітають БПЛА в робочому середовищі, 3) оптимізація ресурсів заряду батареї та 4) кількість даних, яку БПЛА може опрацювати під час роботи в режимі реального часу.

З огляду на вищезазначені критерії, узагальнений критерій планування якості обльоту групою БПЛА задамо так:

$$R_{D_k} = \frac{1}{n_{kr,S,pl}} \sum_{i=1}^{n_{kr,S,pl}} r_{kr,S,pl,i}, \quad (2.16)$$

де  $R_{D_k}$  – значення узагальненого критерію, яке знаходиться в проміжку  $[0,1]$ ,  $r_{kr,S,pl,i}$  – функція з результатом оцінювання показників якості обльоту групою БПЛА,  $n_{kr,S,pl}$  – кількість показників,  $i = 1, 2, \dots, n_{kr,S,pl}$ .

Для автоматизованої системи кількість БПЛА, дальність польоту, мінімальний радіус розвороту та діапазон швидкостей визначено, як вхідні дані. Розподілення цілей між декількома БПЛА, послідовність проходження кожним БПЛА підмножини цілей та спосіб переміщення до кожної робочої області визначено, як характерні властивості у задачах планування маршрутів.

Блок 3. Вхідні дані отримані поточним методом корегуються системою самонавчання ґрунтуючись на  $Q$ -алгоритмові підходу штучного інтелекту навчання з підкріпленням (див. п. 1.3.2) на основі даних про робоче середовище, перешкод та початкових координат БПЛА. Компонент самонавчання (рис. 2.6) утворює матриці координат та працює відповідно до підходу “дія-винагорода”, що виконується так званім інтелектуальним агентом.

Такий підхід до самонавчання маршрутів БПЛА зветься навчанням з підкріпленням, за якого здійснюється ітераційний процес автоматичного корегування координат розташування БПЛА.

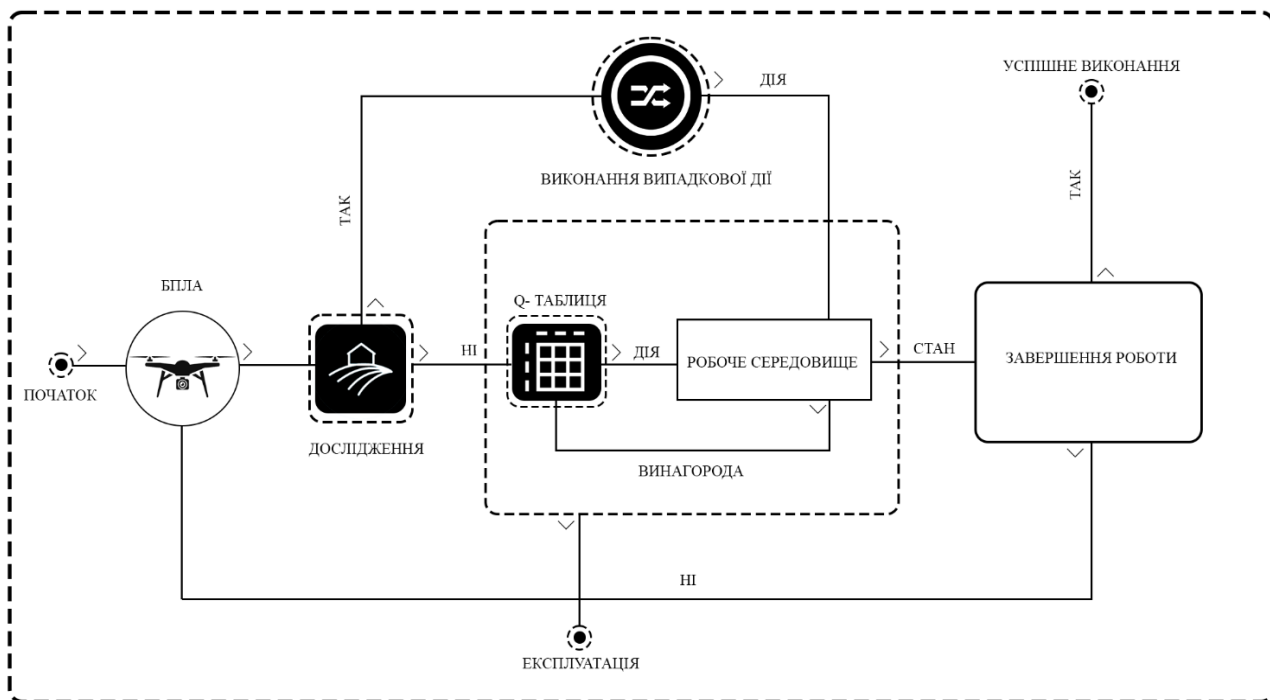


Рисунок 2.6 – Схема роботи компонента побудови маршрутів на основі технології самонавчання

З огляду на повторювані дії компонента, модуль генерує та постійно оновлює  $Q$ -значення за формулою (1.12) для кожного БПЛА при виконанні програмної місії, допоки не буде досягнутого найточнішого значення, яке буде вважатися квазіоптимальним.

Приклад результату роботи компонента побудови маршрутів із  $Q$ -значеннями (1.12) подано в табл. 2.1.

Таблиця 2.1

#### Ітераційний процес навчання планування маршрутів

Стан	Дія $A_1$	Дія $A_2$	...	Дія $A_T$
$S_1$	$Q(S_1, A_1)$	$Q(S_1, A_2)$	...	$Q(S_1, A_T)$
$S_2$	$Q(S_2, A_1)$	$Q(S_2, A_2)$	...	$Q(S_2, A_T)$
...	...	...	...	...
$S_T$	$Q(S_T, A_1)$	$Q(S_T, A_2)$	...	$Q(S_T, A_T)$

В табл. 2.1  $Q(S_t, A_t)$ ,  $t = 1, \dots, T$ , є кількісним вираженням винагороди, яку отримує інтелектуальний агент  $Q$ -алгоритму за досягнення поточного стану  $S_t$  та виконання дії  $A_t$  в момент часу  $t$ ,  $T$  – загальний час роботи інтелектуального агента.

Інтелектуальний агент  $Q$ -алгоритму навчається на основі даних, отриманих у процесі виконання попередньої робочої місії групи БПЛА. У разі, якщо робоча місія виконується вперше, використовуються первинні дані, які задаються системою та вводяться оператором. Створення первинних даних включає наступні етапи:

а) для максимального забезпечення покриття всіх структурних об'єктів у робочих сегментах вираховується значення координатних точок у робочому середовищі залежно від кількості апаратних пристроїв у групі, з огляду на такі параметри: кут та висота охоплення камери БПЛА, значення відстані між пристроями у групі та значення висоти у групі;

б) розподілення зон відповідальності між кожним БПЛА у групі БПЛА, що полягає в розбитті робочого сегмента на умовні кластери з унікальними ідентифікаторами, де координатні точки траєкторії руху БПЛА задаються щодо кластерної площини.

У результаті створення первинних даних кожен пристрій приймає власний заданий вектор руху, а робочий сегмент рівномірно розподіляється на зони відповідальності, що закріплюються за кожним БПЛА групи БПЛА.

У певний момент часу  $t$ , коли кожен  $b$ -ий БПЛА групи БПЛА,  $b = 1, \dots, 3$ , виконує дію  $A_t^b$ , інформаційна система генерує подію про результат виконання функції “об’єднання відеокадрів”. Перебуваючи в поточному стані  $S_t$ , якщо результат функції є успішним, то компонент побудови маршрутів буде оновлювати  $Q$ -значення наступним чином:

1. Вхідними даними є сформована множина допустимих координат:  $W_t^b = \{A_{x,y,z,1}^b, A_{x,y,z,2}^b, \dots, A_{x,y,z,n_{xyz}}^b\}$ , де  $A_{x,y,z,n_{xyz}}^b$  – координати різних точок кількістю  $n_{xyz}$  у тривимірному просторі робочого середовища, які належать множині  $W_t^b$ ,



допустимих координат кожного  $b$ -го БПЛА в групі БПЛА,  $n_{xyz}$  – порядковий номер цих точок, що також вказує на їхню кількість.

Множина  $W_t^b$  визначена, як успішна, або початково-сформована, що вважається успішною на поточний момент часу  $t$ .

2. Між кожними координатами множини  $W_t^b$  та координатами  $C_{x,y,z}^b$ , що відображають фактичне перебування БПЛА  $b$  у робочому середовищі, визначається “ліквідна відстань” за формулою:

$$D^b(A_{x,y,z,i}^b, C_{x,y,z}^b) = \sqrt{(A_{x,i}^b - C_x^b)^2 + (A_{y,i}^b - C_y^b)^2 + (A_{z,i}^b - C_z^b)^2}, \quad (2.17)$$

де  $D^b(A_{x,y,z,i}^b, C_{x,y,z}^b)$  – ліквідна відстань між фактичними координатами  $C_{x,y,z}^b$  та координатами  $A_{x,y,z,i}^b$  для кожного БПЛА  $b$  з множини  $W_t^b$ ,  $i = 1, \dots, n_{xyz}$ .

Якщо відстань  $D^b$  набуває значення, що менше за допустимий поріг помилки  $\varepsilon = 0.1$  для будь-якої координати  $A_{x,y,z,i}^b$  з множини  $W_t^b$ , то поточні координати  $C_{x,y,z}^b$  вважаються успішною координатною точкою, і  $Q$ -значення оновлюється за умови:

$$\exists A_{x,y,z,i}^b \in W_t: D(A_{x,y,z,i}^b, C_{x,y,z}^b) < \varepsilon, \quad (2.18)$$

відповідно до формули (1.12), що свідчить про те, що дрон успішно дотримується заданої траєкторії руху та успішно виконує підхід “дія-винагорода”. У такому випадку  $Q$ -значення для даної дії  $A_t$  у поточному стані  $S_t$  буде оновлене.

У випадку, коли інформаційна система генерує подію з негативним результатом виконання функції “об’єднання відеокadrів”,  $Q$ -значення для даної дії  $A_t$  у поточному стані  $S_t$  не буде оновлено та залишиться без змін. Далі компонент планування маршрутів продовжує процес “самонавчання”, забезпечуючи дотримання групою БПЛА заданого маршруту, та обчислює  $Q$ -значення на встановлених координатних точках у робочому середовищі.

Блок 4. В результаті, вихідними даними методу побудови маршрутів БПЛА на основі технологій самонавчання є матриця квазіоптимальних координат для розташування БПЛА в тривимірному просторі. Утворена матриця координат передається в модуль управління групою БПЛА, що є системним компонентом автоматизованої системи.

У режимі реального часу дати оцінку новоствореним координатам можуть лише сторонні чинники інформаційної інфраструктури автоматизованої системи. Тому, поточний компонент відправляє інформаційну подію у програмний модуль управління БПЛА для встановлення координат розташування у тривимірному просторі.

Таким чином, розроблено метод побудови маршрутів БПЛА згідно технологій самонавчання, який полягає в позиціюванні апаратних пристроїв у тривимірному координатному просторі в реальному часі за рахунок самонавчання при побудові маршрутів їх руху, що дає змогу покращити переміщення та синхронізацію між групою БПЛА або одного БПЛА в межах робочого сегменту і за рахунок цього збільшення кількості опрацьованих даних.

### 2.2.2 Інформаційні потоки автоматизованої системи під час виконання програмної місії

Архітектура типових компонентів автоматизованої системи обчислення структурних об'єктів однієї природи подається та функціонує через різні типи станів. Заразом, функціонування автоматизованої системи має відбуватися з огляду на автономні зміни станів її архітектури без втручання оператора. Порядок передачі інформації між компонентами та всіма рівнями програмних модулів автоматизованої системи (див. рис. 2.3) організовано через аналітичні залежності між ними. Основою для визначення стану певного компонента системи є множина дій, яку виконують БПЛА під час протікання програмної місії. На рис. 2.7 подано діаграму станів автоматизованої системи.



- а) апаратні прилади, що розташовані в робочому середовищу, працюють та забезпечені мережевим зв'язком;
- б) БПЛА зареєстровані в системі та можуть формувати групу;
- в) налаштований супутниковий зв'язок з апаратними приладами;
- г) немає активних програмних місій.

Усі вище перераховані характеристики є обов'язковими, щоб задовольнити функційні потреби в можливості створення програмної місії та перевести центральний програмний модуль до стану “готовність”. Автоматизована система перебуваючи в стані “готовність” може ініціалізувати створення програмної місії оператором та сформувати інформаційну подію для активації інших програмних компонентів.

Відповідно до впроваджених функцій управління та моніторингу із залученням інших підсистем та їхніх рівнів, нижче виділено основні компоненти підсистеми планування маршруту БПЛА (див. рис. 2.8).

Робочий сегмент – це територія робочого середовища, навколо якої група БПЛА, що поєднана програмними характеристиками, виконує обліт та відвідує максимальну кількість робочих зон за найкоротшим шляхом із мінімальною затримкою. Головними характеристиками експериментального середовища є поділ на дозволені зони та заборонені робочі сегменти.

Стан – це програмна структурна частина, яка здійснює самоконтроль щодо аналізу місця перебування групи БПЛА в координатному просторі. У разі тривалого часу перебування в одному й тому ж стані, в автоматичному режимі здійснюється обробка та аналіз поточних завдань та за необхідності переведення до іншого стану всієї групи пристроїв. З огляду на короткий термін виконання роботи БПЛА також інтегровано додатковий модуль для управління станом ресурсів батареї БПЛА. Така реалізація дає змогу керувати всіма можливими станами, як програмними так і апаратними.

Далі розглянемо можливі стани автоматизованої системи.

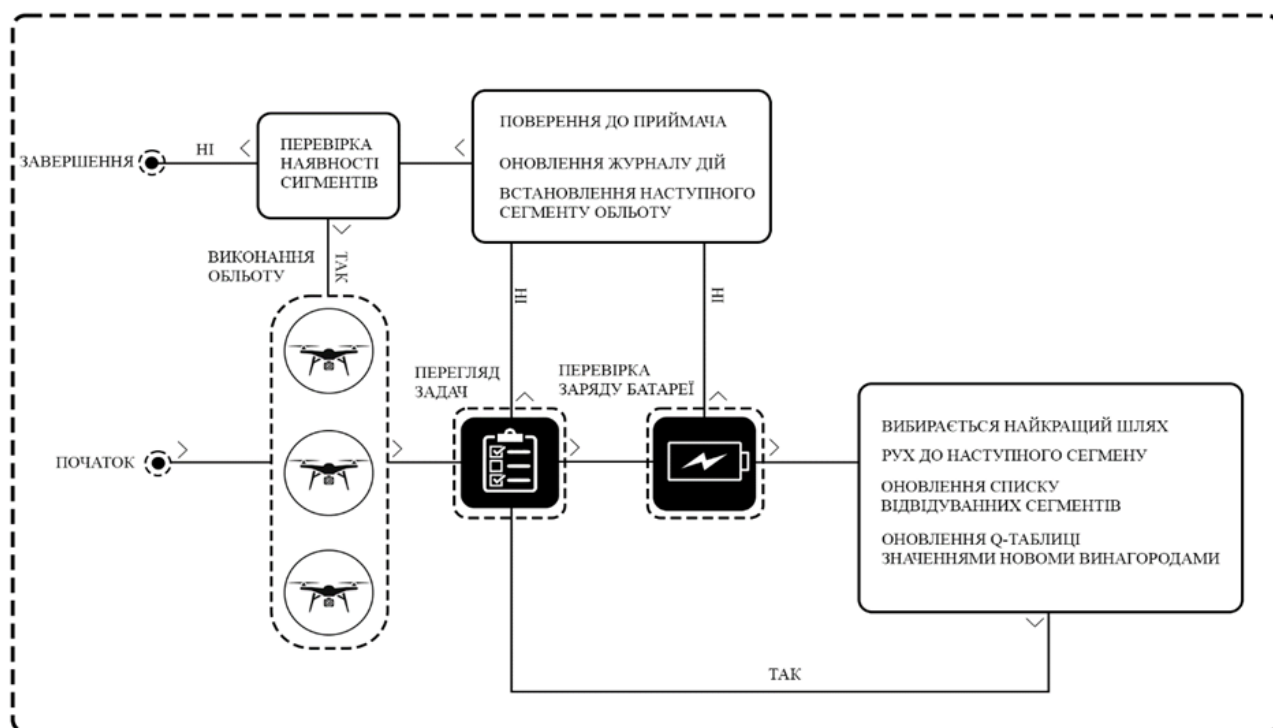


Рисунок 2.8 – Схема руху БПЛА у робочому середовищі

*Стан “Дія”*. Перехід зі стану в стан відбувається через зміну певних параметрів, якими характеризується підсистема утворення автоматичних маршрутів. Основний стан такої програмної структурної підсистеми – дія, яку множина пристроїв виконує у вигляді об'їоту до цільових точок завдяки утвореним траєкторіям у координатному просторі.

*Стан “Середовище”*. Представлення тривимірного координатного простору є основою побудови механізмів та визначення характеристик автоматичних траєкторій. Завдяки програмному модулю відбувається розподіл експериментального середовища на дозволені та заборонені сегменти.

*Стан “Завершення”*. Формування завершення місії групи БПЛА визначається за допомогою станів. Структурні компоненти, які відстежують стани всіх інших підсистем формують механізм завершення. Опираючись на певні параметри стану всієї місії, компонент приймає рішення про завершення. У випадку проходження всіх робочих сегментів та виконання всіх поставлених задач система переходить у стан “позитивного завершення”. Життєвий цикл місії може бути зупинений у випадку, коли модуль отримає параметри, згідно з якими група апаратних пристроїв має перейти у стан “очікування”. Перехід у цей стан

відбувається в результаті зниження рівня ресурсу хоча б одного БПЛА нижче допустимого рівня. Результат відновлення ресурсів БПЛА переводить програмну систему в стан “виконання”. Після цього проводиться аналіз незавершених робочих зон для завершення місії. Обробка критичних подій у системі дає змогу виконати самовідновлення. Заразом вплив зовнішніх чинників може перевести систему в “негативне завершення”; як наслідок, місія не вважатиметься завершеною, а система не відновлюватиме політ групи БПЛА.

*Стан “Винагорода”.* Результативні значення модуля “розумного” планування маршрутів. Компонент ітераційно генерує відповідь у вигляді станів на основі визначених характеристик робочого середовища. Система розглядає оцінку станів модуля в результаті виконання низки дій. Якщо група БПЛА виконала обліт робочого сегмента, то результативним значенням буде “позитивна винагорода”; водночас якщо відбулося потрапляння пристроїв у заборонений сегмент для польотів чи система подала сигнал критичного збою, встановлюється “негативна винагорода”.

2. Модуль управління групою БПЛА. Координація БПЛА у робочому середовищі під час виконання програмної місії функціонує в наступних станах:

*Стан “Початок”.* Особливістю знаходження групи БПЛА на початку робочого середовища є фіксація їхнього положення в координатному просторі.

*Стан “Призупинення”.* Група БПЛА виконує політ у динамічному робочому середовищі, де пориви вітру можуть впливати на траєкторію руху та розташування в просторі. У зв’язку з цим, програмний модуль управління формує подію, яка інформує інші залежні модулі про призупинення виконання своїх функційних зобов’язань. Перебуваючи в цьому стані, автоматизована система приймає рішення щодо подальших кроків виконання:

1 – продовження – отримуючи сигнал про продовження, група БПЛА буде переведена на початкову точку робочого сегмента та встановлюється стан “Початок”;

2 – критичний збій – отримуючи сигнал про критичний збій, програмний компонент встановлює стан “Завершення”.

*Стан “Завершення”*. Перебуваючи в цьому стані, система зберігає для аналізу всю необхідну інформацію про перебіг подій під час виконання програмної місії.

Таке подання компонентів через стани, у яких можуть перебувати програмні модулі під час функціонування, дає змогу визначати стан цілісності всієї автоматизованої системи та стани цілісності її компонентів. Розроблена архітектура передбачає можливість збільшення кількості рівнів підсистем без зміни її архітектури загалом.

2.3. Генерація зображень заданих структурних об’єктів в тривимірному просторі робочого середовища

#### 2.3.1 Створення програмної місії в системі

Активація роботи автоматизованої системи передбачає формування групи БПЛА (або одного БПЛА), яка здійснюватиме політ у робочих сегментах. Необхідною умовою функціонування БПЛА у групі є уніфікація їхніх особливостей та характеристик, а саме: 1) унікальний ідентифікатор БПЛА, 2) ширина охоплення камери, 3) місткість акумулятора та 4) час перебування у польоті.

Зіставлення цих характеристик між БПЛА дає можливість оператору провести реєстрацію пристроїв у системі. Також, для формування групи оцінюються параметри польоту БПЛА, які мають бути однаковими, або наближені один до одного. Розташування кожного БПЛА в групі під час виконання польоту залежить від ширини охоплення його камери: чим вона більша, тим нижчу позицію займає БПЛА. Будь-які новостворені у системі групи БПЛА мають фіксовану кількість пристроїв, які оператор може реєструвати. Розглядатимемо для поставленого завдання групу з не більше чотирьох БПЛА. Упродовж експлуатаційного часу БПЛА можуть мати різного роду механічні пошкодження, тому заздалегідь передбачено можливість утворювати групу БПЛА лише зі справними пристроями.

На рівні перетворення інформаційних потоків автоматизованої системи  $S$  (див. рис. 2.2) відбувається проектування робочого середовища у вигляді координатного простору та виділення дозволених і заборонених зон для польоту БПЛА. Підсистему управління польотними даними інтегровано в автоматизовану систему на основі технології DJI GS PRO (див. п. 1.3.1). Результатом застосування технології DJI GS PRO є набір вихідних даних поданих у вигляді тривимірних GPS координат. Основна централізована підсистема залучає ці дані, як вхідні параметри підсистеми для перетворення координат у вигляді матриці станів. Водночас ці матриці слугують вхідними даними для планування та автоматичного формування маршрутів програмних місій.

Програмна місія формується в автоматизованій системі  $S$ , відповідно до лінійної матриці векторів підмножини структурних об'єктів за формулою (2.5). Вона слугує відображенням робочої місії (див. п. 2.2), яку здійснює група БПЛА в робочому середовищі. Для розгляду основних кроків управління місіями в автоматизованій системі вважатимемо, що група БПЛА не виконувала жодного польоту, а інсталяція всіх потрібних модулів та взаємодій усіх компонентів завершилась успішно. Створення програмної місії в системі  $S$  щодо обльоту БПЛА зон робочого середовища подамо такими кроками:

Крок 1. Вибір та підключення групи БПЛА до програмної місії в системі.

1.1. Налаштовані та зареєстровані у системі БПЛА встановлюються на початкові точки польоту в робочому середовищі. Оператор безпосередньо здійснює перевірку на механічні пошкодження та можливості працювати з мережею. Якщо результати перевірки є позитивними, оператор формує групу БПЛА.

1.2. Перевіряється налаштування камери та якість відеоряду.

1.3. Визначаються стан та якість мережі. Особлива увага приділяється можливості перепідключення до мережі усіх пристроїв у разі втрати зв'язку.

1.4. Під час перевірки та з огляду на позитивні результати виконання кроків 1.1-1.3, оператор отримує можливість створити програмну місію в системі для обльоту робочого середовища.



Крок 2. Управління місією обльоту групи БПЛА. Автоматизована система забезпечує управління у двох режимах: 1) початковий та 2) автоматичний.

2.1. У початковому режимі оператор розробляє маршрут обльоту робочого середовища власноруч. Головна мета такого підходу – це встановити початкові та кінцеві точки маршруту групи БПЛА. Отримані результати аналізуються та корегуються оператором для отримання найбільш точних даних, які надалі слугуватимуть вхідними даними для автоматичного режиму. Всі стани у початковому режимі програмних модулів є детермінованими, із наперед ініціалізованими значеннями. Для отримання цілком прогнозованого результату, оператор має змогу вносити початкову кількість структурних об'єктів, наприклад плодів на деревах, у систему для подальшого аналізу.

2.2. Прийняття рішення щодо подальшої роботи всієї системи загалом згідно даних початкового кроку виконується в автоматичному режимі завдяки системі самонавчання. Водночас забороняється вносити корегування станів управлінських підсистем. Переходи системи з одного стану в інший здійснюється шляхом подій, на які реагує система у результаті роботи групи БПЛА. На цьому кроці оператор має можливість відстежувати усі дії групи завдяки системі моніторингу. Для запуску автоматичного режиму управління групою БПЛА обов'язковим є використання даних хоча б одного завершеного повного маршруту.

Крок 3. Завершення місії групи БПЛА.

3.1. Оператор має можливість завершити виконання програмної місії в будь-який момент часу. Після прийняття такого рішення, автоматизована система переводиться в стан “завершення” ще на етапі виконання робочої місії групою БПЛА. Як наслідок, група БПЛА повертається на початкову або кінцеву точки задля уникнення механічних пошкоджень, які можуть бути отримані за негативного впливу погодних умов. Група БПЛА прямує до тієї точки, відстань до якої є меншою.

3.2. У випадку виходу з ладу хоча б одного пристрою із групи, підсистема самовідновлення надсилає у центральний модуль сигнал про “критичний” стан

системи. В результаті, стан місії визначається “завершеним”, а програмні модулі приймають належні стани для подальшого аналізу подій.

3.3. За умови втрати зв'язку всієї групи БПЛА, або хоча б одного БПЛА, в програмний модуль моніторингу надаються координати тривимірного простору робочого середовища з номером вектору підмножини (наприклад, номер фруктового ряду та місце), коли пристрій був виявлений востаннє. Надання відомостей про географічне розташування БПЛА дає оператору можливість легко знайти апаратні пристрої у робочому середовищі та провести їхній належний огляд.

Перебування програмних модулів управління маршрутом та управління групою БПЛА в одному стані впродовж більше 10 хв свідчить про критичну помилку в функціонуванні системи загалом. Як наслідок, програмна місія перейде в стан “завершення” автоматично, а група БПЛА прямуватиме до найближчої точки (початкової або кінцевої). Належне функціонування автоматизованої системи та всіх її підсистем за сприятливих погодних умов в робочому середовищі фруктового саду призведе до “позитивного” завершення програмної місії групи БПЛА.

Отримання відповідей про стани та результати в процесі роботи програмної місії подається на модуль моніторингу. Аналізуючи ці дані, оператор має змогу керувати майбутніми діями та корегувати стани наступних місій для покращення значень критеріїв оцінювання роботи групи БПЛА. Обробка програмним модулем невизначеностей, пов'язаних з відсутністю належних станів різних підсистем, зумовлює формування звіту про помилки в системі.

Таким чином, в автоматизованій системі передбачений програмний модуль моніторингу, який здійснює контроль стану мережі та повідомляє про всі критичні стани зв'язку з групою БПЛА та кожним пристроєм окремо. Модуль моніторингу надає інформацію про стан мережі та забезпечує управління приладами, які встановлені в робочому середовищі.

### 2.3.2 Програмний інтерфейс автоматизованої системи

Відповідно до сформованих вимог до автоматизованої системи управління декількома БПЛА, початок роботи всього програмного інтерфейсу розпочинається після утворення програмної місії оператором. Програмний інтерфейс автоматизованої системи використовує наступні системні компоненти: а) обробка координації групи БПЛА, б) визначення та корегування станів, в) визначення ресурсу батареї кожного БПЛА у групі, г) програмна система самовідновлювання.

Програмний інтерфейс взаємодіє з головним центральним програмним модулем, що дає змогу здійснювати перехід між різними визначеними рівнями на основі інформації, отриманої з інших рівнів автоматизованої системи (див. рис. 2.2). Схему програмного інтерфейсу зображено на рис. 2.9.

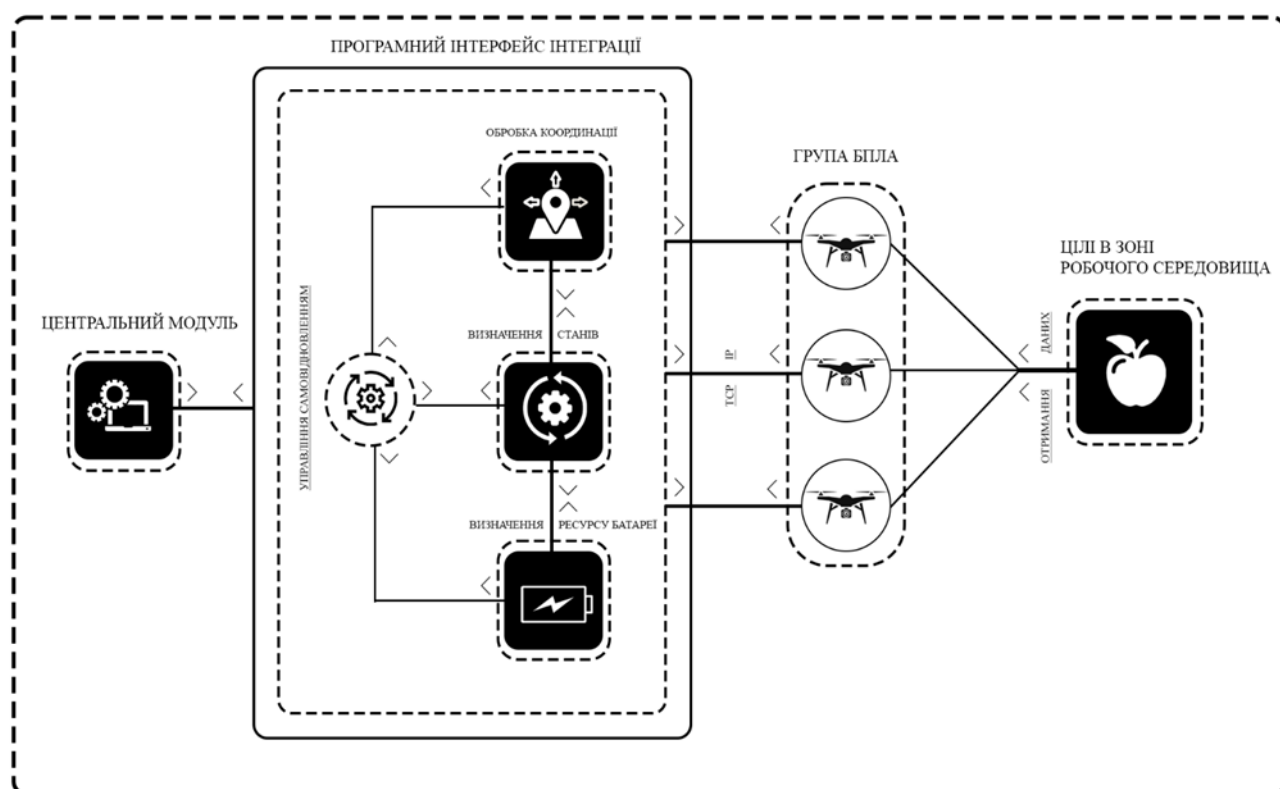


Рисунок 2.9 – Схема програмного інтерфейсу для генерації зображень структурних об'єктів однієї природи в тривимірному просторі

Визначення координатних точок у тривимірному програмному просторі робочого середовища та фіксування чітких робочих зон призводить до прийняття

“початкового” стану системи на рівні програмного інтерфейсу. Група БПЛА отримує координати початкової точки в просторі, що зумовлює початок обльоту ними робочої цільової зони у робочому середовищу за вказаним маршрутом (рис. 2.10).

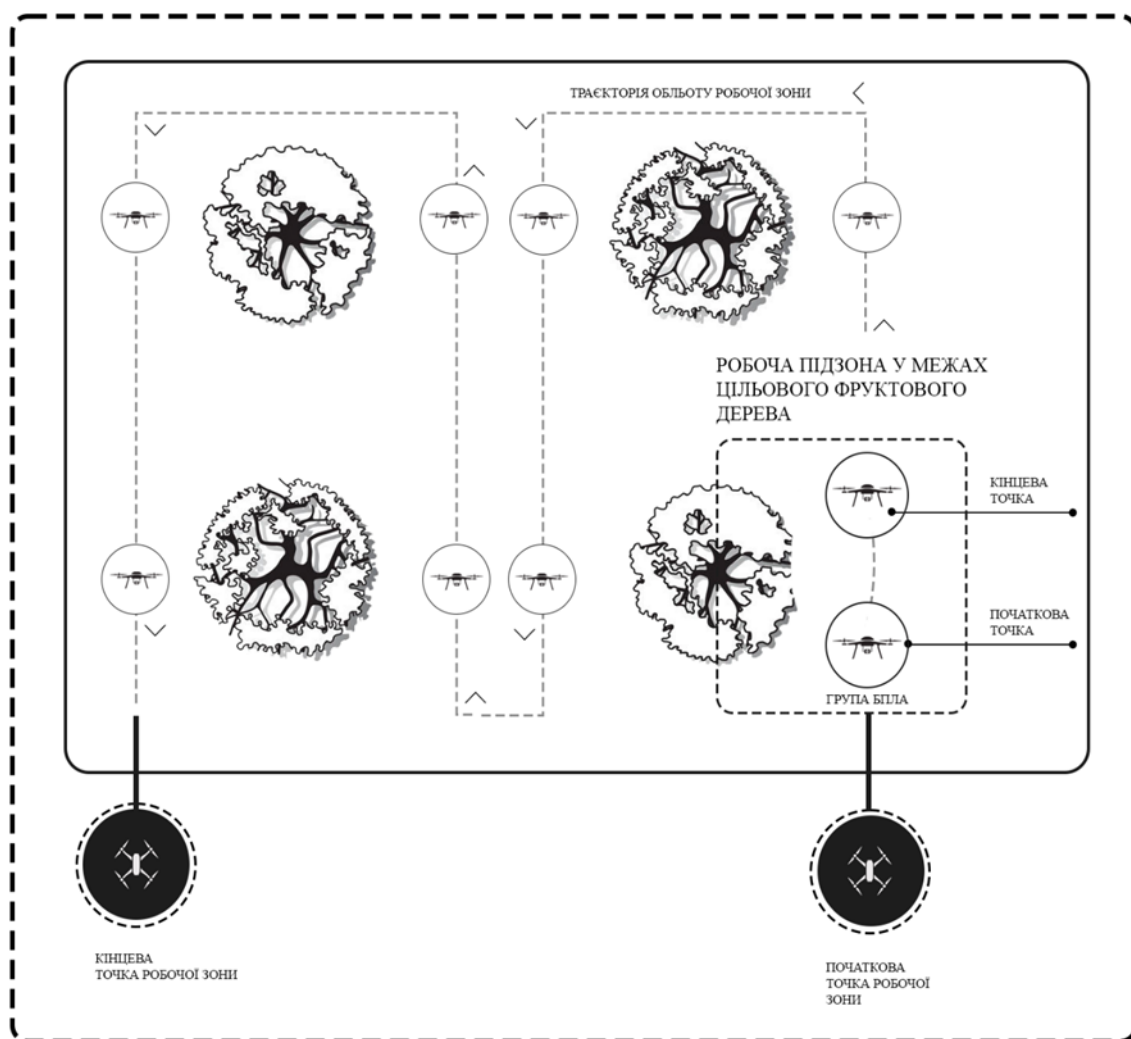


Рисунок 2.10 – Схема руху групи БПЛА у робочій зоні

Окремо встановлюються координати кінцевої точки для програмної місії БПЛА. Початкова та кінцева точки подані у вигляді програмних координат неробочих зон, де БПЛА можуть виконувати операції зльоту та приземлення. Під час перебування групи БПЛА у неробочих зонах, оператор може виконувати технічну підтримку над БПЛА та корегувати їхнє розташування відносно один одного. Результатом переміщення групи БПЛА у кінцеву точку є завершення програмної місії та призначення стану “завершення” всьому програмному інтерфейсу.

Вхідні дані про стан ресурсу батареї кожного БПЛА у групі передаються в результаті підключення БПЛА до мережі та їхнього перебування в початковій точці. Підсистема управління групою БПЛА постійно оцінює шанс успішного завершення місії з огляду на зниження ресурсу батареї в процесі роботи БПЛА. У випадку неможливості “позитивного” завершення місії через зниження рівня ресурсу батареї хоча б одного БПЛА нижче допустимого рівня, системою приймається рішення про переведення в стан “завершення” і група автоматично відправляється в кінцеву точку.

Необхідним засобом підтримки цілісності архітектури всього програмного інтерфейсу є підсистема самовідновлення після критичних збоїв. З огляду на виконання програмних місій у складному робочому середовищі та через мінливість погодних умов у процесі роботи групи БПЛА, є велика ймовірність того, що дані, які БПЛА отримують і подають у систему через мережу, можуть спотворюватись. У разі виявлення спотворених даних під час виконання програмної місії, компоненти автоматизованої системи можуть перевестися у некоректний стан, що призведе до негативного завершення місії.

Розроблений програмний інтерфейс є основою розподілення інформації про характерні особливості між кожним компонентом в процесі свого функціонування. Інтеграція у центральну програмну систему дає можливість використовувати його, як самостійну підсистему для динамічного отримання зображень структурних об’єктів тривимірному координатному просторі за допомогою групи БПЛА.

### 2.3.3 Метод динамічного отримання зображень заданих структурних об’єктів в тривимірному просторі за допомогою декількох БПЛА

Забезпечення повної функціональності автоматизованої системи можливе із впровадженням нового методу динамічного отримання зображень заданих структурних об’єктів використовуючи групу БПЛА. Відповідно до мети роботи, формалізацію методу подамо у вигляді схеми на рис. 2.11.

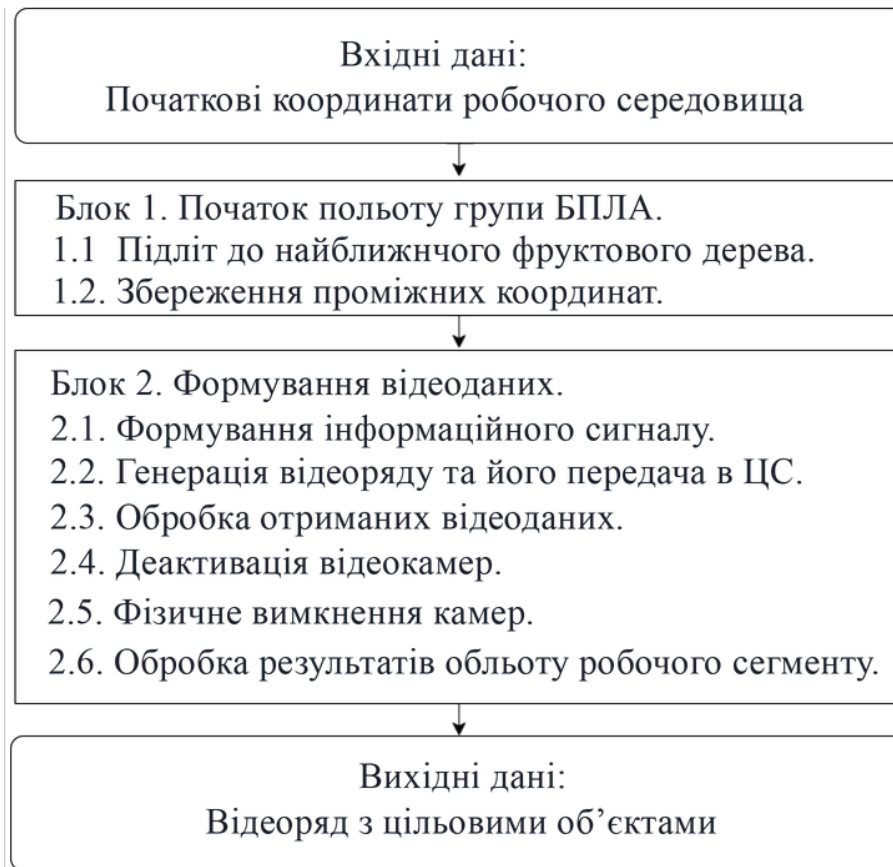


Рисунок 2.11 – Схема методу динамічного отримання зображень заданих структурних об'єктів в тривимірному просторі

Вхідними даними методу динамічного отримання зображень заданих структурних об'єктів є множина даних:

- а) координати початкової точки робочого середовища;
- б) координати кінцевої точки робочого середовища;
- в) матриця координат початкових точок робочих сегментів;
- г) матриця координат кінцевих точок робочих сегментів;
- д) матриця координат початкової траєкторії руху БПЛА.

Припустимо, що запуск центрального програмного модуля здійснено успішно, тобто оператор отримав можливість сформувати групу БПЛА, під'єднати всі пристрої до комп'ютерної мережі та віднести їх до початкової точки, звідки БПЛА розпочнуть зліт до першої робочої зони. Крім того, вважатимемо що програмний модуль планування маршрутів отримав початкові дані просторових координат, а оператор розпочав програмну місію через програмний інтерфейс.

Тоді, метод динамічного отримання зображень структурних об'єктів однієї природи в тривимірному просторі (рис. 2.11) полягає у покроковому виконанні наступних блоків.

Блок 1. Зліт групи БПЛА до першої початкової точки.

1.1. Виконання цього блоку відбувається на початку запуску програмної місії. Програмний координатний простір, в якому функціонує група БПЛА, має підпростори, що визначаються початковою та кінцевою точкою робочої зони в межах цільового фруктового дерева. Зміна траєкторії БПЛА у тривимірному координатному просторі відбувається шляхом зміни їхнього положення у програмній системі координат. Група БПЛА виконує підліт до найближчого фруктового дерева поточної робочої зони. Схематичне подання такого об'льоту наведено на рис. 2.12.

Визначення стану для програмного модуля розташування кожного БПЛА включає перевірку координації та відповідність фізичного розташування всієї групи щодо вказаних програмних координат на початку.

1.2. Занесення інформації, отриманої на кроці 1.1, у центральний модуль та всіх підсистем й визначення станів всіх системних модулів.

Блок 2. Формування відеоданих, отриманих від групи БПЛА у цільовій зоні.

2.1. Після завершення підльоту до цільового фруктового дерева, відповідно до заданих програмних координат, група БПЛА формує інформаційний сигнал, який передається у програмну місію. Обов'язковим є отримання відповіді від програмної системи на активізацію відеокамери кожного БПЛА у групі. Охоплена площа відеоряду визначається шириною охоплення відеокамери, що вмонтована у кожен БПЛА та зміни положення кожного БПЛА у групі від точки початку до точки завершення в межах цільового фруктового дерева у програмній системі координат.

2.2. В процесі активації відеокамер кожен БПЛА генерує відеоряд, який передається в центральну систему.

2.3. Виконується обробка відеоданих у центральній системі та занесення отриманої інформації на кроці 2.2 у базу відеорядів.

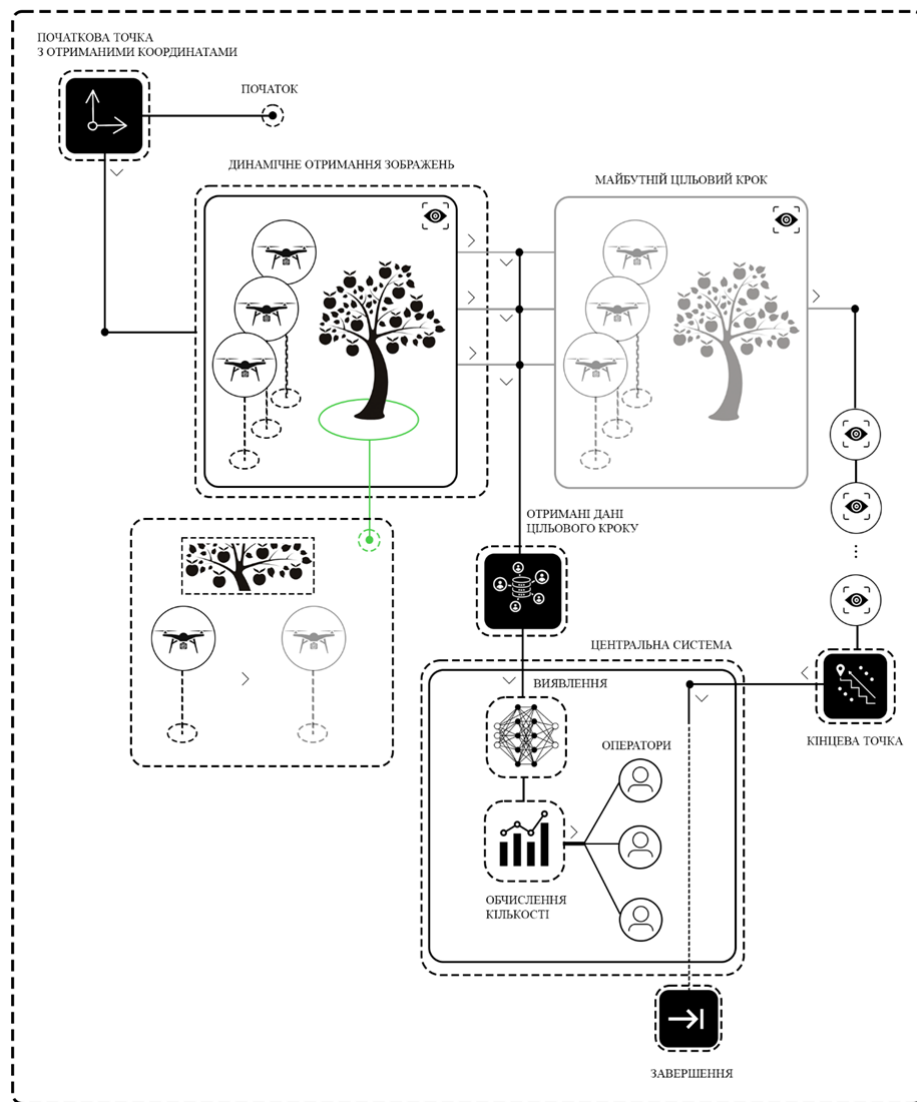


Рисунок 2.12 – Схема обльоту робочої зони групою БПЛА для відстеження заданих структурних об’єктів однієї природи

2.4. Програмна система управління координації групи БПЛА, отримавши значення положення БПЛА у точці завершення цільового фруктового дерева у системі координат, надсилає сигнал про деактивацію камер кожного БПЛА.

2.5. Підсистема управління місії визначає стан камери кожного БПЛА, як деактивований і фізично вимикає камери.

2.6. Здійснюється обробка результатів обльоту та визначення станів для переходу до наступного цільового фруктового дерева.

Прийняття рішень про подальшу роботу групи БПЛА, підтвердження цілісності програмної місії та визначення рівня критичності для продовження



виконання роботи групи БПЛА, відбувається через визначення станів програмних модулів після виконання блоку 2.

Вихідними даними методу є згенерований відеоряд із цільовими об'єктами фруктового саду.

Процес утворення відеорядів для одного цільового фруктового дерева визначається та фіксується, як програмна підсистема. Поведінка програмної підсистеми динамічного отримання зображень заданих структурних об'єктів ґрунтується на комплексному врахуванні вимог розподілення, багаторівневості та автоматизованості, що забезпечує переміщення всієї групи БПЛА від заданих початкових до кінцевих точок програмної місії автономно. Оскільки формування даних для обчислення кількості структурних об'єктів є трудомістким процесом, ця підсистема формалізує наступні характерні властивості:

- а) значення програмних координат розташування;
- б) значення програмних координат розташування початкової та кінцевої точки;
- в) час перебування у точках початку та кінця;
- г) кількість виявлених структурних об'єктів;
- д) фактична кількість структурних об'єктів;
- е) тип та характеристика відеокамери, яка формує відеоряд.

Отримані дані цільового блоку 2 визначають стан та ефективність роботи групи БПЛА у робочій зоні. Якщо програмні модулі набули всі необхідні ознаки для продовження місії, підсистема управління групою БПЛА відправляє сигнал на переміщення групи до наступного цільового фруктового дерева.

Вважатимемо, що динамічне отримання зображень у робочій зоні в межах цільового фруктового дерева відбувається в ідеальних умовах робочого середовища. Тоді просторове розташування БПЛА для виконання місії у робочій зоні у межах цільового фруктового дерева встановлюється на етапі формування групи БПЛА. Характерною ознакою вертикальної позиції у групі є ширина охоплення камери БПЛА, оскільки група може бути сформована різними моделями БПЛА. Кут охоплення відеокамери для цільового фруктового дерева за замовчуванням становить приблизно 135 градусів. Схема виконання програмної



підсистемах управління та планування маршрутів. Спроектowana архітектура розподіленого багаторівневого програмного інтерфейсу, який інтегрується в централізовану систему, забезпечує управління групою БПЛА та її взаємодію із центральним модулем системи в режимі реального часу, а розроблений метод дає змогу динамічно генерувати зображення заданих структурних об'єктів в робочій зоні у межах цільового фруктового дерева.

Таким чином, розроблено метод динамічного отримання зображень заданих структурних об'єктів в тривимірному просторі за допомогою декількох БПЛА, який забезпечує в процесі активації кожного БПЛА генерацію відеоряду в центральну систему, прийняття рішень про подальшу роботу групи БПЛА, підтвердження цілісності програмної місії та визначення рівня критичності для продовження виконання роботи групи БПЛА за рахунок визначення станів модулів та комплексному врахуванні вимог розподілення, багаторівневості та автоматизованості, що покращує узгодження між різними БПЛА та досягнення переміщення всієї групи БПЛА від заданих початкових до кінцевих точок програмної місії автономно.

#### 2.4. Висновки до другого розділу

Розроблено метод побудови маршрутів БПЛА згідно технологій самонавчання, в якому для позиціювання апаратних пристроїв у тривимірному координатному просторі в реальному часі реалізовано самонавчання при побудові маршрутів їх руху. Це дало змогу покращити переміщення та синхронізацію між групою БПЛА або одного БПЛА в межах робочого сегменту і за рахунок цього збільшити кількість опрацьованих даних. Крім того, в ньому передбачена можливість накопичувати дані про минулі програмні місії, що дало змогу виконувати порівняння отриманих поточних результатів із минулими в режимі реального часу. Взаємодія реалізованого модуля синхронізації відеопотоків із програмним модулем планування маршрутів групи БПЛА на основі використання технологій обробки знань для здійснення самонавчання та організація їхньої

взаємодії в програмному просторі робочого середовища дала можливість забезпечувати оперативне отримання результатів та здійснювати виявлення та обчислення нових структурних об'єктів, які були пропущені в процесі минулих програмних місій.

Розроблені механізми поведінки групи БПЛА дають змогу динамічного отримання зображень структурних об'єктів однієї природи в робочій зоні у межах цільового фруктового дерева. Більш того, запропонована модель обробки даних із однієї робочої зони дала змогу реалізувати множинну обробку фруктових дерев базуючись на системі управління та планування маршрутів. Архітектура розподіленого багаторівневого програмного інтерфейсу управління групою БПЛА, який інтегровано в централізовану систему, дозволив динамічно отримувати зображення структурних об'єктів у програмній місії, яка представлена у вигляді множини рядів фруктових дерев однієї природи.

Розроблено метод динамічного отримання зображень заданих структурних об'єктів в тривимірному просторі за допомогою декількох БПЛА, який полягає в тому, що забезпечено в процесі активації кожного БПЛА генерацію відеоряду в центральну систему, прийняття рішень про подальшу роботу групи БПЛА, підтвердження цілісності програмної місії та визначення рівня критичності для продовження виконання роботи групи БПЛА за рахунок визначення станів модулів та комплексному врахуванні вимог розподілення, багаторівневості та автоматизованості, що покращує узгодження між різними БПЛА та досягнення переміщення всієї групи БПЛА від заданих початкових до кінцевих точок програмної місії самостійно.

Основні результати розділу опубліковані у працях [121-123, 127-130].

## РОЗДІЛ 3.

МОДЕЛІ ЗОБРАЖЕНЬ ЗАДАНИХ СТРУКТУРНИХ ОБ'ЄКТІВ В ЗАДАЧАХ  
РОЗПІЗНАВАННЯ ОБРАЗІВ

У природі всі об'єкти є унікальними та типізованими. Кожний структурний об'єкт характеризується певними властивостями. Характеристики цих властивостей або їхня відсутність розглядаються як ознаки об'єкта. У природі не існує двох і більше об'єктів, для яких збігаються всі ознаки, усі об'єкти є унікальними, а це дає змогу, теоретично, відрізнити один об'єкт від іншого. Часто ознаки різних об'єктів можуть збігатися, і тому ці об'єкти належать до одного класу, або типу.

Основним завданням розпізнавання образів є побудова, на основі експериментальних та систематичних теоретичних досліджень, ефективних обчислювальних засобів, щоб виявлені дані віднести до своїх відповідних класів. У теорії розпізнавання образів класом прийняти називати певну сукупність об'єктів, які мають спільні ознаки. Клас об'єднує фізичні сутності, але водночас може бути високорівневою абстракцією. Відрізнити певні ознаки одного класу від іншого, називають інформативними ознаками. А ознаки, які є спільними для всіх представників класу є інваріантами класу. Об'єкти, які пов'язані між собою певним відношенням еквівалентності, або толерантності. Для поняття толерантності транзитивне поняття не виконується, а виконується відношення “бути схожим”. У той самий час відношення еквівалентності є симетричним, транзитивним та рефлексивним. Не обов'язково, але деколи інформативні і інваріантні ознаки можуть збігатись.

У розділі розглянемо моделі та методи, за допомогою яких стане можливим реалізувати технологію для розпізнавання та обчислення кількості заданих структурних об'єктів, а також синхронізації відеопотоків в режимі реального часу.

### 3.1. Метод синхронізації відеопотоків в режимі реального часу

Особливістю автоматизованої багаторівневої системи виявлення та обчислення кількості структурних об'єктів однієї природи групою БПЛА є одночасне використання кількох апаратних пристроїв для захоплення відеорядів із цільовими структурними об'єктами. Запропонована автоматизована система забезпечує виявлення заданого структурного об'єкта групою БПЛА та його подальше врахування для обчислення кількості таких об'єктів строго один раз.

Процес синхронізації відеопотоків із різних БПЛА може бути ускладнений багатьма чинниками, наприклад, БПЛА можуть мати різні типи відеокамер, швидкість отримання відеорядів може різнитись, а самі відеоряди можуть спотворюватись або не отримуватись взагалі. Крім того, характеристики польоту та захоплення відео БПЛА від різних виробників можуть суттєво відрізнятися, що також може негативно відобразитись на якості виявлення та точності обчислення заданих структурних об'єктів. Тому, щоб не допустити подібні проблеми з отриманням відеорядів, запропоновано новий метод синхронізації відеопотоків у режимі реального часу. Розроблений метод полягає в поєднанні відеорядів, отриманих із кожного БПЛА в групі під час виконання робочої місії та автоматичному злитті відеокадрів в одне зображення фруктового дерева. Метод синхронізації відеопотоків реалізовано через певну кількість програмних блоків, що об'єднані в одну інформаційну систему формування поведінкової сигнатури виявлення та обчислення кількості заданих структурних об'єктів. Схему роботи розробленого методу зображено на рис. 3.1.

Метод складається з таких основних кроків:

1. Блок корегування швидкості передачі відеорядів. У процесі виконання програмної місії в робочих зонах група БПЛА утворює відеоряди, які транслюються через мережу в програмний модуль виявлення та обчислення кількості заданих структурних об'єктів. Відеоряди потрапляють у програмний модуль виявлення в різні моменти часу, що може бути зумовлене: 1) коливаннями

мережевого з'єднання, 2) типом камер БПЛА та 3) швидкістю переміщення БПЛА у робочій зоні та іншими чинниками.

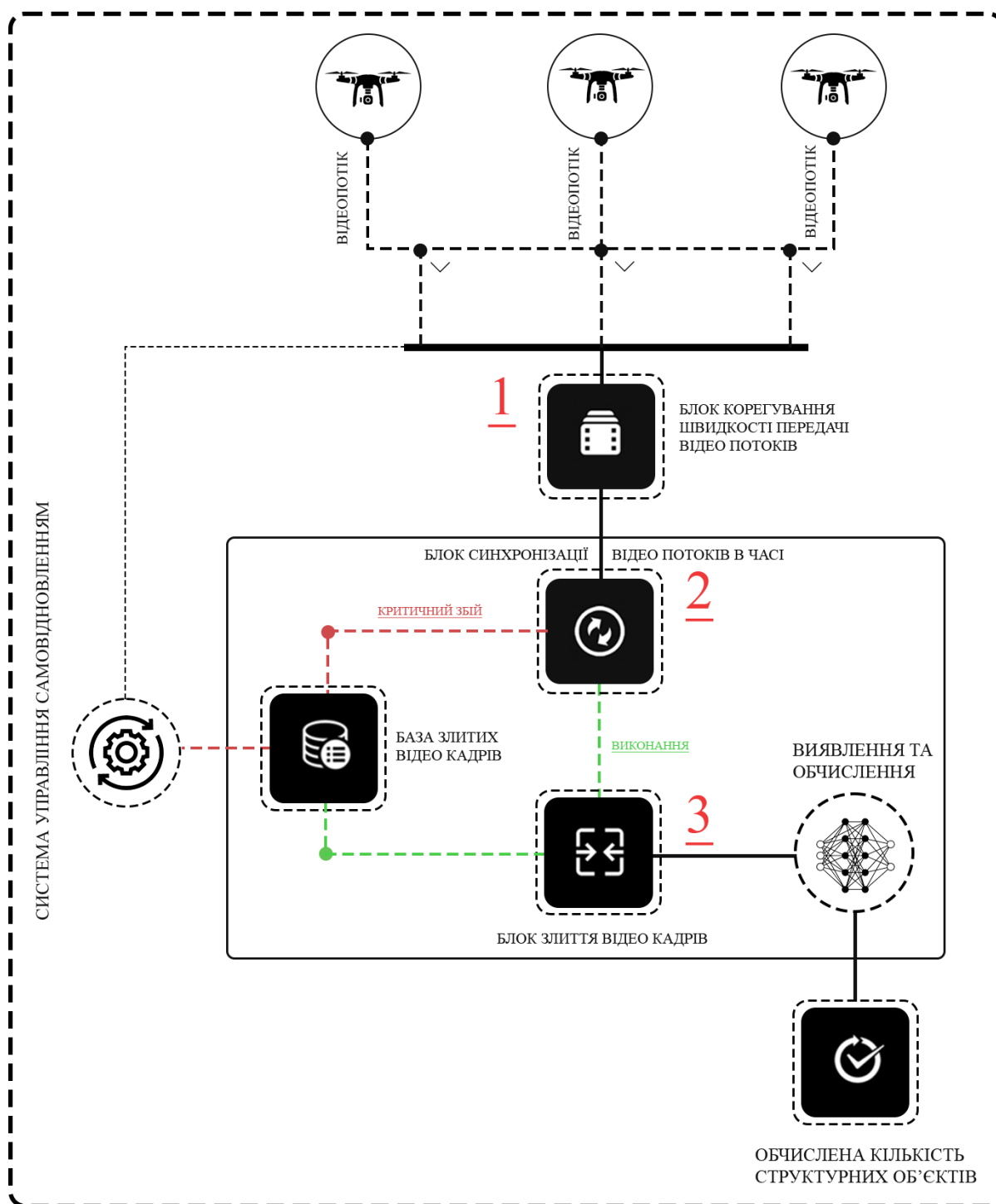


Рисунок 3.1 – Схема виконання синхронізації відеопотоків

З огляду на перераховані особливості переміщення відеопотоків у робочому середовищі, блок корегування швидкості утворює програмну структуру, яка

поєднує в собі механізми обробки швидкості передачі відеорядів та формує функціональність зупинки та отримання відеоряду. Перехід блока корегування швидкості передачі відеоряду в стан “в очікуванні” забезпечує отримання всіх відеорядів із кожного БПЛА в один момент часу.

Утворена програмна структура блоку має наступні характеристики: 1) унікальний ідентифікатор БПЛА; 2) множина байтів відеокадру; 3) частота кадрів відеоряду; 4) час отримання відеокадру блоком корегування швидкості з БПЛА; 5) формат кодування відеокадру. Результатом виконання блоку корегування швидкості є перехід блоку в стан “виконання” та передача утвореної множини програмних структур на наступний блок виконання.

2. Блок синхронізації відеопотоків у часі. Загальне функціонування блоку зумовлене кількістю множин програмних структур, отриманих від попереднього блоку щодо корегування швидкості передачі відеорядів. Першочергово блок синхронізації відеопотоків у часі здійснює перевірку кількості множини програмних структур, що призводить до оброблення наступних сценаріїв:

2.1) Якщо кількість отриманих множин програмних структур не збігається з кількістю БПЛА, то блок синхронізації переходить у критичний стан. Як наслідок, формується запит на отримання останньої збереженої програмної структури з блоку бази даних збережених злитих відеокадрів. Якщо остання збережена програмна структура міститься в базі, то вона передається запитом на модуль самовідновлення. Водночас, якщо такої структури в базі немає, то критичний запит формується відразу на стан аварійного завершення, оскільки подані характеристики свідчать про невиконання групою БПЛА робочої місії.

2.2) Якщо блок синхронізації відмітив успішну синхронізованість відеопотоків, то поведінкова сигнатура переходить у стан “розкадрування відеорядів”. Водночас кожен утворений відеокадр може мати унікальний формат кодування, що зумовлено різним апаратним забезпеченням БПЛА від різних виробників. Тому, для уніфікації відеокадрів вони перетворюються в спеціальний програмний формат як одне ціле зображення, що забезпечує функціонування блоку виявлення та обчислення структурних об’єктів. Розроблений похідний блок



декодує відеокадри в необхідний системний формат та формує програмні структури, час декодування яких фіксується та додається до структури. Отже, результатом виконання даного блоку є множина програмних структур, що слугують вхідними даними для наступного блоку злиття відеокадрів.

3. Блок злиття відеокадрів. На рис. 3.2 подано схему процесу виконання блоку злиття відеокадрів.

Спершу блок злиття відеокадрів перевіряє еквівалентність часу утворення множини програмних структур даних. Виконання перевірки часових характеристичних забезпечує цілісність сформованих даних під час утворення однієї програмної структури даних.

1. Якщо процес перевірки еквівалентності визначає, що подані програмні структури даних є сумісними подіями, то блок злиття відео кадрів переходить в етап перетворення та підготовки відео кадрів на передачу до нейронної мережі.

2. Якщо процес перевірки визначає нееквівалентність у часі, то блок злиття виконує запис у журнал невизначених програмних структур даних. Запис у журналі подій зумовлює перевірку всіх майбутніх вхідних структур даних із невизначеними програмними структурами даних. Процедура звірки виконується впродовж 1 хв, після чого журнал подій видаляє ті програмні структури даних, що зберігаються довше 1 хв.

З огляду на характеристичні ознаки вхідних параметрів нейронної мережі, програмні об'єкти, які містять конкретне знання про відеокадри, мають відповідати значенням геометричних параметрів базових предикатів. У процесі захоплення відеорядів, БПЛА з відеокамерами розташовуються на різних висотах у просторі робочого середовища й тому захоплюють цільові фруктові дерева під різним кутом. Водночас ширина захоплення об'єктива відеокамери є різною для БПЛА різних виробників. Також, виконання програмної місії групи БПЛА ведеться в динамічному робочому середовищу, де погодні умови та зокрема сила поривів вітру можуть змінюватись. Відповідно, погодні чинники можуть суттєво впливати на рівень стабілізації відеокамери і, як наслідок, спотворюють візуальну область у

робочій зоні. Тому, до отриманих із БПЛА відеокадрів застосовуються алгоритми алгебраїчного перетворення зображень.

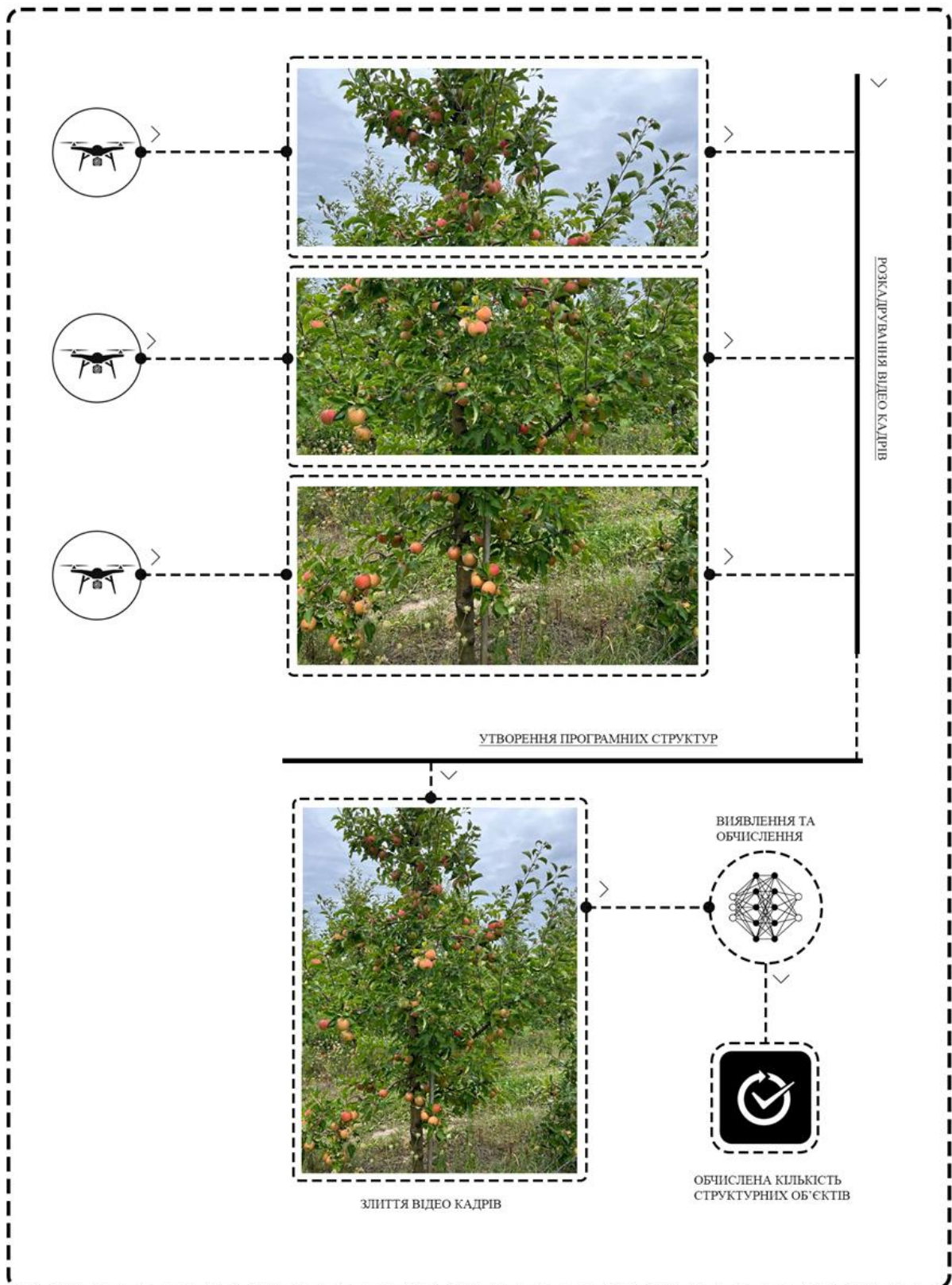


Рисунок 3.2 – Схема процесу блоку злиття відео кадрів

Подальше трансформування відеокадрів виконується за наступними кроками.

Крок 1. Множина байтів, що відображає розкадрований відеокадр, перетворюється у файл. Метадані про характерні особливості файлу перевіряються через програмні механізми через визначення сигнатур новоствореного файлу, зокрема, тип файлу, геометрія файлу, колірний простір, роздільна здатність та глибина каналу.

1.1. Здійснюється корегування кута повороту для множини зображень отриманих на кроці 1 з використанням афінного перетворення. Результатом корегування є трансформовані відеокадри, значення геометричних ознак яких збігаються.

1.2. Результати виконання перетворення зображень надсилаються до програмної функції злиття.

Крок 2. Злиття зображень. Сукупність вище описаних кроків дають змогу утворити одне зображення через злиття множини відкоригованих зображень.

2.1. З огляду на пере- або недоекспоновані області на відеокадрі та різну глибину різкості різних кадрів, злите зображення може містити графічні артефакти. Тому, до відеокадрів застосовується визначений у системі програмний фільтр для маскуванню переходів між ними. Як наслідок, такого фільтрування, злиття відеокадрів в одне ціле зображення відбувається без помітних переходів та із чіткими краями.

2.2. Відфільтровані відеокадри надсилаються в програмний механізм стиснення.

2.3. Процедура стиснення відеокадру відбувається на основі афінного перетворення. Характерними ознаками для параметрів геометричного перетворення слугують особливості вхідних даних алгоритмів нейронних мереж. Оскільки відеокадри, що отримуються через виконання програмної місії групою БПЛА, подаються прямокутною формою, то для забезпечення максимальної ефективності виявлення та обчислення кількості структурних об'єктів механізм стиснення зводить зображення до квадратної форми. Також, механізм стиснення зменшує вагу відеокадрів без зміни значення роздільної здатності та глибини

каналу завдяки використанню програмного фільтра на кроці 2.1. Результатом виконання такої процедури є підготовлена до злиття множина відеокадрів.

2.4. Множина підготовлених відеокадрів на кроці 2.3 надсилається до програмної функції злиття.

2.5. Злиття декількох відеокадрів в одне цілісне зображення полягає в утворенні об'єктної програмної структури даних матричного вигляду. Кожен елемент матриці відповідає значенню коду кольору одного графічного пікселя. Утворена в такий спосіб матриця даних відображає суцільне плодове дерево, де всі відокремлені відеокадри, отримані від різних БПЛА, зливаються в одне ціле.

2.6. Отримані на кроці 2.5 дані, заносяться до внутрішньої бази злитих відеокадрів.

2.7. Матрична програмна структура даних надсилається до програмного модуля виявлення та обчислення кількості структурних об'єктів однієї природи.

Результатом успішного виконання блоку злиття відеокадрів є отримання на виході програмної структури даних у вигляді матриці кодів кольору. Розроблений елемент функціональності блока злиття щодо занесення даних у внутрішню базу злитих відеокадрів забезпечує цілісність системи у випадку її критичного збою.

Перевагою розробленого методу над відомими є його можливість накопичувати дані про минулі програмні місії, що дає змогу виконувати порівняння отриманих поточних результатів із минулими в режимі реального часу. Крім того, взаємодія реалізованого модуля синхронізації відеопотоків із програмним модулем планування маршрутів групи БПЛА згідно використання технологій обробки знань для здійснення самонавчання та організація їхньої взаємодії в програмному просторі робочого середовища дає можливість забезпечувати оперативне отримання результатів та здійснювати виявлення та обчислення нових заданих структурних об'єктів, які були пропущені в процесі минулих програмних місій.

З огляду на використання механізмів перетворення зображень у блоці злиття відеокадрів, автоматизована система в автоматичному режимі обробляє всі відеопотоки, які отримуються в процесі виконання програмної місії групою БПЛА.

У випадку критичних збоїв, які спотворюють цілісність структур даних із джерел отримання відеопотоків, система зберігає такі структури в спеціальному журналі помилок та не відправляє їх як вхідні параметри в наступну обробку. Як наслідок, компонент синхронізації відеопотоків забезпечує цілісність даних та обмежує систему від спотвореної інформації.

3.2. Метод та засоби виявлення, відстеження та обчислення кількості заданих структурних об'єктів

Метою роботи є підвищення точності розпізнавання та обчислення кількості заданих структурних об'єктів, зокрема плодів на деревах фруктового саду. Для досягнення мети в роботі вдосконалено метод виявлення заданих структурних об'єктів, що є формальним відображенням плодів на фруктових деревах та запропоновано спосіб обчислення кількості цих об'єктів.

3.2.1 Метод виявлення заданих структурних об'єктів

Автоматичне виявлення плодів фруктових дерев відбувається в складному сіткоподібному середовищі, де плоди ростуть на деревах, що розміщені послідовними рядами. Під час виконання робочої місії плоди можуть бути захоплені відеокамерою БПЛА та подані на передньому плані робочого сегмента (фруктове дерево), на задньому плані робочого сегмента або на деревах інших рядів. Як наслідок, плоди на відеокадрі можуть бути надто малими, що призведе до їхнього помилкового виявлення автоматизованою системою, або вони можуть потрапити на різні відеокадри під час виконання робочої місії, що призведе до багаторазового підрахування тих же плодів. Також, змінні погодні умови, зміна освітлення впродовж дня, потрапляння в кадр листя та гілок можуть погіршувати точність виявлення плодів. Тому, для недопущення та усунення вищевказаних проблем, вдосконалено метод виявлення плодів заданих структурних об'єктів, що слугують формальним відображенням плодів на фруктових деревах.

Вхідними даними методу є цифрове зображення заданих структурних об'єктів, що подане у вигляді матриці чисел, кожний елемент якої відображає інтенсивність відповідного пікселя в зображенні. Модель цифрового зображення структурних об'єктів однієї природи подамо, як лінійне відображення:

$$f: I \rightarrow \langle C_{xy}, P_{h \times w}^i, B_{u \times v}^i, B_{h \times w}^i \rangle, \quad (3.1)$$

де  $C_{xy} = (c_x, c_y)$  – ширина ( $c_x$ ) та висота ( $c_y$ ) координатної сітки  $C_{xy}$  координатної площини  $XU$ , утвореної для виявлення структурних об'єктів,  $P_{h \times w}^i = (p_h^i, p_w^i)$  – висота ( $p_h^i$ ) та ширина ( $p_w^i$ ) прив'язувальної рамки  $i$ -го цільового структурного об'єкта в координатній площині  $XU$ ,  $B_{u \times v}^i = (b_u^i, b_v^i)$  – координати центру прив'язувальної рамки  $i$ -го цільового структурного об'єкта в координатній площині  $XU$ ,  $B_{h \times w}^i = (b_h^i, b_w^i)$  – висота ( $b_h^i$ ) та ширина ( $b_w^i$ ) обмежувальної рамки, що окреслює  $i$ -ий цільовий структурний об'єкт у координатній площині  $XU$ .

Координати центру обмежувальної рамки  $i$ -го цільового заданого структурного об'єкта за (3.1) в координатній площині  $XU$  обчислюються за формулою, що була модифікована із формули (1.2):

$$\begin{aligned} b_u^i &= \sigma^i(t_x^i + c_x); \\ b_v^i &= \sigma^i(t_y^i + c_y), \end{aligned} \quad (3.2)$$

де  $t_x^i$  та  $t_y^i$  – зміщення центру обмежувальної рамки  $i$ -го цільового структурного об'єкта відносно координатної сітки  $C_{xy}$ .

Висота ( $b_h^i$ ) та ширина ( $b_w^i$ ) обмежувальної рамки із формули (3.2)  $i$ -го цільового структурного об'єкта за (3.1) в координатній площині  $XU$  обраховуються за формулою, що була модифікована з формули (1.3):

$$\begin{aligned} b_h^i &= p_h^i e^{c_y}; \\ b_w^i &= p_w^i e^{c_x}. \end{aligned} \quad (3.3)$$

Математичну формалізацію цільового об'єкта за формулами (3.1)-(3.3) із накладеною прив'язувальною рамкою та прогнозними значеннями розмірів обмежувальної рамки на її основі зображено на рис. 3.3

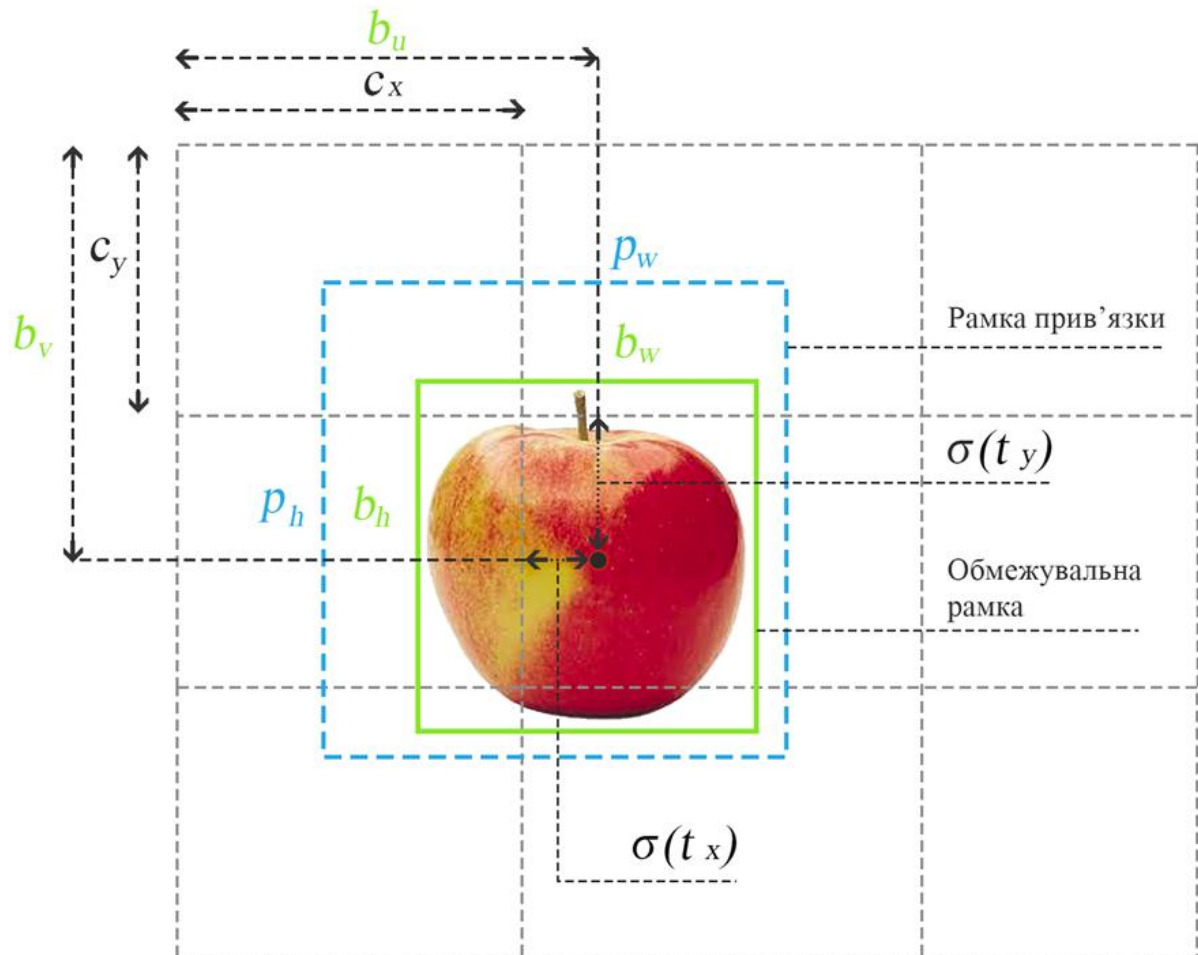


Рисунок 3.3 – Схема подання прив'язувальної рамки та прогнозування значень розмірів обмежувальної рамки на її основі

За механізм накладання прив'язувальної та обмежувальної рамок на цільовий об'єкт взято неймережеву модель на основі архітектури ЗНМ під назвою YOLOv5 (див. п. 1.2.2), до якої внесено вдосконалення. Модифіковану архітектуру назвемо YOLOv5-v1 (рис. 3.4).

Початкові розміри прив'язувальної рамки, що накладається на об'єкт на цифровому зображенні за малих і середніх масштабів, змінено. Це зроблено для підвищення точності виявлення кожного структурного об'єкта на зображенні.

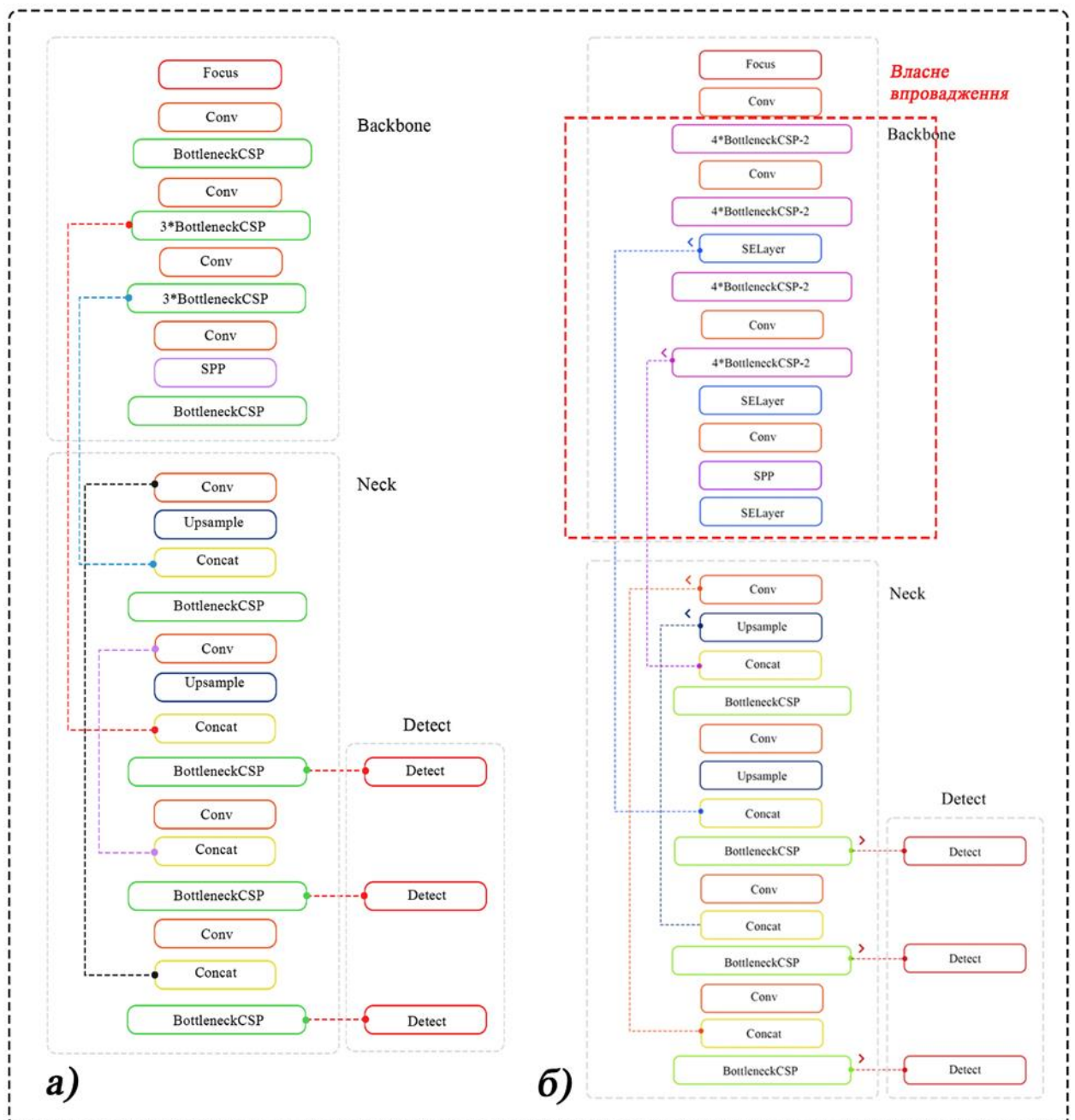


Рисунок 3.4 – Порівняння схем а) класичної архітектури YOLOv5 та б) вдосконаленої архітектури YOLOv5-v1

Нижче подано ініціалізацію значень розмірів обмежувальних та прив'язувальних рамок структурних об'єктів, які потрапляють у відеокадр. Використання автоматизованої системи в реальних умовах передбачає три основні сценарії розміщення заданих структурних об'єктів на зображенні: а) на передньому плані поточного робочого сегмента, б) на задньому плані поточного робочого сегмента, в) на іншому робочому сегменті, що потрапив у кадр. Прив'язувальні



рамки, які мають три розміри в оригінальній архітектурі YOLOv5, встановлюються для кожної карти ознак з ініціалізованими коефіцієнтами, що подано в табл. 3.1.

Таблиця 3.1

Значення початкових розмірів обмежувальних та прив'язувальних рамок оригінальної архітектури YOLOv5

Рамки	Карта ознак 1		Карта ознак 2		Карта ознак 3	
	$p_h$	$p_w$	$p_h$	$p_w$	$p_h$	$p_w$
Обмежувальні рамки	20	20	40	40	80	80
Прив'язувальні рамки, малі	10	13	16	30	33	23
Прив'язувальні рамки, середні	30	61	62	45	59	199
Прив'язувальні рамки, великі	116	90	156	198	373	326

Три карти ознак із табл. 3.1 є вхідними даними для мультिकанального шару вилучення ознак архітектури YOLOv5, який призначений для виявлення цільових об'єктів різного розміру.

З аналізу предметної області (див. п. 1.3.3) встановлено, що відстань між плодами на задньому плані, які розміщені на деревах у віддаленому ряду посадки та БПЛА може бути вкрай великою, що може негативно вплинути на якість виявлення структурних об'єктів. Тому, відповідні їм задані структурні об'єкти на зображенні варто вважати слабо прийнятними цілями для виявлення та обчислення. Щоб запобігти помилковому виявленню структурних об'єктів на задньому плані відеокадру та підвищити точність їхнього виявлення на передньому плані, початкові розміри прив'язувальних рамок малого та середнього масштабу було змінено, з огляду на умови робочого середовища та результати обчислювальних експериментів.

Відношення довжини до ширини початкових розмірів обмежувальної рамки та відношення довжини до ширини початкових розмірів прив'язувальної рамки

встановлено приблизно 1 до 1. Значення початкових розмірів обмежувальної та прив'язувальної рамок вдосконаленої архітектури YOLOv5-v1 подано в табл. 3.2.

Таблиця 3.2

Значення початкових розмірів обмежувальних та прив'язувальних рамок вдосконаленої архітектури YOLOv5-v1

Рамки	Карта ознак 1		Карта ознак 2		Карта ознак 3	
	$p_h$	$p_w$	$p_h$	$p_w$	$p_h$	$p_w$
Обмежувальні рамки	20	20	40	40	80	80
Прив'язувальні рамки, малі	80	70	75	75	85	100
Прив'язувальні рамки, середні	95	110	130	110	115	125
Прив'язувальні рамки, великі	116	90	156	198	373	326

Передбачається, що подання початкових розмірів обох типів рамок у табл. 3.2 дасть змогу підвищити точність виявлення структурних об'єктів на передньому плані та знизити ймовірність відстеження нецільових об'єктів на задньому плані.

Для досягнення високої точності виявлення структурних об'єктів у режимі реального часу фізичний розмір оригінальної нейромережевої моделі YOLOv5 має бути мінімальним, а структура нейронної мережі – відповідно максимально стиснутою. Тому, магістральний блок нейронної мережі, який вилучає ознаки цільового об'єкта на вхідних зображеннях, було модифіковано через зменшення кількості вагових параметрів нейронної мережі. Вдосконалення магістрального блоку полягає в наступних змінах.

1. Модифіковано модуль фокусування нейронної мережі для зменшення часу навчання нейронної мережі (рис. 3.5), а саме:

а) вхідне зображення, що подане трьома каналами RGB та розміром  $3 \times 640 \times 640$ , було розбито на 4 частини розмірами  $3 \times 320 \times 320$  кожна;

б) 4 частини зображення об'єднано в одну карту ознак за RGB каналом із результируючим розміром  $12 \times 320 \times 320$ ;

в) до отриманої карти ознак розміром  $12 \times 320 \times 320$  застосовано операцію згортки із 32 ядрами, яка генерує карту ознак розміром  $32 \times 320 \times 320$ ;

г) до отриманої карти ознак розміром  $32 \times 320 \times 320$  застосовано операцію пакетної нормалізації та активаційну функцію Hardswish; результируюча карта ознак розміром  $32 \times 320 \times 320$  передається на наступний етап оброблення нейронної мережі.

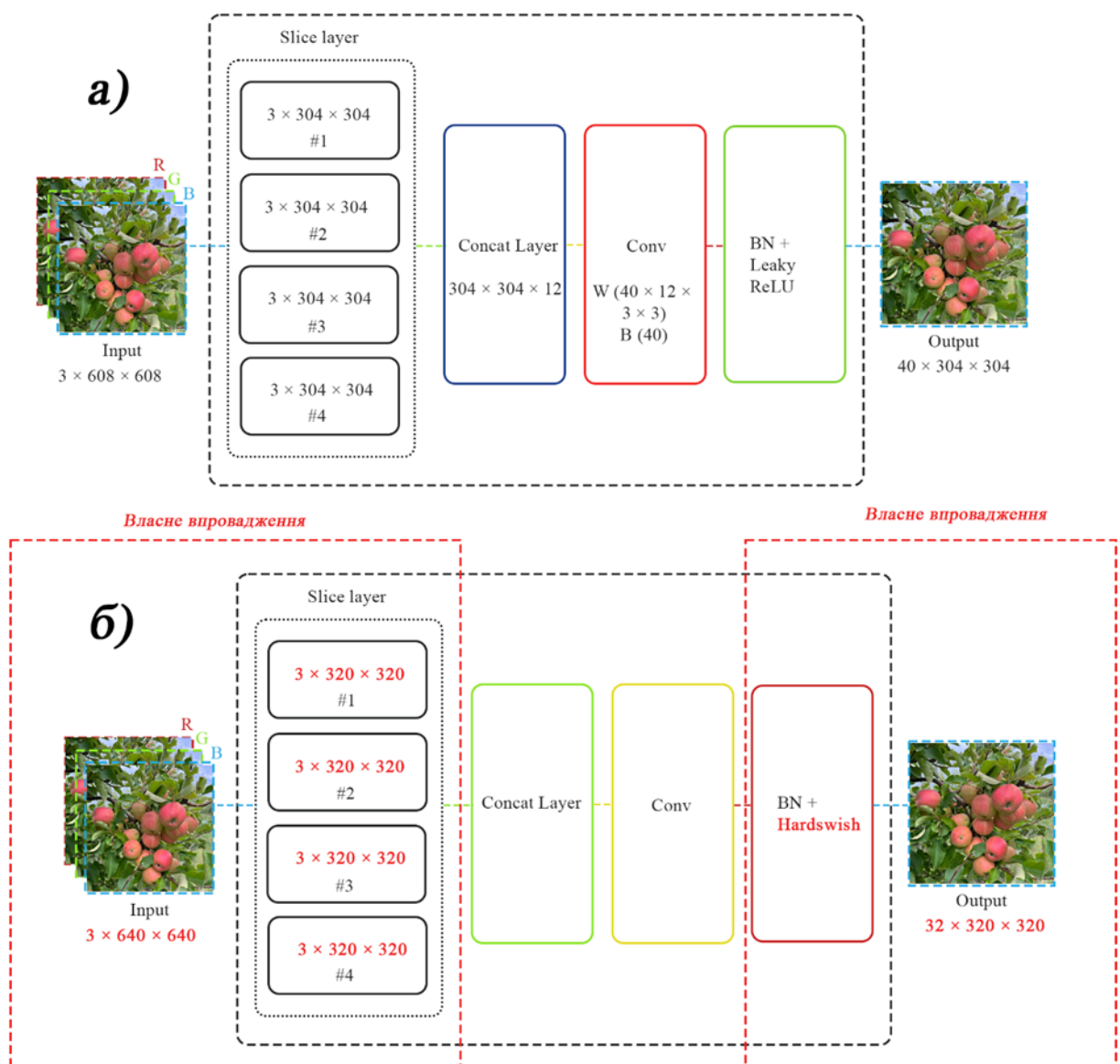


Рисунок 3.5 – Порівняння схем модуля фокусування (focus module) класичної архітектури YOLOv5 а) та вдосконаленої архітектури YOLOv5-v1 б)

2. Модуль BottleneckCSPv1. Вдосконалення модуля BottleneckCSPv1 в архітектурі YOLOv5-v1 (див. рис. 3.5) полягає у видаленні згорткового шару, що в оригінальному модулі BottleneckCSP поєднував вхідну карту ознак з операцією конкатенції (рис. 3.6).

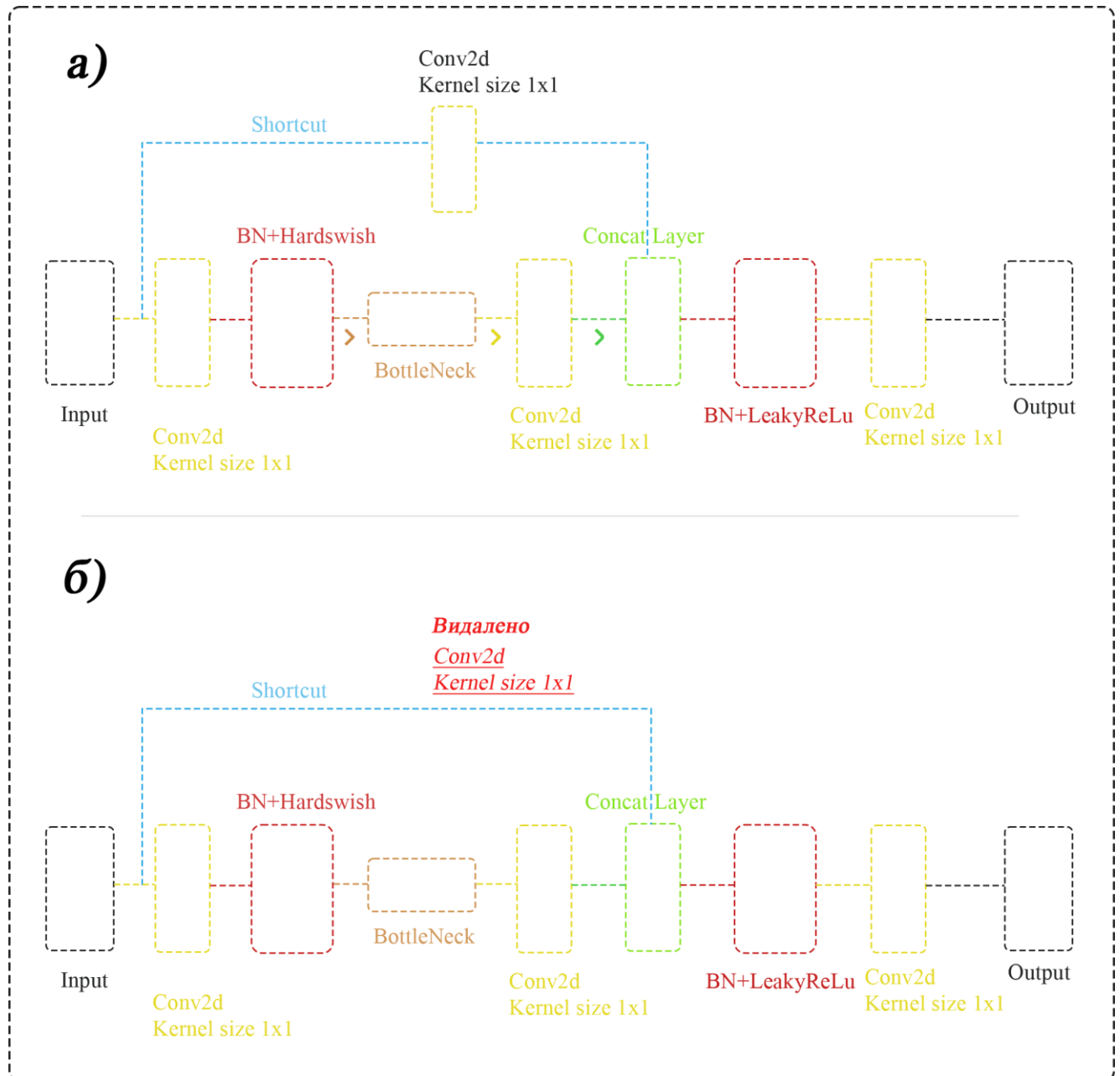


Рисунок 3.6 – Порівняння схем модуля BottleneckCSPv1 класичної архітектури YOLOv5-v1 а) та вдосконаленої архітектури YOLOv5 б)

У вдосконаленому модулі BottleneckCSPv1 вхідна карта ознак безпосередньо пов'язується з вихідною картою ознак у глибину, що дало змогу зменшити

кількість параметрів у модулі. Модифікований BottleneckCSPv1 має обмеження, що полягає в недостатньому вилученні ознак цільових об'єктів на зображенні через зменшення кількості глибоких згорткових операцій. Для усунення цього обмеження в архітектурі YOLOv5-v1 чотири оригінальних модулі BottleneckCSP замінено на модифіковані BottleneckCSPv1.

3. Модуль SENET. Форма й колір плодів на фруктових деревах відрізняються від фонових об'єктів на відеокадрі. Тому, для підвищення точності виявлення відповідних їм структурних об'єктів на відеокадрі в архітектурі YOLOv5-v1 використано механізм візуальної уваги (поданий як модуль SENET) для вилучення ознак, що притаманні саме фруктовому плоду. Механізм візуальної уваги полягає в повторному калібруванні ознак через навчання нейронної мережі. Як наслідок, більш значущі ознаки, що притаманні саме фруктовому плоду, посилюються, а менш значущі – пригнічуються. Модуль SENET вбудовано в магістральну структуру архітектури YOLOv5-v1.

4. Об'єднання карт ознак. Щоб підвищити точність виявлення структурних об'єктів на відеокадрах, до архітектури YOLOv5-v1 застосовано операцію об'єднання карт ознак різних розмірів в одну карту ознак із вищою дискримінаційною здатністю. З огляду на спрощену структуру модифікованого модуля BottleneckCSPv1 та об'єднання карт ознак, що виводяться зі згорткових шарів вдосконаленої архітектури YOLOv5-v1, об'єднання шарів 4 і 15, 6 і 11, 10 і 21 оригінальної архітектури YOLOv5 було замінено на об'єднання шарів 5 і 18, 8 і 14, 13 і 24 вдосконаленої архітектури YOLOv5-v1.

З аналізу зображень фруктових дерев (див. п. 1.2.1) встановлено, що структурні об'єкти на зображенні переважно належать до середнього розміру. Також, вихідна карта ознак 23-го шару вдосконаленої архітектури YOLOv5-v1 є вхідною картою ознак для функції класифікації цільових об'єктів середнього розміру. Тому, для зменшення часу виявлення цільових структурних об'єктів під час використання YOLOv5-v1, вихідні карти ознак 14-го та 21-го шарів вдосконаленої архітектури об'єднано між собою.

В результаті виконаних модифікацій отримано вдосконалений метод, який полягає в автоматизованому перетворенні вхідної інформації, що подана як цифрове зображення структурних об'єктів (плодів фруктових дерев), у результуючу інформацію у вигляді множини координат, що відповідають розташуванню цих об'єктів на зображенні.

### 3.2.2 Обчислення кількості заданих структурних об'єктів

Для успішного обчислення кількості плодів за відеопотоком необхідним є встановлення унікальності кожного плоду на двох послідовних відеокадрах, отриманих від БПЛА. Відстеження унікальних плодів є нетривіальним завданням, оскільки їхнє розташування та зовнішній вигляд можуть змінюватися на послідовних відеокадрах з огляду на умови проведення робочої місії, наприклад, зміна освітлення, повороти камери, зміна траєкторії роботи БПЛА тощо. Тому, актуальним та необхідним видається розроблення нового методу для обчислення кількості структурних об'єктів, що відповідають плодам на кожному кадрі відеоряду. Запропонований метод обчислення у вигляді вхідних даних отримує множину координат, що відповідають розташуванню структурних об'єктів на зображенні та виводить кількість плодів, що були виявлені під час захоплення відеоряду.

Відстеження унікальних структурних об'єктів на різних відеокадрах є нетривіальним завданням. Визначення унікальності ґрунтується на характерних відмінних ознаках кожного об'єкта, їхньому положенні на послідовних кадрах через зміну траєкторії кожного БПЛА в групі та непередбачувані чинники довкілля. Тому, особливістю нового методу обчислення кількості структурних об'єктів є встановлення залежності між візуальними ознаками виявлених структурних об'єктів, що відображаються на послідовних відеокадрах. Схему способу обчислення кількості структурних об'єктів однієї природи подано на рис. 3.7.

Запропонований спосіб полягає в послідовному виконанні таких блоків: 1) попередня обробка вхідних даних у вигляді об'єднаних відеокадрів із накладеними обмежувальними рамками виявлених структурних об'єктів,

2) першочергове порівняння споріднених координат виявлених об'єктів між двома сусідніми кадрами, 3) присвоєння унікальних ідентифікаторів нововиявленим об'єктам, 4) обчислення кількості унікальних структурних об'єктів, 5) обчислення прогнозних значень координат розташування об'єктів для наступного кадру та б) обробка вихідних даних, якими є обчислена кількість структурних об'єктів у робочому середовищі.

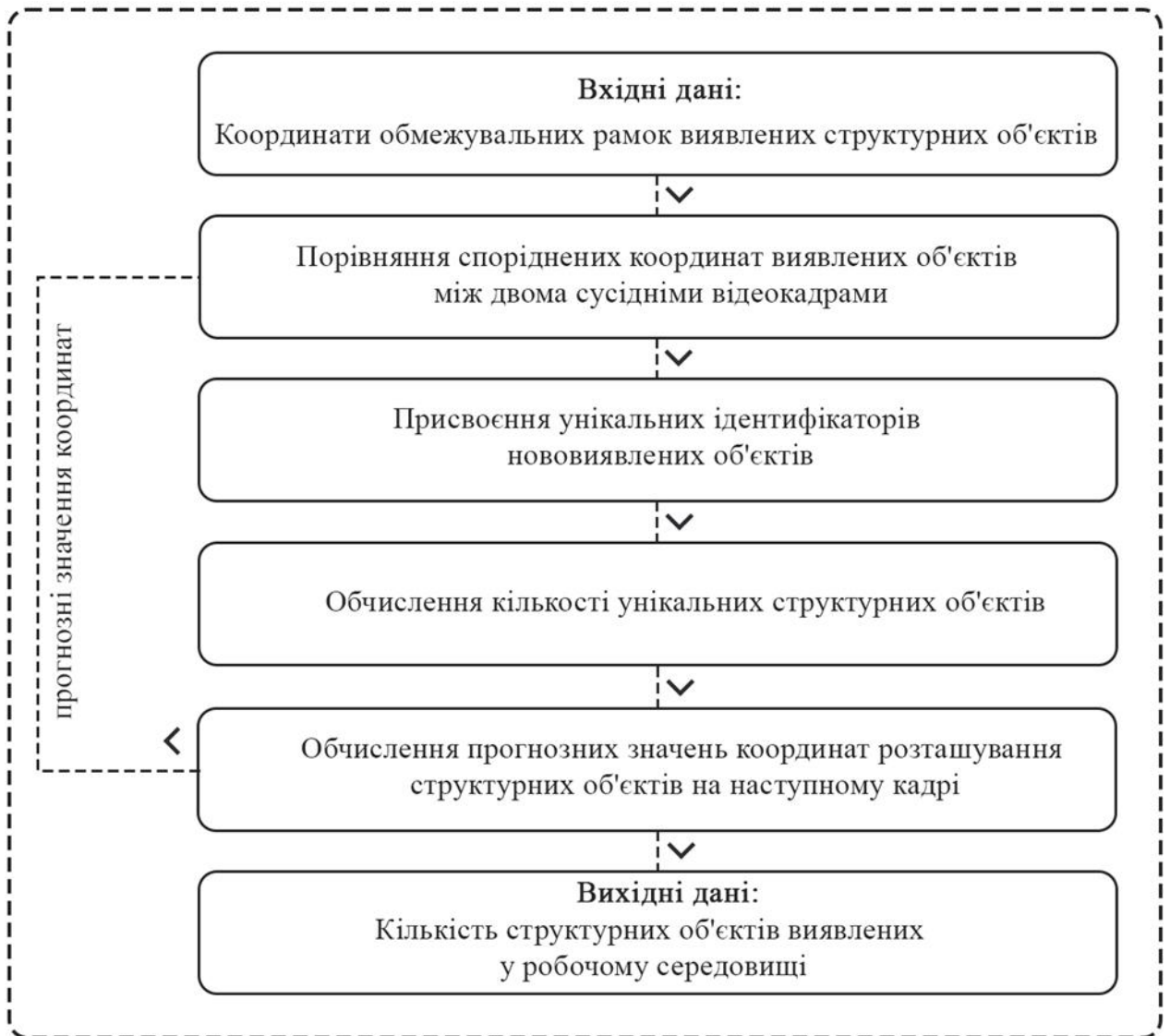


Рис. 3.7 – Схема кроків способу обчислення кількості структурних об'єктів однієї природи

Блок 1. Попередню обробку вхідних даних виконано, відповідно до впроваджених та модернізованих алгоритмів у методі виявлення, де необхідним є

сегментація малих, середніх та великих структурних об'єктів. Малі об'єкти вважаються нецільовими для відстеження, тому вони не підлягають обчисленню. Поведінковою сигнатурою поточного блоку є фільтрування даних через вилучення обмежувальних рамок виявлених малих структурних об'єктів. Утворена нова структура даних передається на наступний блок.

Блок 2. Першочергове порівняння споріднених координат виявлених об'єктів між двома сусідніми кадрами. На основі отриманої чисельної інформації з блоку 1 формується стан виявлених структурних об'єктів на поточному відео кадрі, кожен із яких представляється у вигляді:

$$(x, y, \gamma, h, v_x, v_y, v_\gamma, v_h), \quad (3.4)$$

де  $(x, y)$  – координати центра виявленого об'єкта,  $\gamma$  – відношення ширини до висоти обмежувальної рамки виявленого об'єкта,  $v_x, v_y, v_\gamma, v_h$  – швидкість зміни відстежувальної рамки (tracking box) між послідовними відеокадрами.

Виявлених об'єктів на одному відеокадрі може бути багато, тому доцільним є формування групи ймовірностей розташування кожного виявленого об'єкта для наступному кадрі. Введення характерних груп поданих у вигляді виявлених та відстежуваних структурних об'єктів за формулою (3.4) дає змогу формалізувати простір, у якому постійно формується нова група підмножин на основі алгебраїчних структур. Група підмножин, що відображає прогнозоване розташування кожного виявленого об'єкта, обчислене на попередньому кадрі, формується в результаті застосування фільтра Калмана (див. п. 1.2.3): стан руху кожного об'єкта на попередньому кадрі оцінює інформацію про модель руху структурного об'єкта на поточному відеокадрі, що дає можливість отримати оптимальну оцінку загального стану послідовності кадрів. Угорський алгоритм використано для обчислення матриць відстаней між спорідненими координатами розташування виявлених об'єктів та прогнозованих, що отримані з попереднього відеокадру. Обчислення відстаней ґрунтується на дисперсії вхідних змінних та кореляції між ними.



Позначимо через  $r_j$  матрицю характерних ознак кожного виявленого структурного об'єкта у відеокадрі, через  $R_j$  – стан усіх числових характеристик виявлених об'єктів із попередніх 100 кадрів. Відповідно мінімальну відстань між  $i$ -ою ознакою прогнозованого об'єкта з попереднього відеокадру та  $j$ -єю ознакою виявленого структурного об'єкта в поточному відеокадрі обчислюємо так:

$$d_{(i,j)}^{(1)} = \min \left\{ 1 - r_j^T r_i^{(k)} \mid r_i^{(k)} \in R_i, k \in (1,100) \right\}. \quad (3.5)$$

Поєднання двох структурних об'єктів за відстанню за формулою (3.5) визначається перетином характерних ознак, що визначається за формулою з контрольним порогом  $t^{(1)}$ :

$$b_{(i,j)}^{(1)} = \begin{cases} 1, & d_{(i,j)}^{(1)} \leq t^{(1)}; \\ 0, & d_{(i,j)}^{(1)} > t^{(1)}. \end{cases} \quad (3.6)$$

Результатом функції (3.6) є значення дисперсії 1, якщо характеристичні ознаки перетинаються, 0 – якщо такого перетину немає.

Значення та характеристики руху відстежувального об'єкта між послідовними відеокадрами в багатовимірному просторі виразимо через значення квадрата відстані Махаланобіса (див. п. 1.2.3) між прогнозованим та виявленим розташуванням структурного об'єкта на поточному кадрі:

$$d_{(i,j)}^{(1)} = (d_j - y_i)^T S_i^{-1} (d_j - y_i), \quad (3.7)$$

де  $y_i$  – прогнозні значення обмежувальної рамки  $i$ -ого об'єкта на поточному відеокадрі,  $d_j$  – обмежувальні рамки  $j$ -ого цільового об'єкта на поточному відеокадрі, а  $S_i$  – коваріаційна матриця для  $i$ -ого об'єкта на поточному відеокадрі.

Визначення процесу поєднання за відстанню (3.7) із встановленим контрольним порогом  $t^{(2)}$  подамо за формулою:

$$b_{(i,j)}^{(2)} = \begin{cases} 1, & d_{(i,j)}^{(2)} \leq t^{(2)}; \\ 0, & d_{(i,j)}^{(2)} > t^{(2)}. \end{cases} \quad (3.8)$$

Отримавши значення характеристики руху відстежувального об'єкта між послідовними відеокадрами за формулами (3.7-3.8), сформуємо чисельну матрицю, на основі відстані Махаланобіса, яка визначатиме чи перетинаються два показники у вибраній області інтересу

$$c_{i,j} = \lambda d_{(i,j)}^{(1)} + (1 - \lambda) d_{(i,j)}^{(2)}; \\ b_{i,j} = \prod_{m=1}^2 b_{(i,j)}^{(m)}. \quad (3.9)$$

Чисельна матриця (3.9) є результатом блоку 2.

Відстеження в режимі реального часу відбувається відповідно до каскадного підходу, що полягає в багаторазовому перевірці відповідності між розташуванням прогнозованих та виявлених структурних об'єктів. Відстеження множини виявлених об'єктів на відеокадрі виконується через показник IOU (див. п. 1.2.3).

Відстеження продовжується допоки значення IOU для кожного виявленого об'єкта набуває найвищого чисельного значення,  $0 \leq IOU \leq 1$ . Процес відстеження в режимі реального часу ілюстровано на рис. 3.8.

Результатом поточного блоку є матриця станів (3.9) відстежених об'єктів, що передається на наступний блок.

Блок 3. Присвоєння унікальних ідентифікаторів нововиявлених об'єктів. У цьому блоці відбувається додавання нового характерного стану унікальності кожного виявленого об'єкта до стану розташування виявлених об'єктів, отриманого з блоку 2.

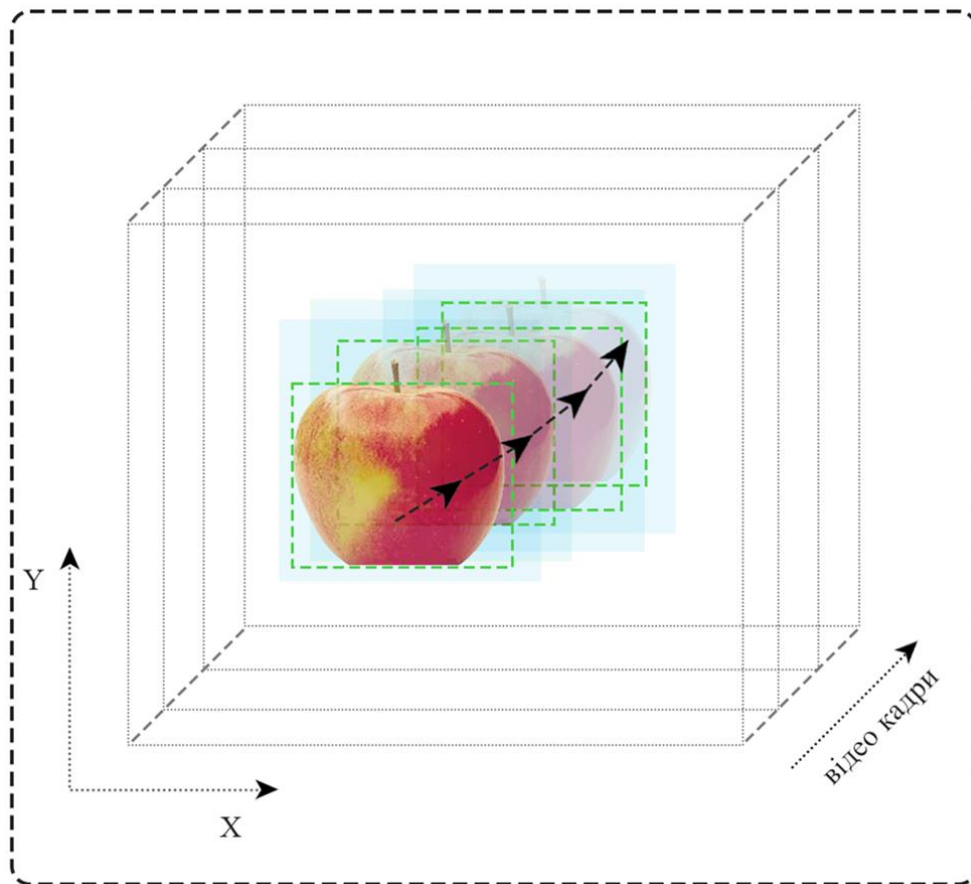


Рисунок 3.8 – Схема процесу відстеження заданого структурного об’єкта в режимі реального часу

Блок 4. Обчислення кількості унікальних структурних об’єктів. Блок отримує на вхід реєстр структурних об’єктів, яка містить тимчасові дані про відстежувальний об’єкт: його тип, рух, унікальний ідентифікатор, стан активності та час запису до реєстру. Допоки об’єкт перебуває в системі відстеження, він вважається “активним”. Якщо об’єкт зникає з наступного відеокадру, система формує подію, за якої його стан змінюється на “неактивний” у системному реєстрі. “Неактивний” стан виявленого структурного об’єкта визначає, що такий об’єкт буде доданий у множину обчислених об’єктів. Водночас доданий об’єкт не видаляється із системного реєстру.

Блок 5. Обчислення прогнозних значень координат розташування об’єктів у наступного кадру. У поточному блоці формується множина прогнозних значень майбутнього розташування структурних об’єктів у наступних послідовних відеокадрах. Якщо об’єкт є присутнім на 100 наступних послідовних відеокадрах,

у цьому часовому проміжку між початковим та кінцевим відеокадром об'єкт вважається ідентичним та унікальним. Приклад визначення ідентичних структурних об'єктів подано на рис. 3.9.

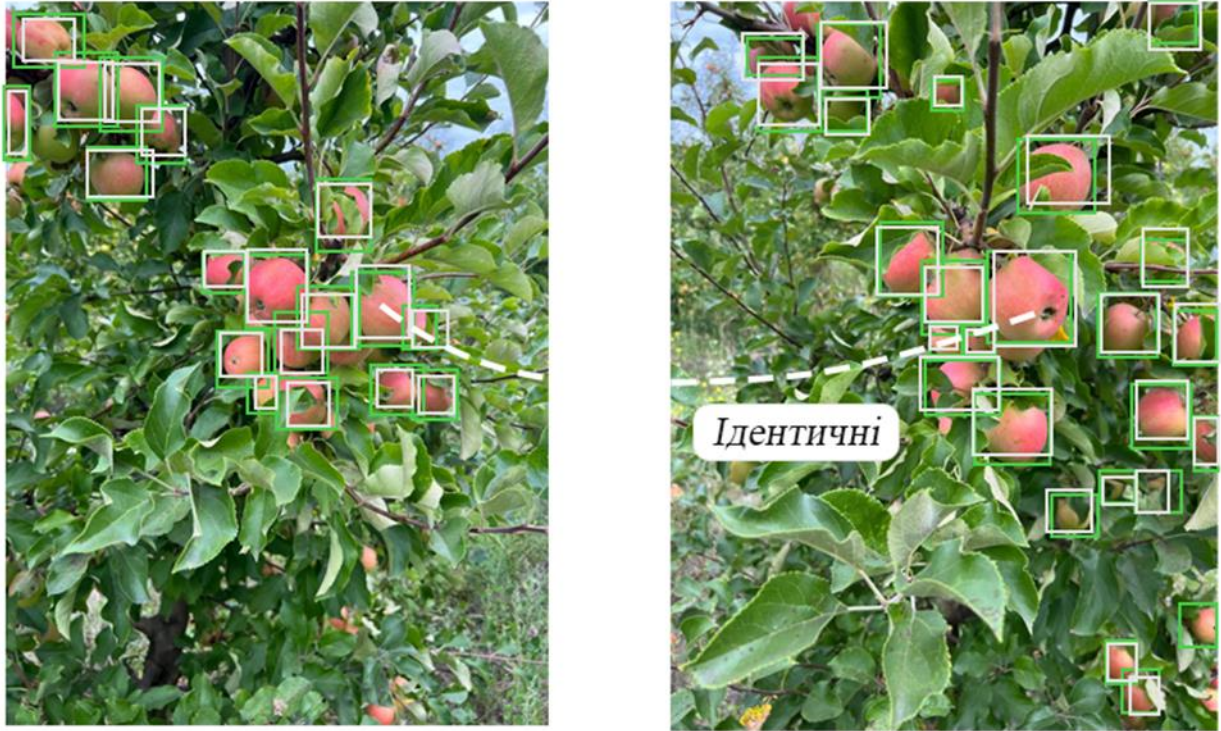


Рис. 3.9 – Приклад ідентичності під час відстеження структурних об'єктів

Також, відбувається формування системної події, яка утворює інформаційний запис цільового об'єкта у тимчасовий системний реєстр відстежувальних об'єктів, де він вважається успішно виявленим та заноситься у загальну статистику робочої місії групи БПЛА.

Прогнозні значення обчислюються за лінійною модифікацією фільтра Калмана з дискретним часом. З огляду на складні умови відстеження структурних об'єктів у робочому середовищі, блок 5 може мати два випадки результируючих даних, а саме:

Випадок 1. Якщо блок 5 обробляє перший відеокадр, то множина всіх виявлених об'єктів обробляється фільтром Калмана для прогнозування розташування на наступному відеокадрі.

Випадок 2. На поточному кадрі та 50 наступних відеокадрах структурний об'єкт може бути виявлений помилково через наявність шумів. Оброблення цього випадку виконує програмний компонент, який фільтрує відеокадри так, щоб відстеження продовжувалося змінною траєкторією відеокадрів допоки метод не видалить відстеження раніше виявленого об'єкту та занесе його до загальної статистики робочої місії.

Таким чином, розроблено спосіб обчислення кількості структурних об'єктів.

### 3.3. Створення детектора у задачі відстеження

Ефективність процесу виявлення та обчислення структурних об'єктів в експериментальному робочому середовищу суттєво залежить, зокрема, і від підготовчого етапу навчальних даних. Схема побудови детектора структурних об'єктів продемонстрована на рис. 3.7.

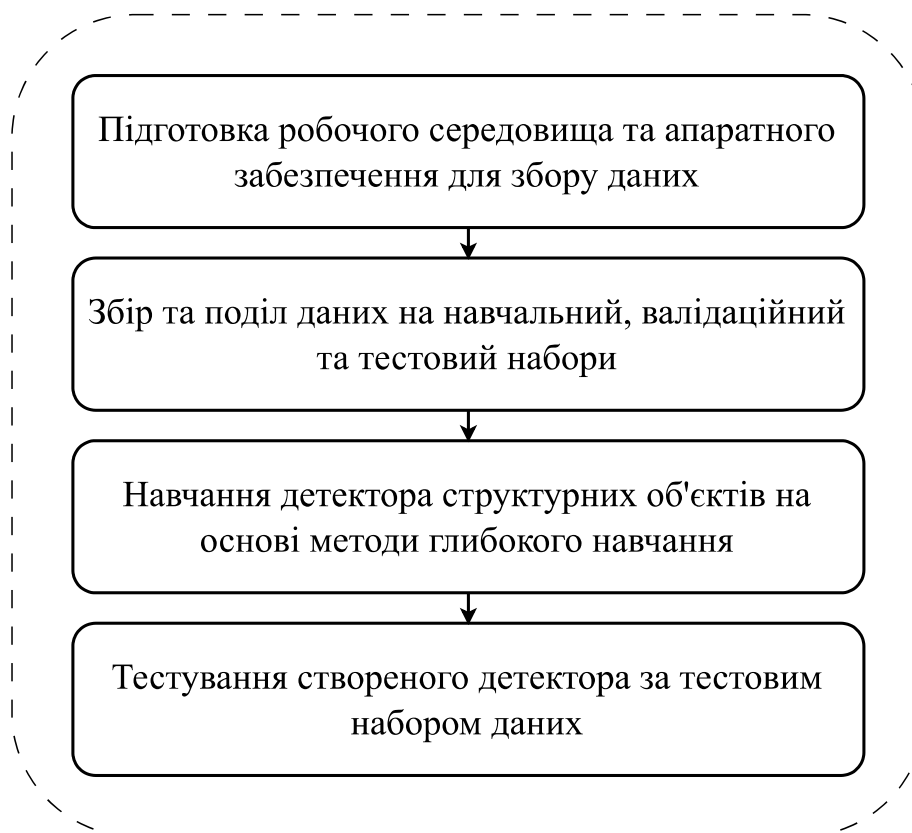


Рисунок 3.7 – Схема реалізації детектора у задачі розпізнавання образів

### 3.3.1 Підготовка робочого середовища

Для підготовки та формування навчальних даних було використано робоче середовище сучасного фруктового саду, яке знаходяться в Хмельницькій області, Городоцького району за координатами 49.265011, 26.548304 відповідно. Фото локації подано на рис. 3.8.



Рисунок 3.8 – Приклад зон робочого середовища: а) робоча зона, б) ряд польоту групою БПЛА

Відстань між рядами яблунь становить 4 м, відстань між плодовими деревами – близько 1 м, а висота дерев – близько 2 м. Також, в експериментальному полі встановлено базову RTK-станцію, яка складається з GNSS-приймача, супутникової антени, що посиляє через GPRS-з'єднання сигнал для підвищення точності звичайного супутникового позиціонування до 2.5 см. У робочому середовищі також встановлено засоби мережевого зв'язку та джерело безперебійного живлення, що дало змогу розраховувати на безперервний мережевий зв'язок з групою БПЛА.

### 3.3.2 Підготовка навчальних наборів даних

Для навчання нейронних мереж використано фруктові плоди яблуні Фуджі, що покриті штрихованим малиновим рум'янцем із жовтими підшкірними цятками

та мають округлену форму. Оригінальні зображення яблуневих плодів було отримано в сонячні та похмурі дні. Для отримання зображень було використано камеру смартфона iPhone 13 Pro Max [132] за різних погодних умов та на різних відстанях відносно дерев (0,5-1,5 м).

Детектор структурних об'єктів зосереджені має працювати в умовах робочого середовища, де органічні об'єкти закриті листям за різного освітлення та за різних природних умов. Тому, для розширення покриття сценаріїв використання автоматизованої системи, в експериментальному робочому середовищі відзнято 1 214 зображень яблук за таких умов: 1) яблука закриті листям, 2) яблука закриті гілками, 3) змішана оклюзія, 4) перекриття між яблуками, 5) кут природного освітлення, 6) кут заднього освітлення та 7) кут бокового світла (рис. 3.9).



Рисунок 3.9 – Отримані зображення плодів в різних умовах

Перших 200 зображень (100 сонячних днів і 100 похмурих днів) були випадковим чином відібрані з 1 214 зображень, як тестовий набір даних, а решта 1 014 зображень – як навчальний набір. У табл. 3.3 наведено розподіл створеного набору даних зображень плодів.

До відзнятих зображень застосовано механізми приросту та покращення даних з використанням технології CLAUD.AI [133], а саме:

- а) підвищення та зменшення яскравості зображення;
- б) горизонтальне віддзеркалення;
- в) вертикальне віддзеркалення;
- г) обертання під різними кутами.

Таблиця 3.3

Детальна інформація про попередню підготовку даних.

Дані	Сонячні	Похмурі	Загалом
Кількість зображень	100	100	200
Зображення із яскраво вираженими плодами	482	525	1007
Зображення із змішаною оклюзією	766	563	1329

Також, були добавлені зображення яблуневих плодів інших кольорів та форм. Фінальний набір даних мав 16 224 зображень плодів, які використовуються як дані остаточного навчального набору для навчання моделі розпізнавання, включаючи 15 210 покращених зображень і 1014 необроблених зображень. Навчальний та тестовий набори даних не містили збігів.

Наступним етапом після підготовки та обробки даних було додавання однієї, або декілька значущих та інформативних міток для надання контексту на зображенні. Для навчання моделі підготовлено позначений набір даних з накладеними мітками. З огляду на велику кількість зображень у зібраному наборі даних у дисертаційній роботі використано автоматичне маркування даних на основі Amazon SageMaker Data Labeling [133]. На рис. 3.10 подано оригінальне зображення плодів та зображення з накладеними мітками.

В результаті використання Amazon SageMaker Data Labeling відібрано 200 тестових зображень, які були позначені мітками у ручному режимі. Решта зображень отримувала мітки вже в автоматичному режимі.



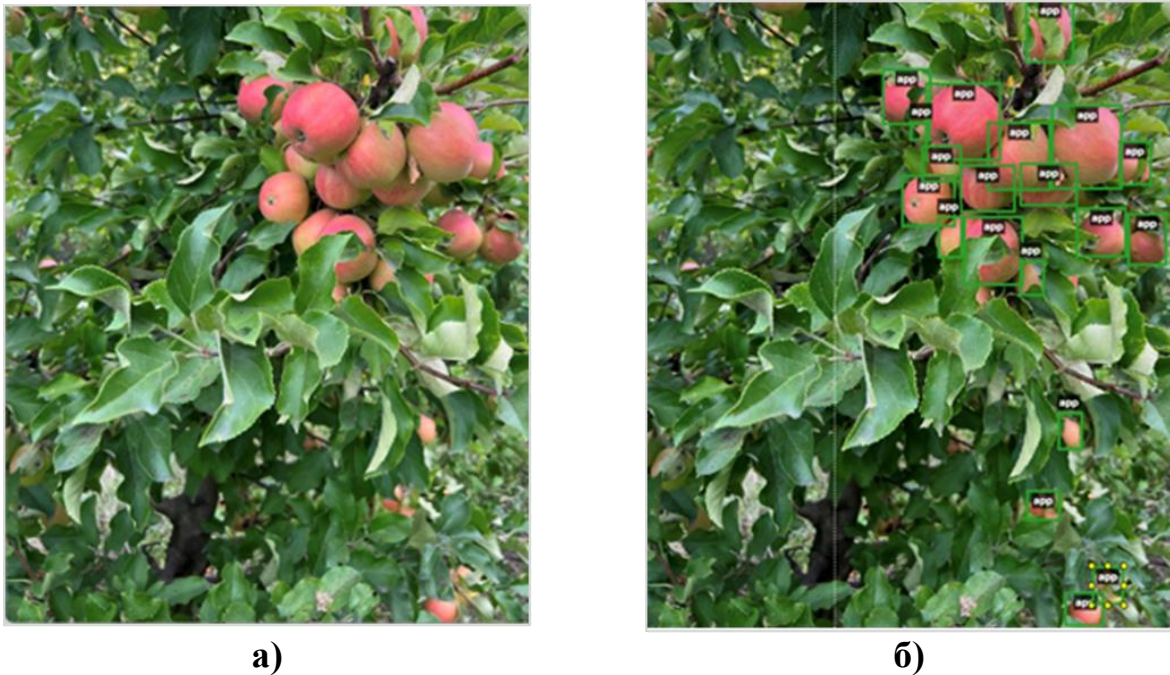


Рисунок 3.10 – Зображення плодів під час додавання міток у ручному режимі: а) оригінальне, б) з накладеними мітками

Навчання нейронної мережі та експериментальне тестування системи виконано з допомогою апаратного забезпечення: процесор Intel (R) Core (TM) I7-9750H, 2,6 ГГц, 32 ГБ оперативної пам'яті; графічний процесор NVIDIA Geforce RTX 2060, 6 ГБ відеопам'яті. Використане програмне забезпечення включало мову програмування Python [13534], програмний комплекс PyTorch [136] та пакет інструментів CUDA [137], cudnn [138], OpenCV [138] під операційною системою Windows 11.

### 3.3.3 Створення детектора

За метрики оцінювання побудованого детектора взято: 1) середня точність класифікації (mAP) (1.6), 2) precision (1.7), 3) recall (1.8), та 4) F<sub>1</sub>-міра (1.9). Результати навчання нейронної мережі за вдосконаленою архітектурою YOLOv5-v1 подано на рис. 3.11.

Значення функцій втрат за валідаційним набором даних і значення функцій втрат за навчальним набором даних показані кривими на рис. 3.11, вказують на те, що значення функції втрат швидко зменшуються на перших 100 ітерацій навчання

нейронної мережі, при цьому більш стабільна тенденція отримується після 250 ітерацій навчання. Таким чином, вихідна модель нейронної мережі утворюється після 300 ітерацій навчання. Показані значення діаграми на рис. 3.11 показують, що модель нейронної мережі добре навчена без перенавчання.

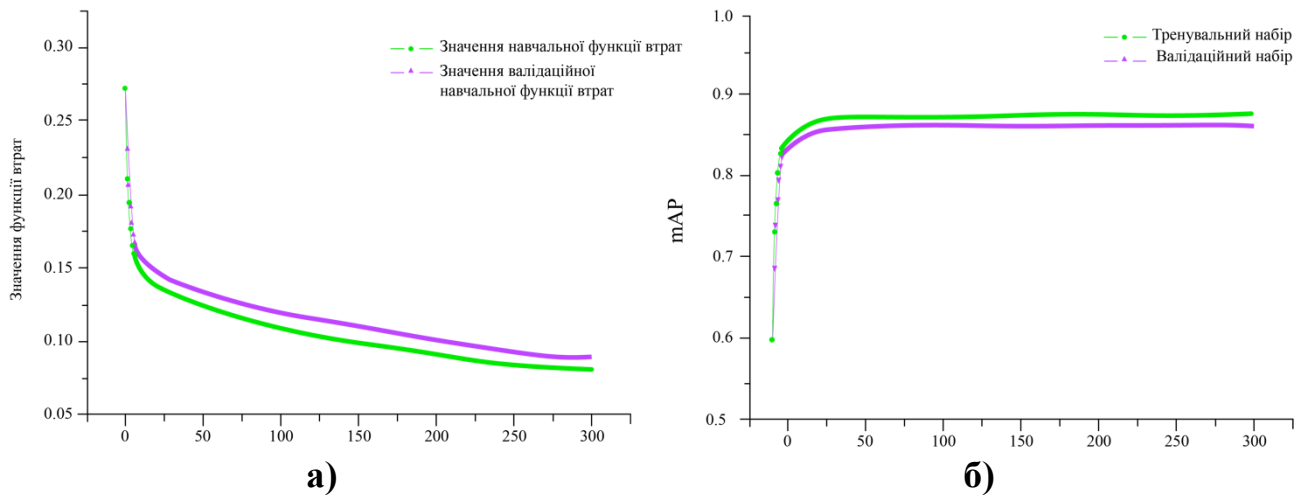


Рисунок 3.11 – Результати навчання мережі: а) втрати під час навчання та перевірки, б) середня точність класифікації навчальних наборів

Модель нейронної мережі після виявлення певної множини структурних об'єктів на відеокадрі, виконує фільтрування за допомогою попередньо встановленого значення порогу вірогідності прогнозування. Влучність та повнота результатів виявлення різняться беручи за основу одну модель розпізнавання, але різне значення порогів ймовірностей розпізнавання. Якщо поріг достовірності моделі не було встановлено належним чином, то отримуються непередбачувані результати, як показано на рис. 3.12.

Об'єкти, які були виявлені на передньому фоні вважаються помилковими (позначено жовтим еліпсом на рис. 3.12а), у випадку якщо значення порогу вірогідності було встановлено занадто низько, при цьому цільовий структурний об'єкт на передньому плані не буде виявлено, або з помилками (позначено жовтим еліпсом на рис. 3.12б), якщо значення порогу ймовірності буде встановлено надто високим.

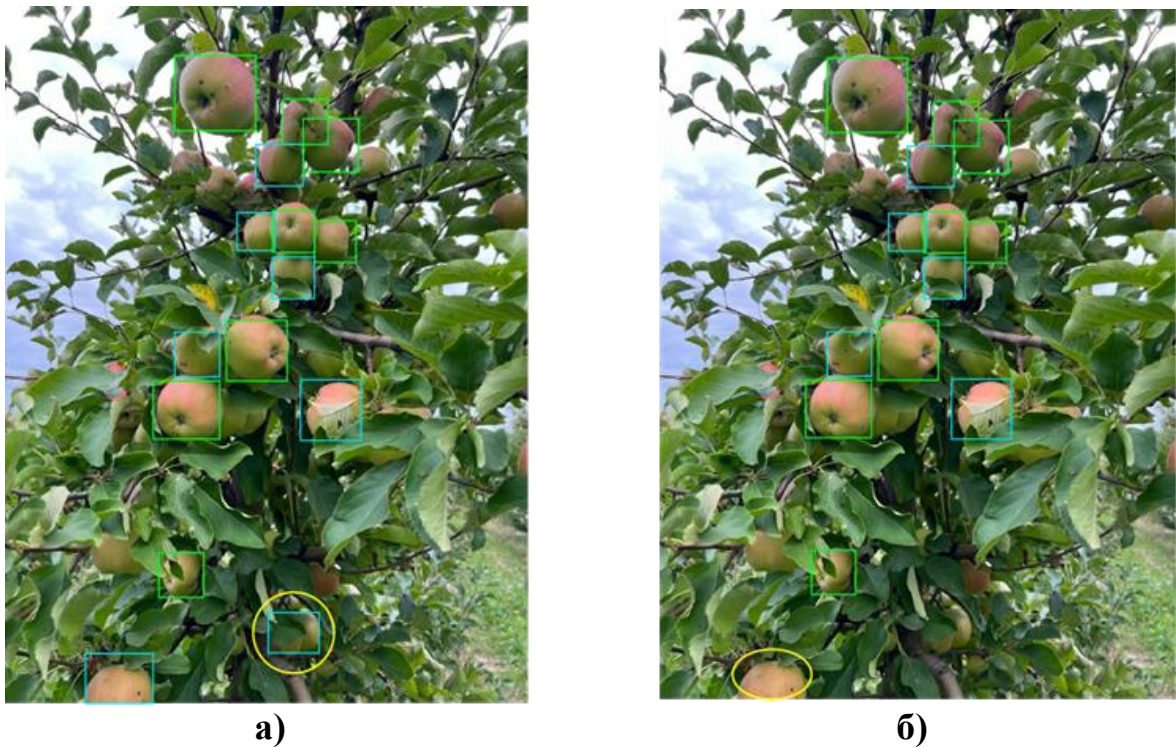


Рисунок 3.12 – Вплив порогу вирогідності на результат виявлення: (а) поріг занадто низький, б) поріг занадто високий

Результати від зміни параметрів ефективності моделі з різним порогом довіри ілюстровано на рис. 3.13.

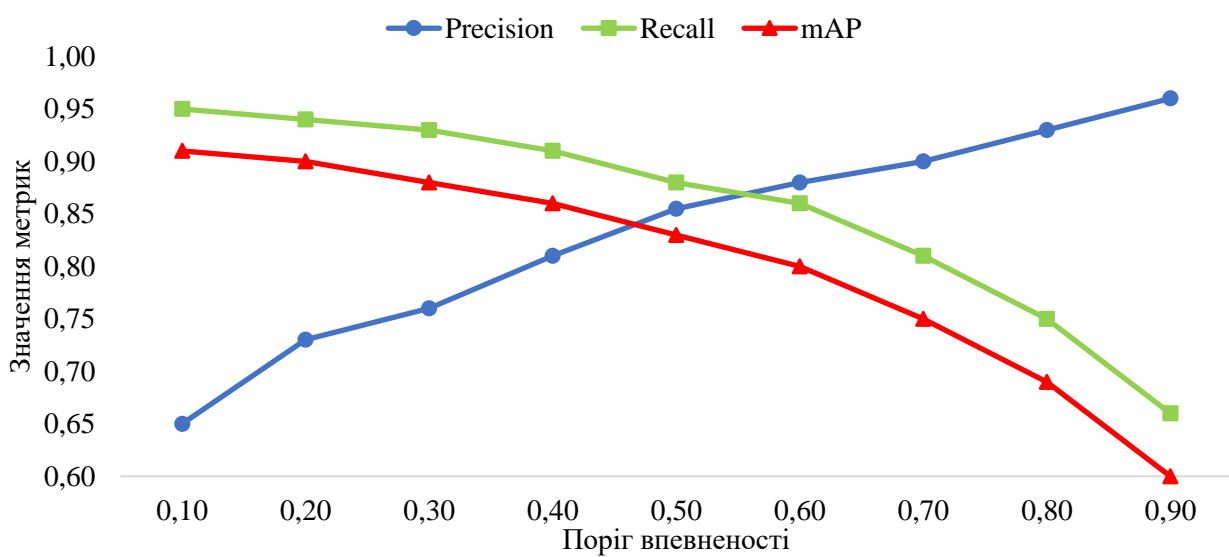


Рисунок 3.13 – Зміни параметрів ефективності моделі з різним порогом довіри

Значення порогу достовірності визначено згідно таких сценаріїв:

1. Порогове значення достовірності було встановлено нижче 0.5. У цьому випадку значення влучності розпізнавання була низькою та становило менше, а ніж 80%.

2. Порогове значення достовірності було встановлено вище 0.5. У цьому випадку значення середньої точності класифікації поступово знижувався.

3. Порогове значення ймовірності було встановлено 0.5.

Водночас ефективність та продуктивність моделі розпізнавання образів є найкращою. Значення середньої точності класифікації об'єктів становить 83,83%, 91,48% та 86,75% відповідно.

#### 3.4. Тестування та результати створеного детектора

Перевірка продуктивності та ефективності моделі нейронної мережі у задачі по розпізнаванню образів у режимі реального часу на основі вдосконаленого дизайну YOLOv5-v1, були додатково проаналізовані на результатах виявлення використовуючи 200 зображень тестового набору. Всього є 2336 зображень з фруктами, серед яких цільова кількість, які можна захопити, становить 1007, а цільова кількість фруктів, які неможливо захопити, – 1329. Специфічні результати розпізнавання методу, запропонованого в дослідженні, показано в табл. 3.4.

Таблиця 3.4

Результати методу з виявлення заданих об'єктів.

Дані	Кількість об'єктів	Precision (%)	Recall (%)	mAP (%)	F <sub>1</sub> (%)
К-сть, яку можна захопити	1007	85,51	94,33	89,23	89,70
К-сть, яку не можна захопити	1329	82,56	89,32	84,87	85,81
Загалом	2336	83,83	91,48	86,75	87,49

Результати із табл. 3.4 вказують на те, що для фруктів, які можна захопити, точність, запам'ятовування, значення mAP і оцінка  $F_1$  запропонованої моделі становили 85,51%, 94,33%, 89,23% і 89,70%, відповідно; для незрозумілих плодів, результати розпізнавання становили 82,56%, 89,32%, 84,87% і 85,81% відповідно. Загальне значення влучності виявлення структурних об'єктів, повноти, mAP і  $F_1$  становили 83,83%, 91,48%, 86,75% і 87,49% відповідно.

Приклади результатів розпізнавання запропонованої моделі нейронної мережі фруктів, які можна захопити, і фруктів, які не можна захопити, за різних погодних умов і умов освітлення показані на рис. 3.14.

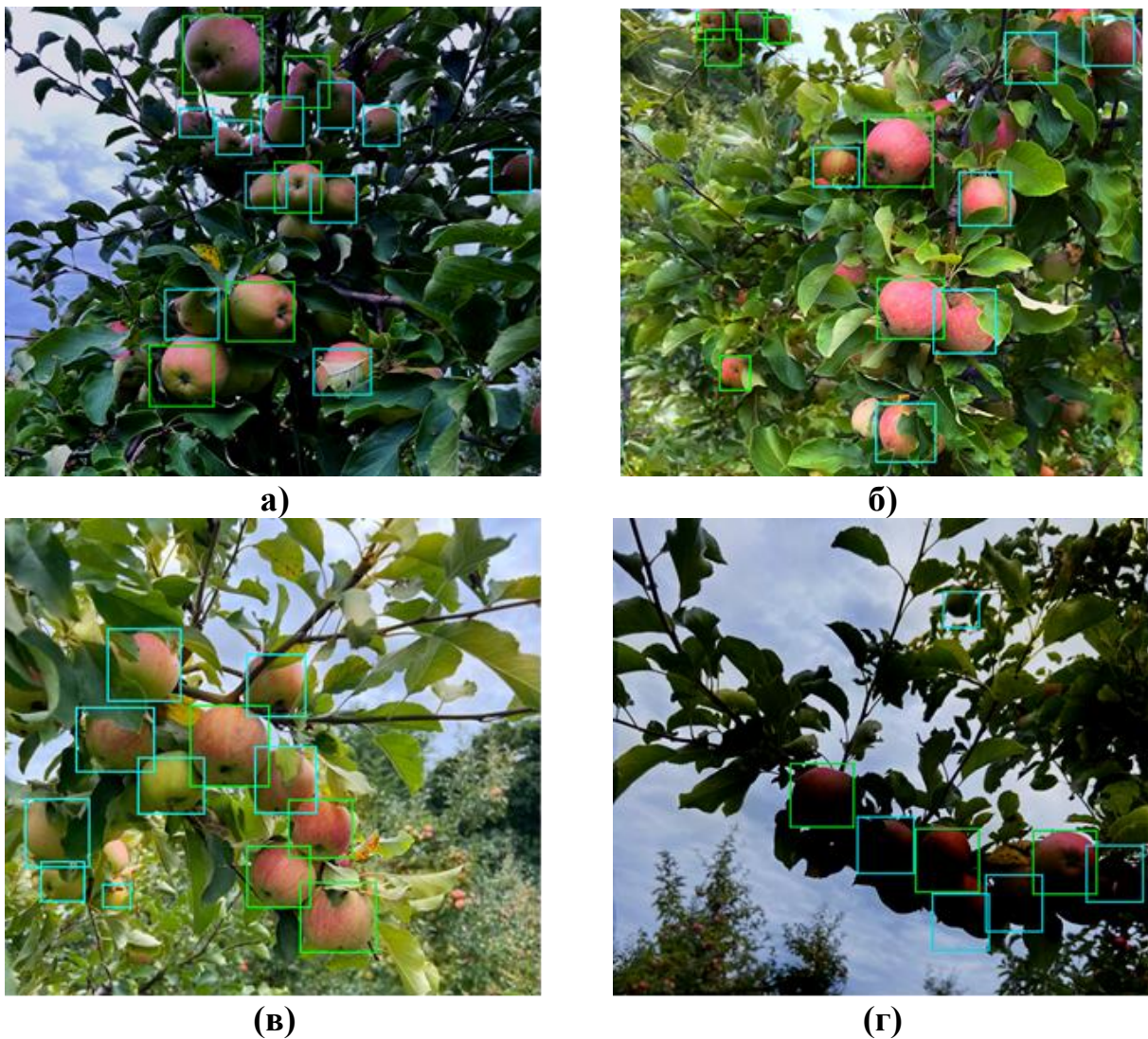


Рисунок 3.14 – Результати розпізнавання фруктових плодів за допомогою вдосконаленої мережі YOLOv5-v1. а) Хмарно б) Бічне сонячне світло в) Підсвічування сонцем г) Сонячний фронт

Обмежувальні рамки зеленого кольору використовувались, як мітки, які можна захопити, тоді як сині використовувались, як мітки, які неможливо захопити. Як видно на рис. 3.14, запропонована модель нейронної мережі виявляє структурні об'єкти на зображеннях знятих при рівномірному освітленні, похмурому освітленні, а також при сонячному освітленні. Крім цього, поточна модель виявляє структурні об'єкти в умовах переднього, заднього та бічного освітлення при сонячному освітленні.

Розглянемо порівняння результатів розпізнавання з використанням різних алгоритмів виявлення об'єктів.

Для подальшого аналізу ефективності розпізнавання запропонованого алгоритму, удосконалену мережу YOLOv5-v1 було порівняно на 200 зображеннях тестового набору з технологіями-аналогами, а саме:

- а) Технологія 1: містить в собі оригінальну мережу YOLOv5;
- б) Технологія 2: містить в собі архітектуру YOLOv3;
- в) Технологія 3: містить в собі архітектуру YOLOv4;
- г) Технологія 4: містить в собі архітектуру EfficientDet-D0.

Оціночними показниками були прийняті значення та середня швидкість розпізнавання моделі. Результати виявлення, розмір і кількість параметрів кожної моделі мережі наведені в табл. 3.5.

Таблиця 3.5

## Результати порівняння

Мережі виявлення об'єктів	mAP (%)	Середня швидкість виявлення (s/pic)	Кількість параметрів	Розмір моделі (Мб)
Технологія 1	81,70	0,013	$7,25 \times 10^6$	14,0
Технологія 2	71,80	0,053	$6,15 \times 10^7$	235,0
Технологія 3	82,01	0,017	$6,39 \times 10^7$	244,0
Технологія 4	80,02	0,038	$3,83 \times 10^6$	15,0
YOLOv5-v1	86,75	0,015	$6,52 \times 10^6$	12,7

Згідно з табл. 3.5, значення mAP для вдосконаленої моделі розпізнавання YOLOv5-v1, запропонованої в дослідженні, було найвищим, і воно було вищим, ніж у «Технології 1», і вищим, ніж у «Технології 2», «Технології 3» і «Технології 4» відповідно, що вказує на те, що запропонована технологія є найкращою для розпізнавання структурних об'єктів серед п'яти інших.

Для швидкості розпізнавання моделі середня швидкість виявлення вдосконаленої моделі YOLOv5-v1 за зображенням (66,7 кадрів/с), яка становила 2,53 і час у всіх інших технологіях відповідно, вказуючи, що запропонована модель нейронної мережі може задовольнити вимоги групи БПЛА для розпізнавання плодів у режимі реального часу.

Приклади результатів ідентифікації плодів за хмарних і сонячних умов запропонованою нейромережевою моделлю YOLOv5-v1 показані на рис. 3.15.

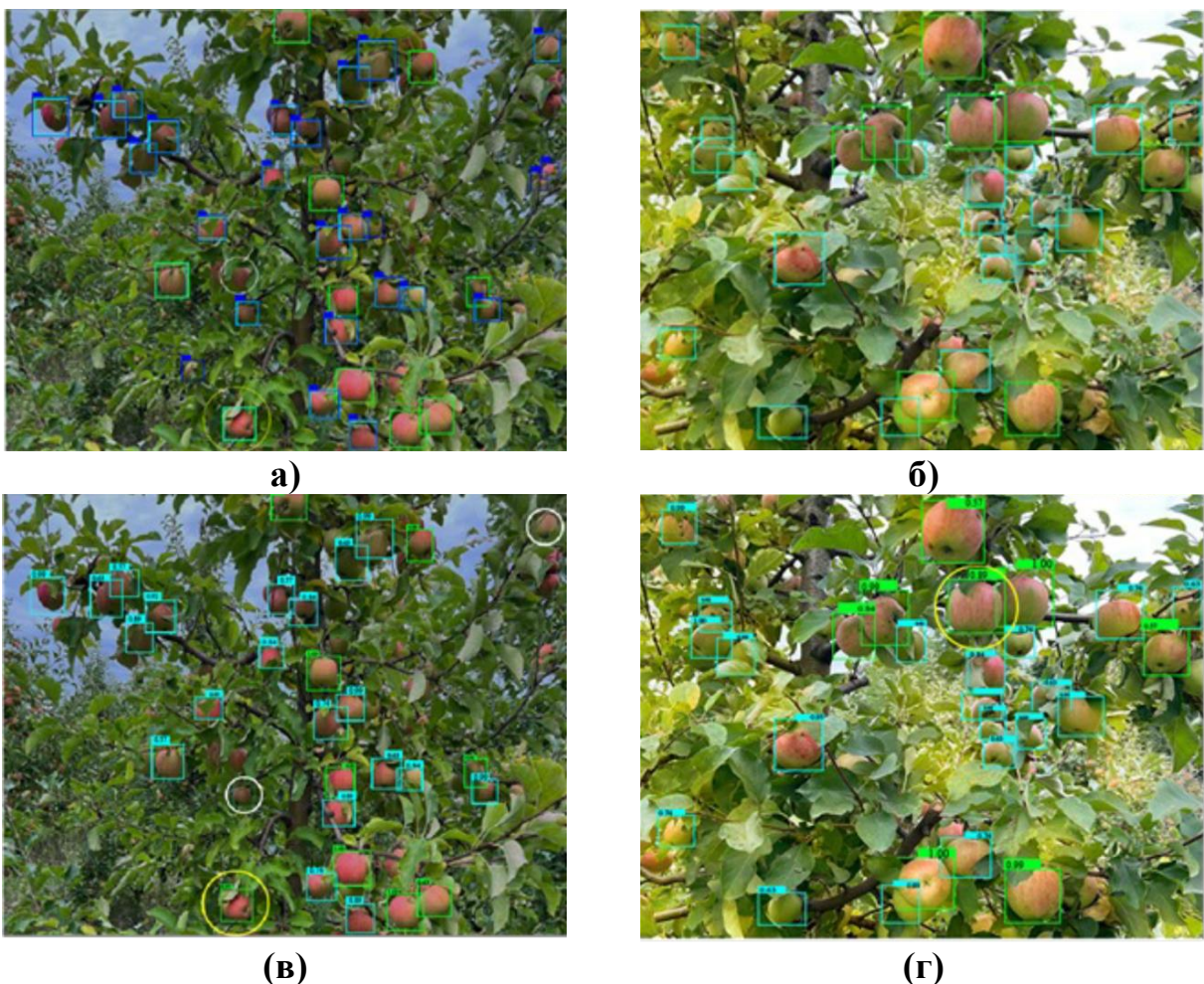


Рисунок 3.15 – Результати виявлення вдосконаленої моделі YOLOv5-v1

Як показано на рис. 3.15а-г), результати виявлення запропонованої вдосконаленої мережі YOLOv5-v1 у дослідженні були точними, без помилкового розпізнавання, або пропущеного розпізнавання.

На рис. 3.16 наведено приклади неправильного розпізнавання фруктового плода за «Технологією 4», «Технологією 3» та «Технологією 2», що позначені жовтим еліпсом на рис. 3.16б), 3.16в) та 3.16г), відповідно.

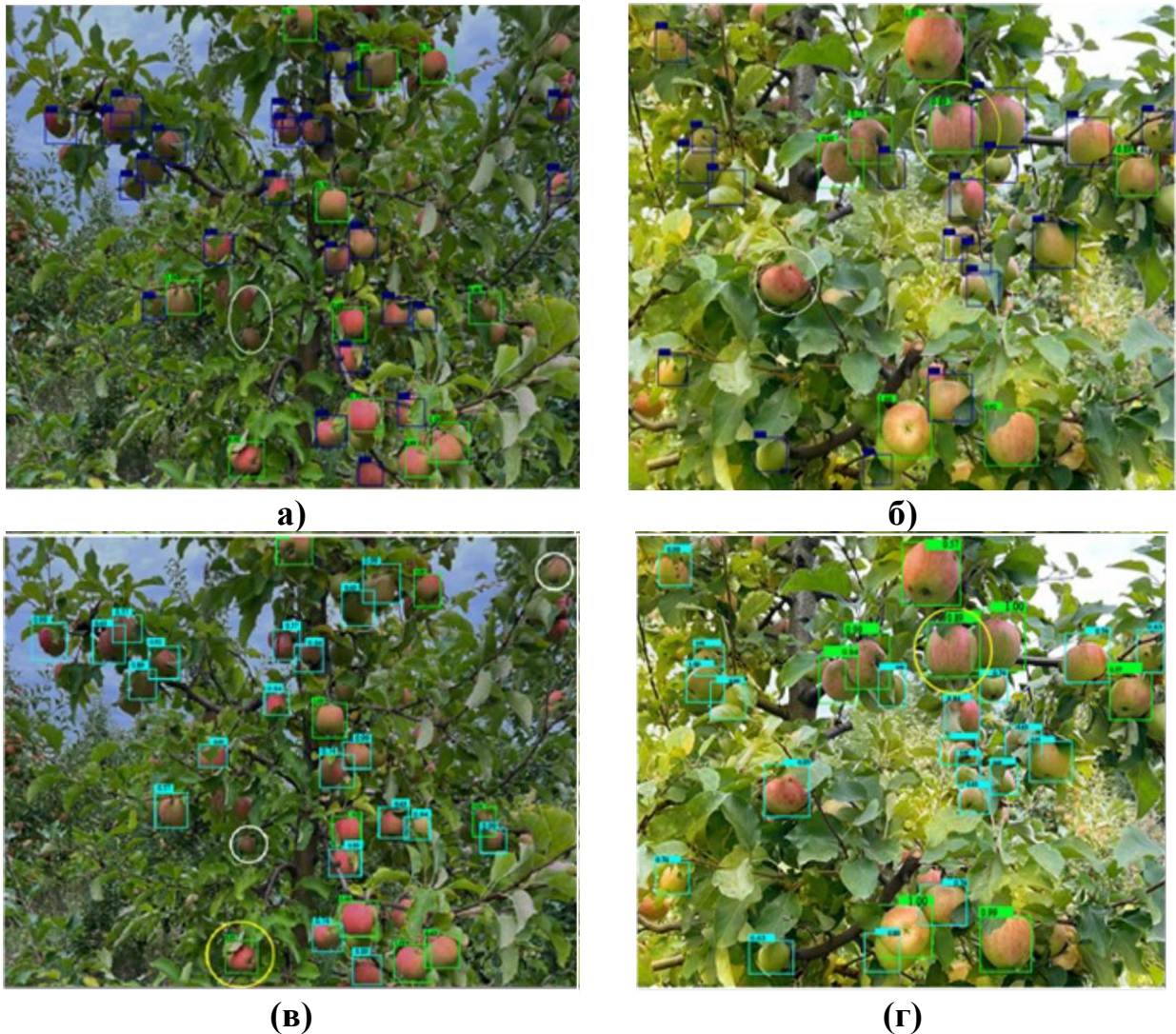


Рисунок 3.16 – Неправильне розпізнавання фруктового плода за допомогою «Технології 4», «Технології 3» та «Технології 2» за похмурих умов

Заразом, «Технології 4», «Технології 3» та «Технології 2» не вдалося розпізнати фруктовий плід, що позначено білим еліпсом на рис. 3.16а), 3.16б) та 3.16в), відповідно. За сонячних умов неправильне розпізнавання того, що



фруктовий плід, який неможливо відстежити відбулось за використання «Технології 4», «Технології 3» та «Технології 2» (позначені жовтим еліпсом на рис. 3.16б), 3.16г) та 3.17а) і неправильне розпізнавання того, що плід, який можна порахувати, було ідентифіковано як неможливе, здійснено «Технологією 4» (позначене рожевий еліпс на рис. 3.17б).



Рисунок 3.17 – Неправильне розпізнавання яблука за допомогою «Технології 1», «Технології 4» та «Технології 2» у похмурих умовах

Отже, запропонована модель YOLOv5-v1 може лише забезпечити високу точність розпізнавання, але також ефективно реалізує властивості легкої нейромережевої моделі. Загалом модель, запропонована в дослідженні, була найлегшою серед п'яти мережевих моделей із найвищим значенням mAP. Це дає можливість для покращити швидкість виявлення саме для випадку коли група БПЛА працює у режимі реального часу.

### 3.5. Висновки до третього розділу

Розроблено метод синхронізації відеопотоків в режимі реального часу, який дає змогу забезпечити накопичення даних про попередні програмні місії БПЛА, у випадку критичних збоїв, які спотворюють цілісність структур даних із джерел отримання відеопотоків. Він передбачає збереження таких структури в

спеціальному журналі помилок та не відправляє їх як вхідні параметри в наступну обробку. Це дало змогу виконувати порівняння отриманих поточних результатів із минулими в режимі реального часу.

Також, було вдосконалено метод виявлення заданих структурних об'єктів на зображеннях. Порівняно з оригінальною архітектурою YOLOv5 в ньому модифіковано модуль фокусування нейронної мережі, видалено згортковий шар поєднуючий вхідну карту ознак з операцією конкатенції. У вдосконаленій спроектованій архітектурі мережі розроблено нові шари для модуля BottleneckCSP-v4, який використовувався для заміни модуля BottleneckCSP в магістральній архітектурі оригінальної мережі YOLOv5.

Також, розроблено спосіб отримання кількості заданих структурних об'єктів із отриманих зображень за допомогою групи БПЛА, який полягає в послідовному виконанні попередньої обробки вхідних даних у вигляді об'єднаних відеокадрів із накладеними обмежувальними рамками виявлених структурних об'єктів, першочерговому порівнянню споріднених координат виявлених об'єктів між двома сусідніми кадрами, присвоєнню унікальних ідентифікаторів нововиявленим об'єктам, обчисленню кількості унікальних структурних об'єктів, обчисленню прогнозних значень координат розташування об'єктів для наступного кадру та обробці вихідних даних, якими є обчислена кількість структурних об'єктів у робочому середовищі.

В роботі проведено порівняння вдосконаленого методу YOLOv5-v1 з 4 іншими технологіями-аналогами розпізнавання на 200 зображеннях тестового набору. В результаті проведених експериментів отримано, що вдосконалений метод YOLOv5-v1 перевершив аналоги, а саме «Технологію 1», «Технологію 2», «Технологію 3» та «Технологію 4», за метрикою mAP на 5,05%, 14,95%, 4,74% та 6,75% відповідно. Отриманий результат відображає покращення точності розпізнавання образів у режимі реального часу.

Основні результати розділу опубліковані у працях [123-126, 131].

## РОЗДІЛ 4.

### СТРУКТУРА, ОЦІНКА ДОСТОВІРНОСТІ ТА ЕФЕКТИВНОСТІ АВТОМАТИЗОВАНОЇ СИСТЕМИ ОБЧИСЛЕННЯ КІЬКОСТІ ЗАДАНИХ СТРУКТУРНИХ ОБ'ЄКТІВ ЗА ДОПОМОГОЮ ГРУПИ БПЛА

У розділі розглянуто опис структури реалізованої автоматизованої системи [140] обчислення кількості заданих структурних об'єктів за допомогою групи БПЛА та оцінювання її достовірності і ефективності. Створена автоматизована система ґрунтується на запропонованих методах виявлення та обчислення фруктових плодів, системі автоматичного планування маршрутів групи БПЛА згідно технології самонавчання, системі самовідновлення та багаторівневій системі управління мережею. Розроблену автоматизовану систему оцінено за критеріями достовірності та ефективності, що включають, результати обчислення кількості структурних об'єктів у робочому середовищі. Проведено експериментальне дослідження автоматизованої системи за показниками ефективності щодо її здатності до інтеграції в робочий процес фермерського господарства з невеликими площами.

4.1. Програмна реалізація автоматизованої системи виявлення та обчислення кількості заданих структурних об'єктів

#### 4.1.1 Опис експериментальної установки

Реалізація автоматизованої системи передбачає її розбиття на відокремлені програмні модулі, що інсталиуються на персональний комп'ютер підприємства. Програмні модулі відповідають за різний функціонал та працюють автономно, проте разом утворюють цілісну розподілену систему, яка здійснює отримання зображень, виявлення та обчислення кількості заданих структурних об'єктів, зокрема яблук на фруктових деревах, за допомогою декількох БПЛА.

Для реалізації всіх програмних модулів в автоматизованій системі було використано мову програмування C#. Вона дає змогу розробляти застосунки на професійному рівні та водночас мати інтеграційний потенціал зі сторонніми модулями, що забезпечує додатковий функціонал до всієї інформаційної системи. Harvested.AI розроблено як кросплатформенне програмне забезпечення на основі технології MAUI. Інтерфейс користувачького інтерфейсу побудовано за декларативною мовою розмітки XAML. Збереження даних, що утворюються під час критичних подій у кожному функційному компоненті, реалізовано в документо-орієнтованій базі даних MongoDB, а збереження даних програмної місії та всіх її системних подій та логів, які утворюються під час роботи, – в PostgreSQL. Програмні модулі виявлення структурних об'єктів та передачі цих даних до програмного інтерфейсу створено в програмних середовищах ML.NET та Alturos.Yolo. Таким чином, навчання нейромережевої моделі, яка є складовим елементом методу виявлення заданих структурних об'єктів, проведено з використанням бібліотеки Keras.NET та хмарних сервісів AWS.

Розроблене програмне забезпечення (ПЗ, далі Harvested.AI) функціонує у вигляді програмних модулів, які за допомогою мережевого інтерфейсу з'єднуються із центральною системою управління за допомогою мережевого інтерфейсу. Сукупно всі автономні програмні модулі утворюють розподілену багаторівневу систему. Окрім цього, мережевий зв'язок групи БПЛА із Harvested.AI встановлюється через впровадження додаткових апаратних приладів, як от антена трансмісії, базова станція, ROVER станція та модуль-приймач. Harvested.AI необхідно інсталиувати на комп'ютерну систему підприємства під управлінням IOS або Windows, після чого з'явиться можливість його запуску. Після інсталиування ПЗ оператор може налаштувати мережеве з'єднання, а саме мережеві порти для комунікації між автономними програмними модулями та БПЛА, які реєструються в системі.

Розгортання системи Harvested.AI ґрунтується на підході настільної програми. Програмні модулі визначаються, як програмні сервіси, які запускаються окремими системними процесами в операційній системі комп'ютера. Кожен

окремий сервіс використовує локальну мережу для передачі даних між іншими компонентами системи; проте за потреби їх можна перемістити в хмарне середовище. Передача даних між програмними модулями здійснюється за допомогою технології gRPC, а кодування даних, які передаються між усіма вузлами, – за proto buffers. Така організація передачі даних дає змогу розширювати інформаційну систему Harvested.AI іншими клієнтськими функціоналами, або новими системними модулями за умови зростання об'єму оброблювальних даних. Оновлення та контроль версій Harvested.AI має відбуватися періодично в процесі її інтеграції в підприємство.

У процесі виконання програмної місії група БПЛА створює відеореєстри, що передаються каналами мережевого зв'язку до бази даних на сервері підприємства. Усі системні події та записи, які обробляються програмними модулями також зберігаються на сервері підприємства. Після завершення програмної місії, бази даних копіюються на хмарний сервер.

Усі компоненти представлені, як автономні програмні модулі. Для збереження реакції та часу відгуку всіх інтерфейсних компонентів на дії оператора, заплановані завдання виконуються в окремому операційному потоці, оскільки потенційно завдання в системі є ресурсоємними.

Компоненти інформаційної системи представлені програмними модулями та за потреби можуть формуватися як апаратно-програмні, причому на апаратний рівень може бути винесено частину функцій, які реалізовано програмно для підвищення рівня стабільності та безпеки цілісності всієї системи через запуск компонентів у різних операційних процесах. Тоді, робота ПЗ може функціонувати в режимах, які дають змогу використовувати будь-яку операційну систему.

Отже, запропонована автоматизована система може бути встановлена у вигляді або настільного ПЗ, або ПЗ, що підтримує інсталювання апаратно-програмних модулів, які функціонують через окремі процеси в операційній системі. Загалом, розроблені засоби демонструють можливість реалізації теоретичних основ та практик створення автоматизованої системи виявлення та обчислення кількості структурних об'єктів.

#### 4.1.2 Засоби налаштування автоматизованої системи

Розглянемо засоби налаштування розробленої автоматизованої системи. Використовуючи головне меню, оператор має можливість переключати інтерфейс ПЗ та виконувати необхідні йому дії. Наступним етапом використання інформаційної системи є реєстрація БПЛА в ній (рис. 4.1).

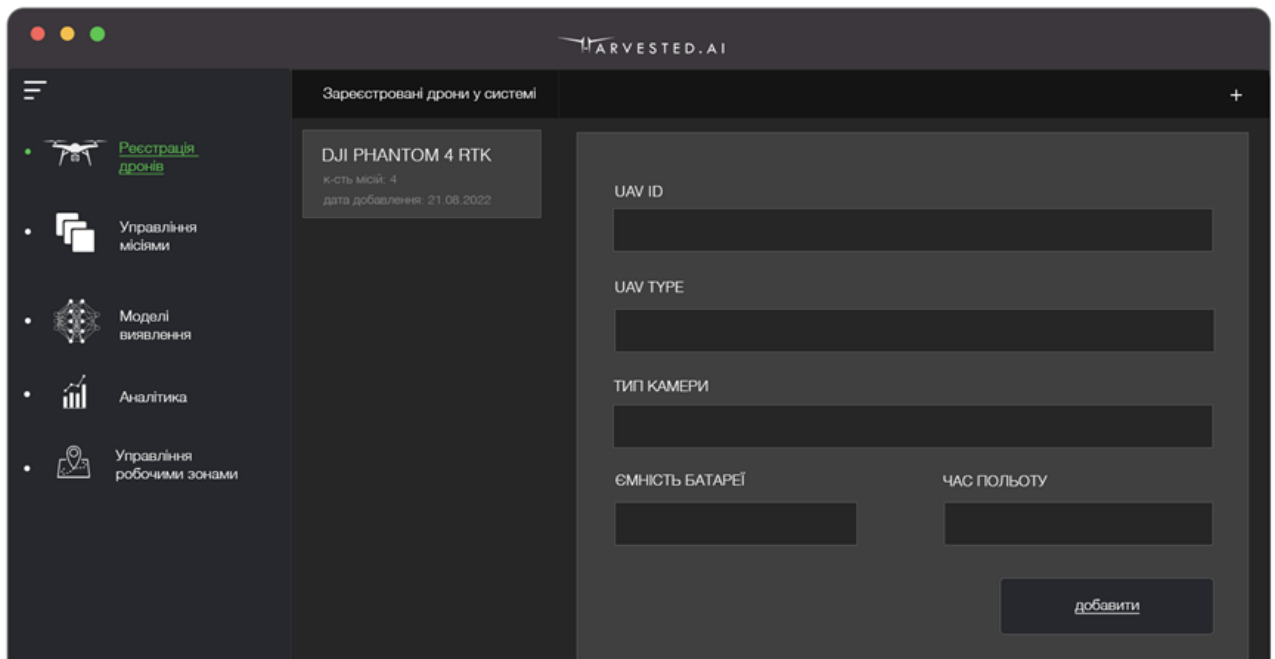


Рисунок 4.1 – Вікно реєстрації БПЛА у системі

Завдяки інтеграційним можливостям Harvested.AI оперативно реагує на введення оператором UAV\_ID, за значенням якої виконує онлайн-перевірку таких характеристик пристрою: 1) можливість самостійного керування, 2) виведення зображення отриманого з відеокамери в режимі реального часу, 3) отримання даних щодо стану заряду батареї та 4) виведення значень параметрів камери.

Наступний крок роботи із системою – утворення координатного маршруту в робочому середовищі для запуску першої програмної місії (рис. 4.2).

У процесі роботи із Harvested.AI оператор може змінювати координатний маршрут або задавати його заново. Інтерфейс утворення координатного маршруту інтегрує підсистему DJI Terra. Створивши початковий маршрут, оператор отримує

можливість вибрати зареєстровані БПЛА та запустити програмну місію із ними. Вікно вибору БПЛА подано на рис. 4.3.

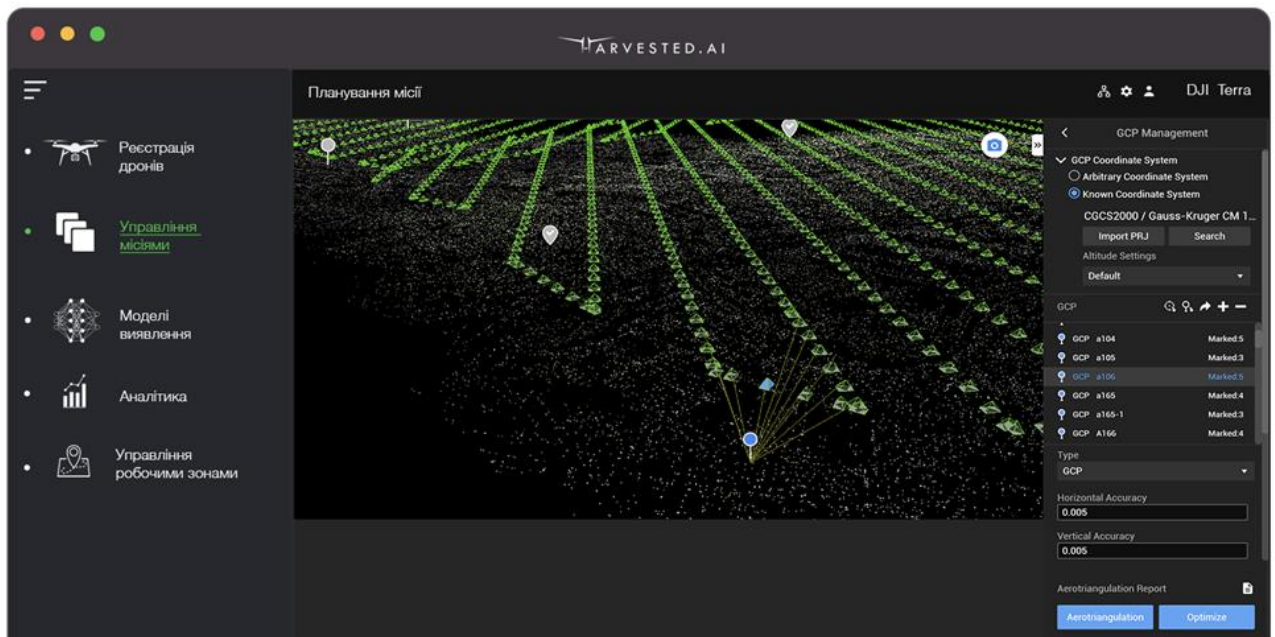


Рисунок 4.2 – Вікно утворення координатного маршруту в робочих зонах

Вікно управління робочими зонами дає змогу оператору вибирати вже наявний маршрут або той, що був утворений у результаті самонавчання. Крім того, оператор має можливість завантажувати новостворені або покращені нейромережеві моделі для виявлення структурних об’єктів у систему через вікно “Моделі виявлення”.

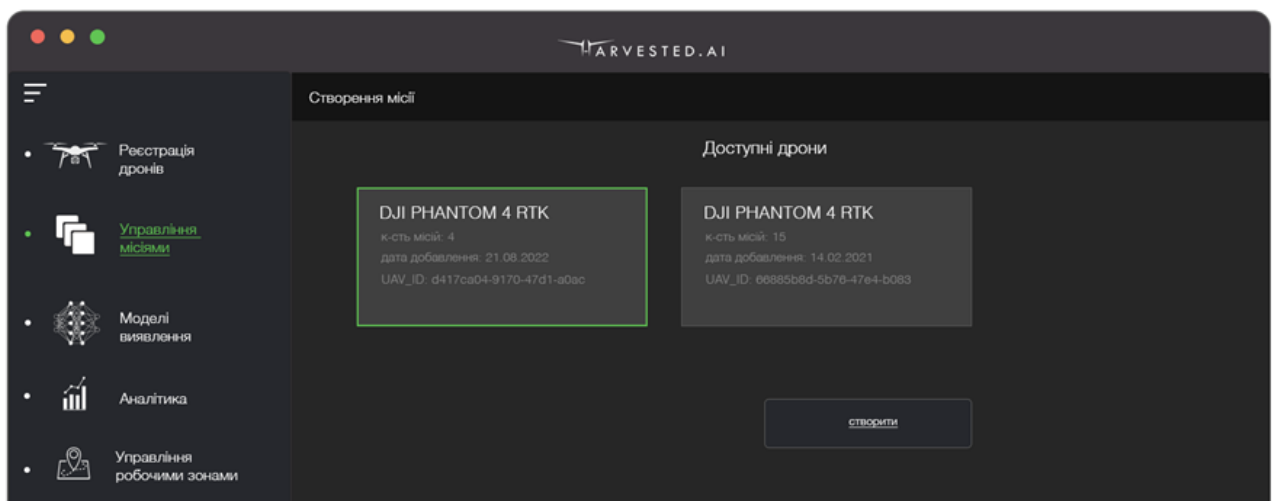


Рисунок 4.3 – Вікно вибору БПЛА для програмної місії

У процесі роботи Harvested.AI створює великі об'єми даних у результаті виконання програмної місії. Тому, для оброблення та аналізу цих даних у системі передбачена можливість створення аналітичних звітів у меню “Аналітика”. Аналітична платформа обробляє результати робочої місії в зручному для користувача вигляді та відфільтровує другорядну інформацію. Ця платформа також дає змогу оператору виявити чи запобігати аварійному завершенню місії заздалегідь. Опрацювання та аналіз даних різних типів необхідне для моделювання розвитку подій та ситуацій у робочому середовищі.

На рис. 4.4 наведено приклад виконання програмної місії групою із трьох БПЛА в робочому середовищі.

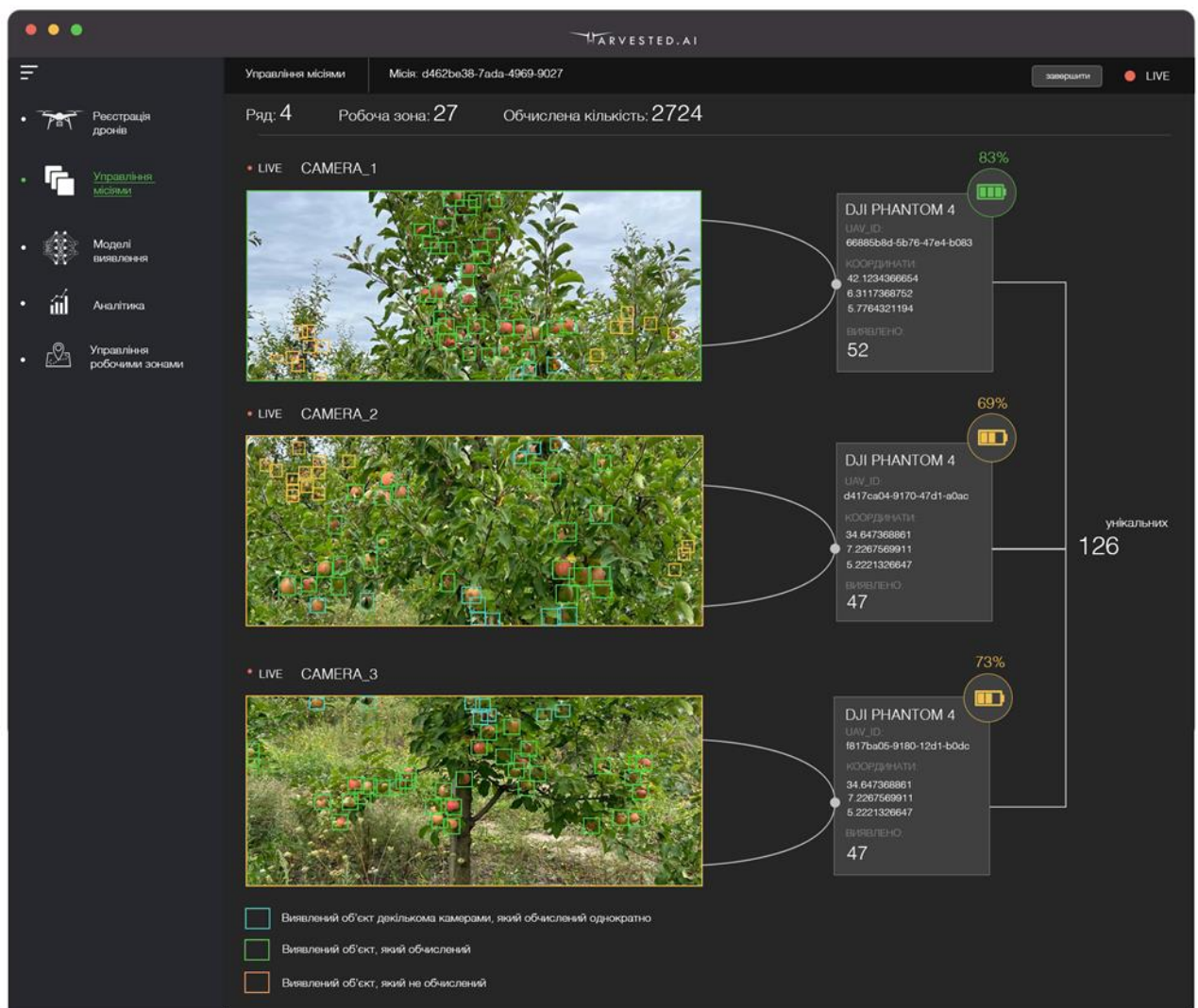


Рисунок 4.4 – Вікно роботи програмної місії у режимі реального часу



Функціональність інтерфейсного рівня розробленого вікна можна поділити на наступні складові:

1. Блок виведення інформації про поточне розташування групи БПЛА та загальна обчислена кількість структурних об'єктів у процесі виконання програмної місії БПЛА.

1.1. Програмний модуль управління маршрутами виводить інформацію про поточне розташування БПЛА, а саме номер фруктового ряду та номер поточної робочої зони. Водночас програмний модуль виявлення та обчислення структурних об'єктів передає значення обчисленої кількості фруктових плодів на інтерфейсний рівень.

2. Виконання програмної місії групи БПЛА в режимі реального часу.

2.1. Оператор має можливість слідкувати за процесом виконання роботи групою БПЛА через відеокамери кожного БПЛА окремо в режимі реального часу. Обчислення кількості структурних об'єктів у режимі реального часу відбувається через динамічне отримання зображень із відеорядів кожного БПЛА в групі. Програмний модуль динамічно обчислює та відображає на інтерфейсі оператора обчислену кількість кожного відеоряду окремо та обчислену загальну кількість у робочій зоні. Поведінкова сигнатура виявлення структурних об'єктів відображена кольоровими прямокутниками: 1) зелена прямокутна рамка вказує на те, що структурний об'єкт був виявлений та обчислений одноразово, 2) бірюзова рамка позначає об'єкти, які були виявлені кількома камерами одночасно, але обчислені одноразово та 3) помаранчева – об'єкти, які не обчислюються в конкретній робочій зоні, оскільки належать іншій робочій зоні.

2.2. Оператор має змогу виконати аварійне завершення місії, натиснувши на кнопку “завершити”, у результаті чого група БПЛА здійснює політ у точку завершення.

3. Характерні значення БПЛА під час виконання програмної місії.

3.1. Під час виконання програмної місії в робочій зоні, група БПЛА рухається по візуально змодельованому тривимірному мапою місцевості зі своїми програмними координатами. Поточні програмні координати руху БПЛА слугують

вхідними даними в програмному модулі планування маршрутів для автоматичного визначення координат (див. п. 2.2.1), що забезпечує подальше переміщення групи БПЛА. Поточне розташування кожного БПЛА з унікальним ідентифікатором у групі відображається в інтерфейсі Harvested.AI (див. рис. 4.4). Як наслідок, у разі зміни погодних умов у робочому середовищі та втрати зв'язку з апаратним пристроєм оператор має змогу його знайти та повернути в групу.

3.2. Час виконання програмної місії залежить від ресурсу батареї кожного БПЛА. Якщо заряд батареї хоча б одного БПЛА опускається нижче визначеного наперед мінімального значення (10%), то група БПЛА направляється до початкової або кінцевої точок робочого середовища для підзарядки. Точка направлення обирається автоматично, залежно від дистанції польоту до кожної із них (вибирається найменша). Програмний модуль управління БПЛА надсилає значення місткості батареї кожного БПЛА в групі на інтерфейсний рівень з інтервалом 30 секунд. Група БПЛА може продовжити виконання робочої місії та завершити її автоматично; такий сценарій роботи залежить від кількості обраних оператором робочих зон під час утворення програмної місії.

У підсумку, запропонована автоматизована система Harvested.AI надає можливість оператору створювати автоматизовані робочі місії з мінімальним ручним втручанням. Harvested.AI надає такі функціональні можливості: 1) введення інформації про апаратний пристрій у систему для його верифікації в розробника БПЛА, 2) утворення візуально змодельованої тривимірної мапи місцевості з програмними координатами для переміщення групи БПЛА в робочих зонах, 3) створення програмної місії з вибором БПЛА, які формуються в робочу групу, 4) утворення відеорядів із камер групи БПЛА та обчислення кількості структурних об'єктів на фруктових деревах та 5) оброблення та збереження даних, що отримані в результаті виконання програмної місії групою БПЛА.

За розробленим програмним забезпеченням Harvested.AI опубліковано 1 свідоцтво про реєстрацію авторського права на твір (програму) [140]. Лістинг програмного коду Harvested.AI подано у додатку Д.

## 4.2. Постановка і проведення експериментальних досліджень застосування розробленої автоматизованої системи

Експерименти із розробленою системою Harvested.AI проведено в робочому середовищі яблуневого саду. Усі результати експериментів збережено у відповідні бази даних. Основною метою проведення експериментальних досліджень щодо обчислення кількості яблук була перевірка ефективності та достовірності роботи автоматизованої системи, як сукупності описаних методів у дисертаційній роботі, проти ручного обчислення, виконаного людиною. Людський ресурс у процесі проведення експериментів використано для порівняння із результатами автоматизованої системи.

### 4.2.1 Опис експериментального середовища

Нижче подано загальний опис процесу проведення експериментів автоматизованої системи.

1. Визначення робочої області для проведення експериментів. Тривалість робочої місії з використання групи БПЛА є обмеженим унаслідок ресурсу заряду батареї та погодних умов довкілля. Для виконання експериментів було використано робоче середовище яблуневого саду Фермерського господарства “САМ+”. Господарство розташоване в Городоцькій міській територіальній громаді Городоцького району Хмельницької області в селі Кузьмин. Тип яблунь – струнке веретено. Загальна площа господарства становить 50 га. Довжина рядів саду становить 500-800 м, дерева посаджені через кожних 2 метри, ширина міжряддя – 4 м, середня висота дерев – 3,5 м, підщепа – 106 мм. За оцінюванням фермерського господарства, кожне яблуневе дерево родить приблизно 200 яблук за сезон.

Експериментальні дослідження у визначеній робочій області проведено за помірних погодних умов, а саме за слабких поривів вітру та дощу, під час сонячної та похмурної погоди. Нижче наведено детальний опис умов експериментів:

а) Експеримент у фруктовому саду за сонячної погоди виконувався 26 вересня о 11:00-12:00, коли сонце перебувало в зеніті. Приклад робочого середовища за сонячної погоди наведено на рис. 4.5а).



а)



б)



в)

Рисунок 4.5 – Експериментальне робоче середовище, сфотографоване за різних погодних умовах

б) Експеримент у фруктовому саду за похмурої погоди виконувався 28 вересня о 13:00-14:00, коли сонце було сховане за хмарами. Приклад робочого середовища за хмарної погоди наведено на рис. 4.5б).

в) Ще один експеримент було проведено 29 вересня за сонячної погоди у вечірню пору дня (19:00-20:00), коли сонце перебувало близько до обрію, породжуючи високу затіненість плодів у яблуневому саду. Приклад робочого середовища за високої затіненості наведено на рис. 4.5в).

Ефективність експериментального дослідження робочої місії залежить від того чи робоча область уже використовувалася в експериментах, чи буде використовуватись уперше, оскільки в разі повторного використання робочої області, або якоїсь її частини, координація групи БПЛА ґрунтуватиметься на автоматично скорегованому маршруті, що може бути ефективнішим, ніж у випадку використання поточного маршруту першочергово. Тому, для дослідження було взято 2 фруктових ряди з робочого середовища, що становило ~1600 м шляху для обльоту БПЛА, приблизна кількість дерев становила 800. Описаний експериментальний маршрут використовувався багаторазово.

2. Встановлення необхідного для експериментів комплекту обладнання в робочому середовищі. Етап визначення технічних характеристик кожного з апаратних приладів залежить від оцінки ефективності виконання робочої місії надалі. Тому, для проведення експериментів обрано обладнання з такими технічними характеристиками:

а) Базова станція. Було обрано легку та переносну базову станцію D-RTK 2 High Precision GNSS Mobile Station, оскільки робочою місією визначення переміщення апаратних прикладів між робочими сегментами саду з наступними технічними характеристиками:

- а.1) Частота передачі даних становить від 1 Гц до 10 Гц.
- а.2) Робоча частота працює на частотах L1, L2, L5 та L-band.
- а.3) Точність горизонтального позиціювання становить менше ніж 2.5 см, а вертикального менше ніж 5 см.

б) RTK модулі. Основою позиціювання групи БПЛА в тривимірному просторі є комунікація між апаратними приладами, дронами та супутниками. Тому, для виконання робочих місій у режимі реального часу та для забезпечення високої точності руху БПЛА, були обрані такі технічні компоненти:

б.1) Для прийняття сигналів зі супутника навігаційних систем (GPS) було обрано та встановлено в робоче середовище багат шарові GNSS – антени із широким кутом приймання сигналів. Поточний апаратний компонент взаємодіє із RTK модулем, який встановлено в БПЛА та виконує функцію корекції та стабілізації положення в тривимірному просторі.

б.2) Взаємодія апаратних RTK компонентів з автоматизованою системою здійснена за допомогою радіо модемів. Це дало змогу отримати швидкий мережевий зв'язок між усіма залежними компонентами експерименту.

в) Живлення. Три БПЛА було підключено до зовнішніх акумуляторів, що забезпечило їхнє використання до 1 год.

3. Формування групи БПЛА для виконання робочої місії. З огляду на технічні характеристики описаних вище апаратних пристроїв, було вирішено використовувати модель БПЛА із модулем RTK. Тому, у поточних експериментах було використано 3 БПЛА DJI Phantom 4 RTK з такими технічними та користувацькими характеристиками: 1) висока роздільна здатність камери з матрицею 1 дюйм та 20-мегапіксельний датчик із можливістю знімання відео до 4К за 60 кадрів у секунду, 2) вбудована система передбачення перешкод, 3) час автономної роботи – до 30 хв та 4) висока стійкість до погодних умов та несподіваних поривів вітру.

#### 4.2.2 Оцінювання ефективності автоматизованої системи

Оцінювання ефективності розробленої автоматизованої системи проведено з одним та кількома програмними модулями системи. Таке порівняння дало змогу оцінити ефективність розроблених методів, що подані компонентами окремо та всієї системи загалом. Оцінку ефективності визначено на основі порівняння обчисленої кількості плодів групою БПЛА та ручного обрахування в робочому середовищі. Нижче наведено результати експериментів за сонячних та похмурих погодних умов.

Однією з основних оцінок усієї системи є ефективність метода планування маршрутів. Оцінку ефективності прокладених автоматичних маршрутів автоматизованою системою для групи БПЛА визначено як відношення вагової суми значень показників роботи БПЛА під час одного польоту до максимального часу роботи БПЛА, що визначається виробником. Оцінка ефективності прокладених автоматичних маршрутів визначатимемо за формулою

$$E = \frac{w_1 \times T + w_2 \times En + w_3 \times P + w_4 \times C}{Tk}, \quad (4.1)$$

де  $E$  – ефективність прокладеного робочого маршруту БПЛА,  $T$  – час, витрачений на виконання робочої місії,  $En$  – екологічність,  $P$  – час затримки при збоях під час виконання робочої місії,  $C$  – кількість технічних/механічних помилок,  $Tk$  – максимальний час роботи БПЛА, що визначається виробником, а  $w_i, i = 1, \dots, 4$  – вагові коефіцієнти відображення важливості кожної метрики.

У процесі проведення тестових експериментів значення вагових коефіцієнтів для оцінки (4.1) визначено так:  $w_1 = 0,25$ ,  $w_2 = 0,42$ ,  $w_3 = 0,08$ ,  $w_4 = 0,25$ . Результати експериментальних досліджень автоматизованої системи в розрізі методу планування маршрутів наведено в табл. 4.1.

Відповідно до табл. 4.1, найкращі технічні результати було отримано групою БПЛА саме за похмурої погоди. За першого виконання робочої місії групою БПЛА, де робота виконувалася за сонячних (сонце в зеніті), умов сталося 3 технічних збоїв у першого БПЛА та 2 – у третього. Технічні збої переводили автоматизовану систему в стан “очікування”; як наслідок, час затримки виконання місії та значення екологічності кожного дрона збільшився також зростали. Під час виконання другої експериментальної робочої місії за умов високої затіненості всі три БПЛА мали критичні помилки на деяких ділянках робочих сегментів, що також збільшило час затримки та екологічність. Заразом, виконуючи експериментальну робочому місію за похмурих погодних умов, збоїв у групи БПЛА не виявлено.

Таблиця 4.1

Результати експериментів автоматизованої системи та методу планування маршрутів

№	Апаратні пристрої	Час виконання робочої місії, хв	Час затримки, хв	Кількість технічних помилок / механічних помилок	Екологічність, Вт	Загальна ефективність, $E$
Сонячно (сонце у зеніті)						
1	БПЛА 1	28,40	0,90	3	179,10	1,7485
2	БПЛА 2	27,18	0,13	0	168,00	1,7124
3	БПЛА 3	28,11	0,40	2	175,00	1,7201
Сонячно (висока затіненість)						
1	БПЛА 1	28,17	0,40	2	178,10	1,7298
2	БПЛА 2	27,60	0,13	1	178,00	1,7178
3	БПЛА 3	28,11	0,37	2	177,00	1,7197
Похмуро						
1	БПЛА 1	27,18	0,10	0	168,10	1,6314
2	БПЛА 2	27,18	0,13	0	168,11	1,6319
3	БПЛА 3	27,17	0,12	0	168,09	1,6393

На рис. 4.6 подано порівняння оцінок ефективності автоматично побудованих маршрутів для кожного БПЛА залежно від погодних умов. З рис. 4.6 бачимо, що попри вплив зовнішніх чинників таких як погана освітленість за похмурої погоди та висока затіненість плодів у вечірню погоду, автоматизована система формує маршрути, оцінки ефективності для яких набули близьких значень. Так, найкраще спрацювали БПЛА за похмурої погоди (1.6314, 1.6319, 1.6393), що можна пояснити сприятливими погодними умовами для виконання польотів. Показники ефективності за сонячної погоди (1.7485, 1.7124, 1.7201) та затіненості (1.7298, 1.7178, 1.7197) виявилися майже однаковими.

Отже, результати дослідження показують, що попри незначний вплив помірного вітру на значення певних метрик, розглядувані погодні умови, за яких



проводились експерименти, загалом не впливають на значення ефективності утворення автоматичних маршрутів.

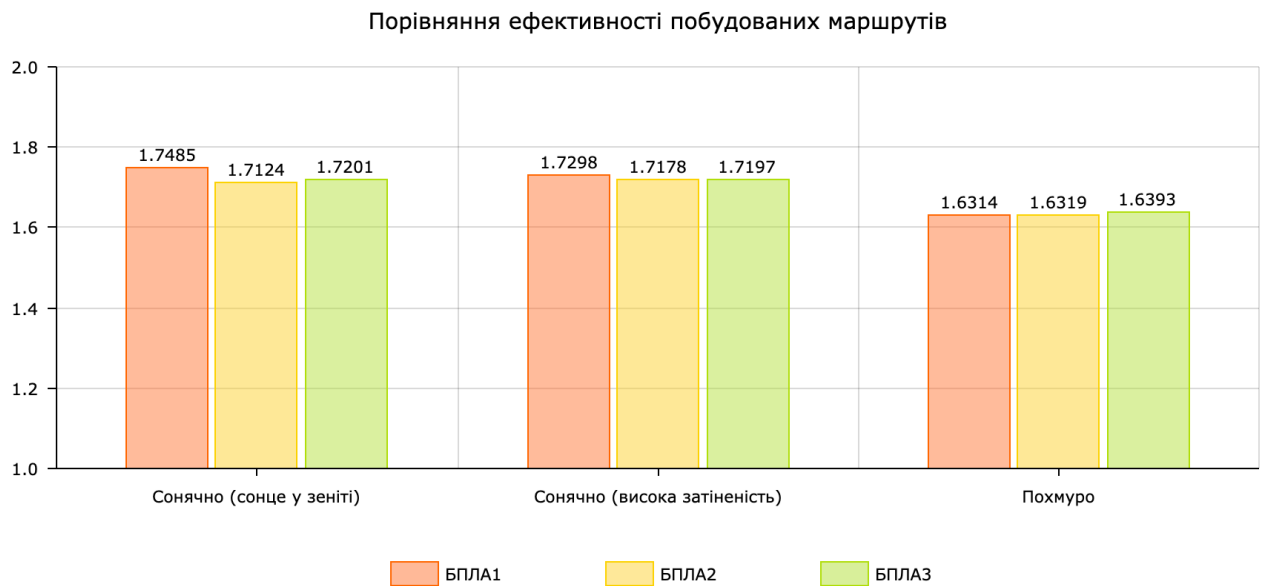


Рисунок 4.6 – Порівняння ефективності побудованих маршрутів автоматизованої системою

Далі розглянемо результати проведених експериментів щодо виявлення та обчислення кількості фруктових плодів. Відповідно до оціночної інформації Фермерського господарства “САМ+”, кількість плодів, що росло на деревах фруктового саду на момент проведення експериментів, становила ~160 тисяч штук. Результати виявлення та обчислення кількості фруктових плодів наведено в табл. 4.2.

У результаті проведених експериментів було правильно виявлено та обчислено 147 382, 145 223 та 132 304 фруктових плодів за трьох різних погодних умов (див. табл. 4.2). Водночас встановлено, що певна кількість фруктових плодів була виявлена, але не обчислена, а саме 72 143, 69 944 та 69 336 для трьох погодних умов; ці об’єкти були ідентифіковані детектором автоматизованої системи як фруктові плоди, проте не враховані під час обрахування, оскільки вони перебували поза поточними робочими зонами.

Таблиця 4.2

Результати експериментів автоматизованої системи щодо виявлення та  
обчислення кількості фруктових плодів

№	Погодні умови	Виявлено та обчислено шт.	Виявлено, але не обчислено шт.	Виявлено багатьма камерами, але обчислено одноразово шт.	Всього шт.
1	Сонце (сонце в zenіті)	147 382	72 143	5 743	~160 000
2	Похмуро	145 223	69 944	6 611	~160 000
3	Сонце (сильна затіненість)	132 304	69 336	8 442	~160 000

В табл. 4.2 бачимо, що певну кількість плодів, а саме 5 743, 6 611 та 8 442 за трьох погодних умов, було одночасно виявлено камерами двох або трьох БПЛА та обчислено одноразово, відповідно до логіки запропонованого методу обчислення структурних об'єктів. Водночас певна кількість фруктових плодів була прикрита листям або перебувала в зонах, погано помітних як для камери БПЛА так і для людського ока. Як наслідок, детектор автоматизованої системи допустив незначну кількість помилок.

Значення статистичних показників, що оцінюють виявлення на обчислення кількості плодів, подано в табл. 4.3.

З табл. 4.3 бачимо, що найвищі значення метрик було досягнуто за сонячної погоди, коли сонце перебувало в zenіті, а найнижчі – за сильної затіненості.

Порівняння значень показників достовірності розпізнавання за різних погодних умов зображено на рис. 4.7.

Таблиця 4.3

## Результати роботи детектора

№	Погодні умови	Precision	Recall	F <sub>1</sub>	Помилка I роду	Помилка II роду
1	Сонце (сонце в zenіті)	0,92	0,89	0,90	0,08	0,11
2	Похмуро	0,86	0,82	0,84	0,14	0,18
3	Сонце (сильна затіненість)	0,78	0,74	0,72	0,22	0,26

Результати проведених експериментальних досліджень показують, що рівень достовірності (Ассурасу) автоматизованої системи складає: 1) у сонячну погоду – 92,11%, 2) у похмуро – 90,76% та 3) у сонячну погоду, але з високою затіненістю – 82,69%. Отримані значення свідчать про високий показник ефективності автоматизованої системи для виявлення та обчислення фруктових плодів у реальних умовах.

Оцінювання достовірності розпізнаних плодів групою бпла.

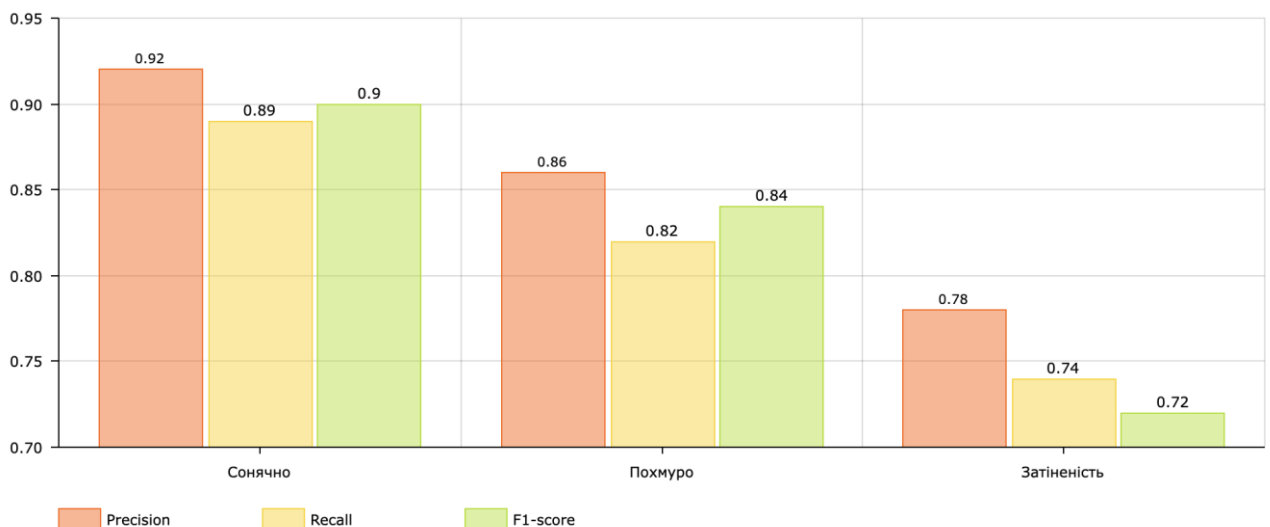


Рисунок 4.7 – Порівняння достовірності розпізнавання за різних погодних умов

Порівняння оцінок рівня помилок, що були допущені групою БПЛА під час проведення експериментів за різних погодних умов, зображено на рис. 4.8.

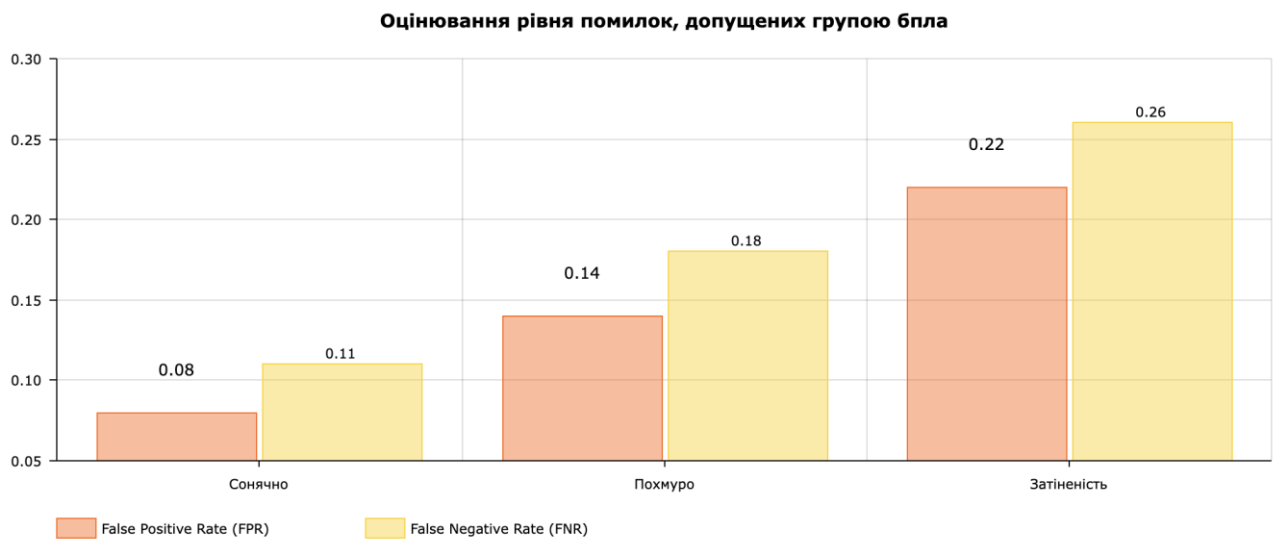


Рисунок 4.8 – Порівняльний аналіз оцінювання рівня помилок, допущених групою БПЛА

З рис. 4.8 можна встановити, що кількість допущених помилок і, як наслідок, якість розпізнавання плодів у реальних умовах суттєво залежить від погодних чинників. Це може бути пов'язано з тим, що певні ділянки цільових робочих зон можуть покриватися тінню інших дерев. Такий чинник у поєднанні з іншими візуальними шумами (листя, гілки дерев тощо) може суттєво ускладнювати ідентифікацію цільових об'єктів та їхнє відстеження в динаміці режиму реального часу через обмежений кут охоплення камер БПЛА.

З аналізу результатів проведених експериментальних досліджень із розробленою автоматизованою системою підтверджено, що використання групи БПЛА як компоненти автоматизованої системи для виконання обльоту у фруктовому саду за сонячних та похмурих погодних умов є ефективним. Зокрема, значення помилок I та II роду за умов сонця в zenіті становили 8 % та 11 %, відповідно, за хмарної погоди – 14 % та 18 %, відповідно, а за умови погоди з високою затіненістю – 22 % та 26 %, відповідно. Такі результати свідчать про те, що за сонячних та похмурих погодних умов із низьким рівнем тіні автоматизована

система демонструє високу точність та надійність у роботі. Водночас варто зауважити, що через наявність у фруктовому саду значних візуальних шумів, як от, прикриття плодів листям та гілками, ефективність роботи групи БПЛА зокрема та автоматизованої системи загалом у реальних умовах не може бути виконана на 100 %, що може слугувати перспективним завданням для подальших досліджень.

За оцінку ефективності розробленого методу синхронізації відеопотоків в режимі реального часу взято індекс SSIM (1.13) та індекс PSNR (1.14). Досягнення ефективності синхронізації відеопотоків за індексом SSIM полягає в отриманні значення від 0,5 до 1, що вважається високим значення ефективності; водночас значення SSIM у діапазоні від 0 до 0,49 свідчить про неефективну синхронізацію. Індекс PSNR демонструє ступінь якості зображення, отриманого в результаті операції об'єднання; значення індексу PSNR обраховується як співвідношенням між максимально можливою потужністю візуального сигналу та шуму, який присутній у зображенні; чим вище значення, тим краща якість отриманого зображення.

У табл. 4.4 наведено результати виконання модуля синхронізації відеопотоків автоматизованої системи, що реалізує відповідний метод, для потоку із 12 послідовних груп відеокадрів, який випадковим чином відібраний для тестування; кожна група містить три відеокадри, що отримані від трьох БПЛА, які надалі об'єднуються в одне зображення.

Відповідно до табл. 4.4, ефективність за індексом SSIM коливається від 0,79 до 0,92, із середнім значенням 0,87. Водночас зображення, для яких значення індексу SSIM менше, ніж 0,50, вважаються системою спотвореними. Заразом ті об'єднані зображення, для яких значення індексу PSNR перевищує 30, вважаються високої якості; водночас значення PSNR менше, ніж 30 свідчать про низьку якість зображення, що може бути зумовлено зовнішніми чинниками робочого середовища (сильні пориви вітру, опади тощо).

Якщо для поточного зображення значення індексу SSIM менше, ніж 0,50, а також значення індексу PSNR менше, ніж 30, то таке зображення вважатиметься системою спотвореним, а тому відкидатиметься та не потраплятиме до наступного модуля виявлення та обчислення кількості заданих структурних об'єктів.

Таблиця 4.4

Результати дослідження ефективності методу синхронізації відеопотоків,  
отриманих групою БПЛА

Об'єднане зображення	SSIM	PSNR
1	0,90	35,20
2	0,45	27,22
3	0,72	37,50
4	0,85	29,11
5	0,87	36,90
6	0,91	39,10
7	0,86	31,43
8	0,47	27,87
9	0,90	30,12
10	0,83	31,54
11	0,52	28,91
12	0,86	30,36

У підсумку, проведено експериментальне дослідження розробленої автоматизованої системи виявлення та обчислення кількості фруктових плодів у реальних умовах. Для оцінювання ефективності автоматизованої системи використано такі показники: а) індекс Е ефективності визначення автоматичних маршрутів для групи БПЛА, б) показники достовірності Ассурасу виявлення фруктових плодів та помилки I та II роду та в) показники ефективності синхронізації відеокадрів у режимі реального часу SSIM та PSNR. У результаті проведених експериментів було доведено ефективність роботи розробленої автоматизованої системи, що підтверджується високим середнім значенням у 82,69% показника достовірності виявлення та обчислення кількості фруктових плодів.

#### 4.3. Висновки до четвертого розділу

В результаті створена автоматизована система з групою БПЛА, яка ґрунтується на алгоритмах комп'ютерного зору, машинного навчання та

оптимізації. Були розроблені програмні компоненти, які обробляють та аналізують відеокадри та дозволяють ідентифікувати й обчислювати фруктові плоди на деревах із високою точністю в режимі реального часу. Управління та координації групи БПЛА забезпечують ефективне спільне виконання робочої місії з мінімізацією ризиків колізій між літальними апаратами.

Розроблена автоматизована система призначена для виявлення та обчислення кількості яблук у фруктовому саду в режимі реального часу. Перевагою розробленої системи над аналогами є отримання нею множини відеокадрів у режимі реального часу з камер кількох БПЛА та синхронізація цих відеокадрів між собою в одну інформаційну структуру даних, що надалі трансформується в суцільне зображення. Крім того, використання функцій оптимізації якості зображення дає змогу максимально ефективно виявляти задані структурні об'єкти під час виконання робочих місій БПЛА в робочому середовищі. Використання такого засобу трансформації дало змогу системі отримувати суцільний потік даних до всіх наступних програмних компонентів автоматизованої системи. Оцінка синхронізації відеопотоків за індексом SSIM коливається від 0,79 до 0,92, із середнім значенням 0,87, а за індексом PSNR – від 22 до 39, що свідчить про високу ефективність роботи розробленої системи із відеопотоками та хорошу якість отриманих об'єднаних зображень.

У результаті проведених експериментальних досліджень було доведено ефективність роботи розробленої автоматизованої системи, що підтверджується високим середнім значенням у 82,69% показника достовірності виявлення та обчислення кількості фруктових плодів та низьким середнім рівнем помилок I (14,67%) та II (18,33%) роду. Результати експериментів із розробленою автоматизованою системою показують, що її ефективність управління групою БПЛА для обчислення кількості структурних об'єктів у фруктовому саду.

Основні результати розділу опубліковані у працях [122, 127, 131, 140].

## ВИСНОВКИ

У результаті виконання дисертаційного дослідження було розв'язано актуальну науково-прикладну задачу покращення ефективності збору, точності розпізнавання та обробки зображень, отриманих із використанням БПЛА, для виявлення заданих об'єктів, а також розроблено відповідні засоби.

У роботі отримано наступні наукові та практичні результати:

1. Здійснено аналіз відомих засобів для збору зображень з використанням БПЛА, існуючих систем збору та розпізнавання зображень заданих об'єктів з використанням БПЛА, встановлено їх переваги та виділено недоліки, зокрема в частині точності отриманих результатів, оскільки збір зображень відбувається в реальних умовах природнього середовища, що впливає на позиціонування засобів збору таких зображень. В результаті, було встановлено, що для збору таких зображень доцільно використовувати групи БПЛА, які потрібно синхронізувати між собою та включити до єдиної системи збору для автоматизації процесу збору зображень об'єктів. Також, визначено оптимальні методи для використання при обробці зображень об'єктів, визначено їх недоліки та запропоновано стратегії їх удосконалення з врахуванням особливостей отримання зображень об'єктів в реальному часі з рухомих об'єктів, враховуючи повторення фрагментів зображень, які отримуються з різних камер.

2. Розроблено новий метод побудови маршрутів БПЛА згідно технології самонавчання, суть якого в позиціюванні апаратних пристроїв у тривимірному координатному просторі в реальному часі за рахунок самонавчання при побудові маршрутів їх руху, що дає змогу покращити переміщення та синхронізацію між групою БПЛА або одного БПЛА в межах робочого сегменту і за рахунок цього збільшення кількості опрацьованих даних.

3. Розроблено новий метод динамічного отримання зображень заданих структурних об'єктів в тривимірному просторі за допомогою декількох БПЛА, суть якого в тому, що забезпечено в процесі активації кожного БПЛА генерацію відеоряду в центральну систему, прийняття рішень про подальшу роботу групи



БПЛА, підтвердження цілісності програмної місії та визначення рівня критичності для продовження виконання роботи групи БПЛА за рахунок визначення станів модулів та комплексному врахуванні вимог розподілення, багаторівневості та автоматизованості, що покращує узгодження між різними БПЛА та досягнення переміщення всієї групи БПЛА від заданих початкових до кінцевих точок програмної місії автономно.

4. Розроблено новий метод синхронізації відеопотоків в режимі реального часу. Він дає змогу забезпечити накопичення даних про попередні програмні місії БПЛА, у випадку критичних збоїв, які спотворюють цілісність структур даних із джерел отримання відеопотоків, зберігає такі структури в спеціальному журналі помилок та не відправляє їх як вхідні параметри в наступну обробку. Це дає змогу виконувати порівняння отриманих поточних результатів із минулими в режимі реального часу і, таким чином, забезпечує оперативне отримання результатів та здійснення виявлення структурних об'єктів, які були пропущені в процесі минулих програмних місій.

5. Вдосконалено метод виявлення заданих структурних об'єктів на зображеннях. На відміну від оригінальної архітектури YOLOv5, він полягає в тому, що модифіковано модуль фокусування нейронної мережі, видалено згортковий шар поєднуючий вхідну карту ознак з операцією конкатенції, актуалізовано механізм візуальної уваги для вилучення ознак, об'єднання шарів 4 і 15, 6 і 11, 10 і 21 оригінальної архітектури замінено на об'єднання шарів 5 і 18, 8 і 14, 13 і 24 вдосконаленої архітектури, вихідні карти ознак 14-го та 21-го шарів вдосконаленої архітектури об'єднано між собою, що дало змогу покращити точність виявлення і зменшити час навчання нейронної мережі.

6. Розроблено новий спосіб отримання кількості заданих структурних об'єктів із отриманих зображень за допомогою групи БПЛА. Він полягає в послідовному виконанні попередньої обробки вхідних даних у вигляді об'єднаних відеокадрів із накладеними обмежувальними рамками виявлених структурних об'єктів, першочерговому порівнянні споріднених координат виявлених об'єктів між двома сусідніми кадрами, присвоєнні унікальних ідентифікаторів

нововиявленим об'єктам, обчисленні кількості унікальних структурних об'єктів, обчисленні прогнозних значень координат розташування об'єктів для наступного кадру та обробці вихідних даних, якими є обчислена кількість структурних об'єктів у робочому середовищі.

7. Створена автоматизована система з групою БПЛА ґрунтується на алгоритмах комп'ютерного зору, машинного навчання та оптимізації. У рамках експериментального дослідження були розроблені програмні компоненти, які обробляють та аналізують відеокадри та дозволяють ідентифікувати й обчислювати фруктові плоди на деревах із високою точністю в режимі реального часу. Управління та координації групи БПЛА забезпечують ефективне спільне виконання робочої місії з мінімізацією ризиків колізій між літальними апаратами.

У результаті проведених експериментальних досліджень було доведено ефективність роботи розробленої автоматизованої системи, що підтверджується високим середнім значенням у 82,69% показника достовірності виявлення та обчислення кількості фруктових плодів та низьким середнім рівнем помилок I (14,67%) та II (18,33%) роду.

Теоретичні та практичні результати дослідження впроваджені в ТОВ «ЮКС++» (м. Хмельницький), Державному підприємстві «Новатор» (м. Хмельницький), ПП «НОЛТ ТЕХНОЛОДЖИС» (м. Хмельницький), ТОВ «Агротех сервіс», а також, в освітньому процесі Хмельницького національного університету

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Aggression against Ukraine : resolution / adopted by the General Assembly UN. *General Assembly (11th emergency special sess. : 2022)*. 18.03.2022. URL: <https://digitallibrary.un.org/record/3965290> (дата звернення: 22.09.2022).
2. Ben Hassen T., El Bilali H. Impacts of the Russia-Ukraine war on global food security: Towards more sustainable and resilient food systems? *Foods*. 2022. Vol. 11, No. 15. P. 2301.
3. Sustainable agrifood systems for a post-growth world / S. R. McGreevy et al. *Nature Sustainability*. 2022. Vol. 5, No. 12. P. 1011–1017.
4. Zurek M., Hebinck A., Selomane O. Climate change and the urgency to transform food systems. *Science*. 2022. Vol. 376, No. 6600. P. 1416–1421.
5. Болтянська Н. І., Маніта І. Ю., Подашевська О. Проблеми і перспективи розвитку інформаційних технологій в сільському господарстві. *Праці Таврійського державного агротехнологічного університету*. 2020. № 20(4). С. 175–185.
6. Human health risk assessment on the consumption of apples growing in urbanized areas: Case of Kharkiv, Ukraine / Y. Medvedeva et al. *International Journal of Environmental Research and Public Health*. 2021. Vol. 18, No. 4. P. 1504.
7. Шевчук Г., Гонтарук Я., Гіренко М. Державне регулювання розвитку садівництва в Україні. *Економіка та суспільство*. 2022. № 43. С. 1–9.
8. Колокольчикова І. В. Промислове садівництво Півдня України в рамках забезпечення продовольчої безпеки. *Science and Education a New Dimension*. 2019. VII(24), № 200. С. 7–10.
9. Development of sustainable plant protection programs through multi-actor Co-innovation: An 8-year case study in Swedish apple production / W. Swiergiel et al. *Journal of Cleaner Production*. 2019. Vol. 234. P. 1178–1191.
10. Fielke S., Taylor B., Jakku E. Digitalisation of agricultural knowledge and advice networks: A state-of-the-art review. *Agricultural Systems*. 2020. Vol. 180. P. 102763.

11. Unmanned aerial vehicles in agriculture: A review of perspective of platform, control, and applications / J. Kim et al. *IEEE Access*. 2019. Vol. 7. P. 105100–105115.
12. Лавренко С. О., Бурим М. І. Застосування безпілотних літальних апаратів у сільському господарстві. *Підземні води як стратегічний ресурс економічного розвитку держави* : матеріали міжнар. наук.-практ. онлайн-конф., м. Київ, 22 бер. 2022 р. / під заг. ред. Я.М. Гадзало. Київ: Інститут водних проблем і меліорації НААН, 2022. С. 60–61.
13. Internet of things (IoT) and agricultural unmanned aerial vehicles (UAVs) in smart farming: A comprehensive review / A. D. Boursianis et al. *Internet of Things*. 2022. Vol. 18. P. 100187.
14. Data acquisition and analysis methods in uav- based applications for precision agriculture / D. C. Tsouros et al. *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS-2019)* : Proceedings. (Santorini, Greece, 19 August 2019). IEEE, Inc., 2019. P. 377–384.
15. Drones in agriculture: A review and bibliometric analysis / A. Rejeb et al. *Computers and Electronics in Agriculture*. 2022. Vol. 198. P. 107017.
16. Crop monitoring in smallholder farms using unmanned aerial vehicles to facilitate precision agriculture practices: A scoping review and bibliometric analysis / S. Gokool et al. *Sustainability*. 2023. Vol. 15, No. 4. P. 3557.
17. Ali A., Imran M. Remotely sensed real-time quantification of biophysical and biochemical traits of Citrus (*Citrus sinensis* L.) fruit orchards – A review. *Scientia Horticulturae*. 2021. Vol. 282. P. 110024.
18. Мехед О. Т., Защепкіна Н. М. Дрони (БПЛА), їх роль у сучасному житті та екології. *XV Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Погляд у майбутнє приладобудування»* : тези XV всеукр. наук.-практ. конф., м. Київ, 14–15 чер. 2022 р. Київ: КПІ ім. Ігоря Сікорського, 2022. С. 128–130.
19. Іщенко В. С. Підвищення ефективності систем візуальної навігації БПЛА. *Наукові перспективи*. 2021. № 6(12). С. 23–31.

20. A comprehensive survey of the recent studies with UAV for precision agriculture in open fields and greenhouses / M. F. Aslan et al. *Applied Sciences*. 2022. Vol. 12, No. 3. P. 1047.

21. The potential of remote sensing and artificial intelligence as tools to improve the resilience of agriculture production systems / J. Jung et al. *Current Opinion in Biotechnology*. 2021. Vol. 70. P. 15–22.

22. Zhu J., Zhou J. et al. Key technology progress of plant-protection UAVs applied to mountain orchards: A review. S. Yu. *Agronomy*. 2022. Vol. 12, No. 11. P. 2828.

23. Improving estimation of LAI dynamic by fusion of morphological and vegetation indices based on UAV imagery / L. Qiao et al. *Computers and Electronics in Agriculture*. 2022. Vol. 192. P. 106603.

24. Exploring UAV-imagery to support genotype selection in olive breeding programs / P. Rallo et al. *Scientia Horticulturae*. 2020. Vol. 273. P. 109615.

25. A cloud-based environment for generating yield estimation maps from apple orchards using UAV imagery and a deep learning technique / O. E. Apolo-Apolo et al. *Frontiers in Plant Science*. 2020. Vol. 11. P. 1086.

26. Predicting individual apple tree yield using UAV multi-source remote sensing data and ensemble learning / R. Chen et al. *Computers and Electronics in Agriculture*. 2022. Vol. 201. P. 107275.

27. Gavahi K., Abbaszadeh P., Moradkhani H. DeepYield: A combined convolutional neural network with long short-term memory for crop yield forecasting. *Expert Systems with Applications*. 2021. Vol. 184. P. 115511.

28. FCN network-based weed and crop segmentation for IoT-Aided agriculture applications / S. Kamal et al. *Wireless Communications and Mobile Computing*. 2022. Vol. 2022. P. 1–10.

29. A five convolutional layer deep convolutional neural network for plant leaf disease detection / J. A. Pandian et al. *Electronics*. 2022. Vol. 11, No. 8. P. 1266.

30. Passion fruit detection and counting based on multiple scale faster R-CNN using RGB-D images / S. Tu et al. *Precision Agriculture*. 2020. Vol. 21, No. 5. P. 1072–1091.
31. Deep learning-based apple detection using a suppression mask R-CNN / P. Chu et al. *Pattern Recognition Letters*. 2021. Vol. 147. P. 206–211.
32. AgroScout: Precision agriculture data analytics for good industry. *Agroscout, Ltd.* 2023. URL: <https://agro-scout.com/> (дата звернення: 09.08.2022).
33. Aerobotics | Optimize and protect your yields, season after season. *Aerobotics, (Pty), Ltd.* 2023. URL: <https://www.aerobotics.com/> (дата звернення: 09.08.2022).
34. VineView: The leading provider of crop diagnostics for vineyards. *VineView, Ltd.* 2023. URL: <https://vineview.com/> (дата звернення: 10.08.2022).
35. Root AI – Join the future of farming. *Root AI, Inc.* 2023. URL: <https://root-ai.com/#intro> (дата звернення: 10.08.2022).
36. PIX4Dmapper: Professional photogrammetry software for drone mapping. *Pix4D S.A.* 2023. URL: <https://www.pix4d.com/product/pix4dmapper-photogrammetry-software/> (дата звернення: 10.08.2022).
37. A technical study on UAV characteristics for precision agriculture applications and associated practical challenges / N. Delavarpour et al. *Remote Sensing*. 2021. Vol. 13, No. 6. P. 1204.
38. Extraction of information about individual trees from high-spatial-resolution UAV-acquired images of an orchard / X. Dong et al. *Remote Sensing*. 2020. Vol. 12, No. 1. P. 133.
39. Opportunities of UAVs in orchard management / C. Zhang et al. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. 2019. Vol. XLII-2-W13. P. 673–680.
40. Comprehensive assessment of intelligent unmanned vehicle techniques in pesticide application: A case study in pear orchard / Y. Jiang et al. *Frontiers in Plant Science*. 2022. Vol. 13. P. 959429.
41. Challenges and opportunities for autonomous micro-UAVs in precision agriculture / X. Liu et al. *IEEE Micro*. Vol. 42, No. 1. P. 61–68.

42. Altaheri H., Alsulaiman M., Muhammad G. Date fruit classification for robotic harvesting in a natural environment using deep learning. *IEEE Access*. 2019. Vol. 7. P. 117115–117133.
43. Khan R., Rameswar D. Multi class fruit classification using efficient object detection and recognition techniques. *International Journal of Image, Graphics and Signal Processing*. 2019. Vol. 11, No. 8. P. 1–18.
44. Recent advancements in fruit detection and classification using deep learning techniques / C. C. Ukwuoma et al. *Mathematical Problems in Engineering*. 2022. Vol. 2022. P. 1–29.
45. Image convolution to obtain color ROI after segmentation process with fuzzy k-means / K. Anwar et al. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*. Vol. 7, No. 2. P. 376–380.
46. Радюк П. М. Аналітичний огляд архітектур згорткових нейронних мереж у задачах аналізу медичних зображень. *Science, Engineering and Technology: Global and Current Trends : Proceedings of inter. and pract. conf.*, м. Прага, 27–28 груд. 2019 р. Прага, 2019. С. 32–35.
47. Radiuk P. Applying 3D U-Net architecture to the task of multi-organ segmentation in computed tomography. *Applied Computer Systems*. 2020. Vol. 25, No. 1. P. 43–50.
48. Advanced fully convolutional networks for agricultural field boundary detection / A. Taravat et al. *Remote Sensing*. 2021. Vol. 13, No. 4. P. 722.
49. Gill H. S., Khehra B. S. An integrated approach using CNN-RNN-LSTM for classification of fruit images. *Materials Today: Proceedings*. 2022. Vol. 51, No. 1. P. 591–595.
50. Real-time apple detection system using embedded systems with hardware accelerators: An edge AI application / V. Mazzia et al. *IEEE Access*. 2020. Vol. 8. P. 9102–9114.
51. Barmak O., Radiuk P. Web-based information technology for classifying and interpreting early pneumonia based on fine-tuned convolutional neural network. *Computer systems and information technologies*. 2021. Vol. 3, No 1. P. 12–18.

52. Kang H., Chen C. Fast implementation of real-time fruit detection in apple orchards using deep learning. *Computers and Electronics in Agriculture*. 2020. Vol. 168. P. 105108.

53. Real-time fruit detection using deep neural networks on CPU (RTFD): An edge AI application / D. Mao et al. *Computers and Electronics in Agriculture*. 2023. Vol. 204. P. 107517.

54. Deep learning in object detection and recognition [1st ed.] / X. Jiang et al. Singapore : Springer Nature, Inc., 2019. 240 p.

55. Dhillon A., Verma G. K. Convolutional neural network: A review of models, methodologies and applications to object detection. *Progress in Artificial Intelligence*. 2020. Vol. 9, No. 2. P. 85–112.

56. Faster R-CNN with classifier fusion for automatic detection of small fruits / X. Mai et al. *IEEE Transactions on Automation Science and Engineering*. 2020. Vol. 17, No. 3. P. 1555–1569.

57. Ma Z., Li N. Improving apple detection using RetinaNet. *The International Conference on Image, Vision and Intelligent Systems (ICIVIS-2021)* : Lecture Notes in Electrical Engineering. Vol. 813. (Changsha, China, 15–17 June 2021). Singapore : Springer Nature, Inc., 2022. P. 131–141.

58. Lightweight apple detection in complex orchards using YOLOV5-PRE / L. Sun et al. *Horticulturae*. 2022. Vol. 8, No. 12. P. 1169.

59. Detection of tomato plant phenotyping traits using YOLOv5-based single stage detectors / A. Cardellicchio et al. *Computers and Electronics in Agriculture*. 2023. Vol. 207. P. 107757.

60. A novel apple fruit detection and counting methodology based on deep learning and trunk tracking in modern orchard / F. Gao et al. *Computers and Electronics in Agriculture*. 2022. Vol. 197. P. 107000.

61. Intelligent detection of multi-class pitaya fruits in target picking row based on WGB-YOLO network / Y. Nan et al. *Computers and Electronics in Agriculture*. 2023. Vol. 208. P. 107780.



62. Welch G. F. Kalman Filter. *Computer vision: A reference guide*. Boston : Springer Nature, Inc., 2020. P. 1–3.
63. Multiple Hungarian method for k-assignment problem / B. Gabrovšek et al. *Mathematics*. 2020. Vol. 8, No. 11. P. 2050.
64. Hashim Z. K., Shiker M. A. K. A new technique to solve the assignment problems. *AIP Conference Proceedings*. 2023. Vol. 2414, No. 1. P. 040018.
65. Abdelhamed A., Brubaker M., Brown M. Noise flow: Noise modeling with conditional normalizing flows. *2019 IEEE/CVF International Conference on Computer Vision (ICCV-2019) : Proceedings*. (Seoul, Korea (South), 27 October 2019 – 02 November 2019). IEEE, Inc., 2019. P. 3165–3173.
66. Ghorbani H. Mahalanobis distance and its application for detecting multivariate outliers. *Facta Universitatis, Series: Mathematics and Informatics*. 2019. Vol. 34, No. 3. P. 583–595.
67. Tharwat A. Classification assessment methods. *Applied Computing and Informatics*. 2020. Vol. 17, No. 1. P. 168–192.
68. Unmanned ground vehicles for smart farms / P. Gonzalez-De-Santos et al. *Agronomy – Climate Change & Food Security*. IntechOpen, 2020.
69. Savenko O., Lysenko S., Nicheporuk A. et al. Metamorphic viruses' detection technique based on the equivalent functional block search. *13th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge (ICTERI-2017) : CEUR-Workshop Proceedings*. Vol. 1844. (Kyiv, Ukraine, 15–18 May 2017). Aachen : CEUR-WS.org, 2017. P. 555–568.
70. Unmanned aerial vehicles in smart agriculture: Applications, requirements, and challenges / P. K. Reddy Maddikunta et al. *IEEE Sensors Journal*. 2021. Vol. 21, No. 16. P. 17608–17619.
71. Morgan G. R., Wang C., Morris J. T. RGB indices and canopy height modelling for mapping tidal marsh biomass from a small unmanned aerial system. *Remote Sensing*. 2021. Vol. 13, No. 17. P. 3406.
72. Kravchenko Y., Bondarenko V., Tyshchenko M. Model of information protection system database of the mobile terminals information system on the territory of

Ukraine. *2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T-2020) : Proceedings.* (Kharkiv, 6–9 October 2020). IEEE, Inc., 2020. P. 785–790.

73. Detection of GPS spoofing attack on unmanned aerial vehicle system / C. Liang et al. *Machine Learning for Cyber Security.* Cham : Springer Nature, Inc., 2019. P. 123–139.

74. A comprehensive review on recent applications of unmanned aerial vehicle remote sensing with various sensors for high-throughput plant phenotyping / L. Feng et al. *Computers and Electronics in Agriculture.* 2021. Vol. 182. P. 106033.

75. Aggarwal S., Kumar N. Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. *Computer Communications.* 2020. Vol. 149. P. 270–299.

76. A review on early wildfire detection from unmanned aerial vehicles using deep learning-based computer vision algorithms / A. Bouguettaya et al. *Signal Processing.* 2022. Vol. 190. P. 108309.

77. Stott E., Williams R. D., Hoey T. B. Ground control point distribution for accurate kilometre-scale topographic mapping using an RTK-GNSS unmanned aerial vehicle and SfM photogrammetry. *Drones.* 2020. Vol. 4, No. 3. P. 55.

78. Wind measurement and simulation techniques in multi-rotor small unmanned aerial vehicles / P. Abichandani et al. *IEEE Access.* 2020. Vol. 8. P. 54910–54927.

79. Design of differential GPS system based on BP neural network error correction for precision agriculture / G. Wu et al. *2019 Chinese Intelligent Automation Conference (CIAC-2019) : Lecture Notes in Electrical Engineering.* Vol. 568. Singapore : Springer Nature, Inc., 2020. P. 426–438.

80. Amami M. Fast and reliable vision-based navigation for real time kinematic applications. *International Journal for Research in Applied Science and Engineering Technology.* 2022. Vol. 10, No. 2. P. 922–932.

81. Pirtı A. Evaluating the accuracy of post-processed kinematic (PPK) positioning technique. *Geodesy and Cartography.* 2021. Vol. 47, No. 2. P. 66–70.

82. Harder P., Pomeroy J. W., Helgason W. D. Improving sub-canopy snow depth mapping with unmanned aerial vehicles: Lidar versus structure-from-motion techniques. *The Cryosphere*. 2020. Vol. 14, No. 6. P. 1919–1935.

83. Comparison of lithium ion batteries, hydrogen fueled combustion engines, and a hydrogen fuel cell in powering a small unmanned aerial vehicle / C. Depcik et al. *Energy Conversion and Management*. 2020. Vol. 207. P. 112514.

84. eBee×mapping drone – Drones | AgEagle Aerial Systems, Inc. *AgEagle Aerial Systems, Inc.* 2023. URL: <https://ageagle.com/drones/ebee-x/> (дата звернення: 15.06.2022).

85. Parrot Bluegrass Fields – AEROMOTUS. *AEROMOTUS, LLC*. 2023. URL: <https://www.aeromotus.com/product/parrot-bluegrass/> (дата звернення: 19.06.2022).

86. AGRAS T20 – DJI. *DJI Technology Co., Ltd.* 2023. URL: <https://www.dji.com/t20> (дата звернення: 20.06.2022).

87. DJI Phantom 4 RTK – DJI. *DJI Technology Co., Ltd.* 2023. URL: <https://www.dji.com/phantom-4-rtk> (дата звернення: 22.06.2022).

88. DJI GS Pro – DJI. *DJI Technology Co., Ltd.* 2023. URL: <https://www.dji.com/ground-station-pro> (дата звернення: 22.06.2022).

89. Moghadam P., Lowe T., Edwards E.J. Digital twin for the future of orchard production systems. *Proceedings*. 2020. Vol. 36, No. 1. P. 92.

90. Anagnostis A., Tagarakis A. C., Asiminari G. et al. A deep learning approach for anthracnose infected trees classification in walnut orchards. *Computers and Electronics in Agriculture*. 2021. Vol. 182. P. 105998.

91. A visual identification method for the apple growth forms in the orchard / J. Lv et al. *Computers and Electronics in Agriculture*. 2022. Vol. 197. P. 106954.

92. Utamima A., Djunaidy A. Agricultural routing planning: A narrative review of literature. *Procedia Computer Science*. 2022. Vol. 197. P. 693–700.

93. Optimization methods applied to motion planning of unmanned aerial vehicles: A review / A. Israr et al. *Drones*. 2022. Vol. 6, No. 5. P. 126.

94. Information system for public places and institutions visualization with opportunities of inclusive access and optimal routing / O. Pavlova et al. *Computer systems and information technologies*. 2022. Vol. 1, No 6. P. 62–68.
95. Route planning based on parallel optimization in the air-ground integrated network / K. Cai et al. *IEEE Transactions on Intelligent Transportation Systems*. 2023. P. 1–11.
96. Grid-based dynamic robust multi-objective brain storm optimization algorithm / Y. Guo et al. *Soft Computing*. 2020. Vol. 24, No. 10. P. 7395–7415.
97. Path-length analysis for grid-based path planning / J. P. Bailey et al. *Artificial Intelligence*. 2021. Vol. 301. P. 103560.
98. Feng X., Murray A. T. Allocation using a heterogeneous space Voronoi diagram. *Journal of Geographical Systems*. 2018. Vol. 20, No. 3. P. 207–226.
99. Hexagon-based generalized Voronoi diagrams generation for path planning of intelligent agents / F. Tang et al. *Mathematical Problems in Engineering*. 2020. Vol. 2020. P. 5750739.
100. Q-learning algorithms: A comprehensive classification and applications / B. Jang et al. *IEEE Access*. 2019. Vol. 7. P. 133653–133667.
101. Quality-oriented hybrid path planning based on A\* and Q-learning for unmanned aerial vehicle / D. Li et al. *IEEE Access*. 2022. Vol. 10, P. 7664–7674.
102. Reinforcement learning and its applications in modern power and energy systems: A review / D. Cao et al. *Journal of Modern Power Systems and Clean Energy*. 2020. Vol. 8, No. 6. P. 1029–1042.
103. An efficient hardware implementation of reinforcement learning: The Q-learning algorithm / S. Spanò et al. *IEEE Access*. 2019. Vol. 7. P. 186340–186351.
104. Ray A., Ray H. Proposing  $\epsilon$ -greedy reinforcement learning technique to self-optimize memory controllers. *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC-2021) : Proceedings*. (Jalandhar, India, 21–23 May 2021). IEEE, Inc., 2021. P. 318–323.
105. Regularized softmax deep multi-agent Q-learning / L. Pan et al. *Advances in Neural Information Processing Systems*. 2022. Vol. 34. P. 1365–1377.

106. Biomass and vegetation coverage survey in the Mu Us sandy land – based on unmanned aerial vehicle RGB images / Z. Guo et al. *International Journal of Applied Earth Observation and Geoinformation*. 2021. Vol. 94. P. 102239.

107. Gonçalves G., Andriolo U. Operational use of multispectral images for macro-litter mapping and categorization by unmanned aerial vehicle. *Marine Pollution Bulletin*. 2022. Vol. 176. P. 113431.

108. Retrieval of nitrogen content in apple canopy based on unmanned aerial vehicle hyperspectral images using a modified correlation coefficient method / M. Li et al. *Sustainability*. 2022. Vol. 14, No. 4. P. 1992.

109. Shakya S. Unmanned aerial vehicle with thermal imaging for automating water status in vineyard. *Journal of Electrical Engineering and Automation*. 2021. Vol. 3, No. 2. P. 79–91.

110. Robust detection, classification and localization of defects in large photovoltaic plants based on unmanned aerial vehicles and infrared thermography / A. Fernández et al. *Applied Sciences*. 2020. Vol. 10, No. 17. P. 5948.

111. Coupling of machine learning methods to improve estimation of ground coverage from unmanned aerial vehicle (UAV) imagery for high-throughput phenotyping of crops / P. Hu et al. *Functional Plant Biology*. 2021. Vol. 48, No. 8. P. 766–779.

112. Highway crack segmentation from unmanned aerial vehicle images using deep learning / Z. Hong et al. *IEEE Geoscience and Remote Sensing Letters*. 2022. Vol. 19. P. 1–5.

113. Kang H., Chen C. Fruit detection, segmentation and 3D visualisation of environments in apple orchards. *Computers and Electronics in Agriculture*. 2020. Vol. 171. P. 105302.

114. Mature pomegranate fruit detection and location combining improved F-PointNet with 3D point cloud clustering in orchard / T. Yu et al. *Computers and Electronics in Agriculture*. 2022. Vol. 200. P. 107233.

115. Multiview machine vision research of fruits boxes handling robot based on the improved 2D kernel principal component analysis network / X. Li et al. *Journal of Robotics*. 2021. Vol. 2021. P. e3584422.

116. On-tree apple fruit size estimation using stereo vision with deep learning-based occlusion handling / O. Mirbod et al. *Biosystems Engineering*. 2023. Vol. 226. P. 27–42.

117. Sara U., Akter M., Uddin M. S. Image quality assessment through FSIM, SSIM, MSE and PSNR—A comparative study. *Journal of Computer and Communications*. 2019. Vol. 7, No. 3. P. 8–18.

118. Журавська І. М. Теоретичні основи, методи та засоби створення та функціонування швидкодинамічних гетерогенних комп'ютерних мереж критичного застосування : дис. ... докт. техн. наук : 05.13.05. Миколаїв, 2019. 350 с.

119. Chapter 9—An internet of drone-based multi-version post-severe accident monitoring system: Structures and reliability / H. Fesenko et al. *Dependable IoT for Human and Industry* [1st ed.]. New York : River Publishers, 2022. P. 197–217.

120. Haar wavelet-based classification method for visual information processing systems / W. Huan et al. *Applied Sciences*. 2023. Vol. 13, No. 9. P. 5515.

121. Мельниченко О. В. Архітектура автоматизованої системи розпізнавання сукупності структурних об'єктів однієї природи в тривимірному просторі. *Вимірювальна та обчислювальна техніка в технологічних процесах*. 2022. № 4(2022). С. 128–133.

122. Мельниченко О. В. Автоматизована система самоорганізації для керування безпілотними літальними апаратами з метою виявлення об'єктів. *Вимірювальна та обчислювальна техніка в технологічних процесах*. 2023. № 1(2023). С. 116–122.

123. Мельниченко О. В. Методи збору та обробки зображень отриманих з використанням БПЛА для виявлення заданих об'єктів. *Вісник Хмельницького національного університету. Технічні науки*. 2022. № 6, Т. 1(315). С. 131–137.

124. Melnychenko O. Method of real-time video stream synchronization in the working environment of an apple orchard. *Computer systems and information technologies*. 2023. No. 1(10). P. 91–97.

125. Мельниченко О. Методи розпізнавання та обробки зображень за модифікованою YOLOv5-v1. *Information Technology: Computer Science, Software Engineering and Cyber Security*. 2023. Вип. 1. С. 74–84.

126. Мельниченко О. В. Метод обчислення кількості структурних об'єктів у фруктовому саду в реальному часі. *Вісник Хмельницького національного університету. Технічні науки*. 2023. № 2(319), Т. 1. С. 213–219.

127. Melnychenko O., Savenko O. A self-organized automated system to control unmanned aerial vehicles for object detection. . *The 4th International Workshop on Intelligent Information Technologies & Systems of Information Security (IntelITSIS-2023)* : CEUR-Workshop Proceedings. Vol. 3373. (Khmelnyskyi, Ukraine, 22–24 March 2023). Aachen : CEUR-WS.org, 2023. P. 589–600.

128. Мельниченко О. В. Метод та підсистема самовідновлення після критичних збоїв. *Актуальні проблеми комп'ютерних наук АПКН-2022»* : Матеріали XIV всеукр. наук.-практ. конф., м. Хмельницький, 18–19 лист. 2022 р. Хмельницький : ХНУ, 2022. С. 202–204.

129. Мельниченко О. В. Самоорганізована система управління декількома БПЛА для динамічного отримання зображень в тривимірному просторі. *Математичне та програмне забезпечення інтелектуальних систем (МПЗІС-2022)* : Тези доповідей XX міжнар. наук.-практ. конф., м. Дніпро, 23–25 лист. 2022 р. / під заг. ред. О.М. Кісельової. Дніпро : ДНУ, 2022. С. 139–140.

130. Мельниченко О. В. Метод керування та самоорганізована система планування маршрутів БПЛА. *2022 International Conference on Innovative Solutions in Software Engineering (ICISSE-2022)* : Conference Proceedings, Ivano-Frankivsk, Ukraine, 29–30 November 2022 / Kuz M., Kozenko M. eds. Ivano-Frankivsk : VSPNU, 2022. P. 26–30.

131. Мельниченко О. В. Метод обчислення кількості розпізнаних структурних об'єктів певного класу. *Комп'ютерні ігри та мультимедіа як інноваційний підхід до комунікації* : Матеріали II Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів, м. Одеса, 29–30 верес. 2022 р. Одеса : Видавництво ОНТУ, 2022 р. С. 124–126.

132. Dempsey P. Reviews – Consumer Technology. The Teardown – Apple iPhone Pro 13 smartphone. *Engineering & Technology*. 2021. Vol. 16, No. 11. P. 68–69.

133. Claid.ai: Stunning Product Photos for CPG and Marketplaces. *Let's Enhance, Inc.* 2023. URL: <https://claid.ai/> (дата звернення: 04.12.2021).

134. Mishra A. Amazon Sagemaker. *Machine learning in the AWS cloud: Add intelligence to applications with Amazon Sagemaker and Amazon Rekognition*. John Wiley & Sons, Inc., 2019. P. 353–385.

135. Welcome to Python.org. *Python.org*. 2023. URL: <https://www.python.org/> (дата звернення: 13.11.2021).

136. PyTorch: An imperative style, high-performance deep learning library / A. Paszke et al. *Advances in Neural Information Processing Systems* : Proceedings. Vol. 32. (Vancouver, BC, Canada, December 8–14, 2019). Curran Associates, Inc., 2019. P. 8024–8035.

137. CUDA Toolkit – Free Tools and Training. NVIDIA Developer. *NVIDIA, Inc.* 2023. URL: <https://developer.nvidia.com/cuda-toolkit> (дата звернення: 10.12.2021).

138. NVIDIA cuDNN. NVIDIA Developer. *NVIDIA, Inc.* 2023. URL: <https://developer.nvidia.com/cudnn> (дата звернення: 10.12.2021).

139. Gollapudi S. OpenCV with Python. Learn computer vision using OpenCV: With deep learning CNNs and RNNs. In. ed. S. Gollapudi. Berkeley, CA : Apress, 2019. P. 31–50.

140. Мельниченко О. В. А. С. 116949, Україна. Комп'ютерна програма «Автоматизована система розпізнавання та обчислення кількості структурних об'єктів однієї природи». Дата реєстрації 09.03.2023.



ДОДАТОК А.  
СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

*Наукові праці, в яких опубліковані основні наукові результати дисертації*

1. Мельниченко О. В. Архітектура автоматизованої системи розпізнавання сукупності структурних об'єктів однієї природи в тривимірному просторі. *Вимірювальна та обчислювальна техніка в технологічних процесах*. 2022. № 4(2022). С. 128–133. (внесений до переліку фахових видань України категорії «Б»). DOI: <https://doi.org/10.31891/2219-9365-2022-72-4-18>
2. Мельниченко О. В. Автоматизована система самоорганізації для керування безпілотними літальними апаратами з метою виявлення об'єктів. *Вимірювальна та обчислювальна техніка в технологічних процесах*. 2023. № 1(2023). С. 116–122. (внесений до переліку фахових видань України категорії «Б»). DOI: <https://doi.org/10.31891/2219-9365-2023-73-1-16>
3. Мельниченко О. В. Методи збору та обробки зображень отриманих з використанням БПЛА для виявлення заданих об'єктів. *Вісник Хмельницького національного університету. Технічні науки*. 2022. № 6, Т. 1(315). С. 131–137. (внесений до переліку фахових видань України категорії «Б»). DOI: <https://doi.org/10.31891/2307-5732-2022-315-131-138>
4. Melnychenko O. Method of real-time video stream synchronization in the working environment of an apple orchard. *Computer systems and information technologies*. 2023. No. 1(10). P. 91–97. DOI: <https://doi.org/10.31891/csit-2023-1-12>
5. Мельниченко О. Методи розпізнавання та обробки зображень за модифікованою YOLOv5-v1. *Information Technology: Computer Science, Software Engineering and Cyber Security*. 2023. Вип. 1. С. 74–84. (внесений до переліку фахових видань України категорії «Б»). DOI: <https://doi.org/10.32782/IT/2023-1-10>
6. Мельниченко О. В. Метод обчислення кількості структурних об'єктів у фруктовому саду в реальному часі. *Вісник Хмельницького національного університету. Технічні науки*. 2023. № 2(319), Т. 1. С. 213–219. (внесений до

переліку фахових видань України категорії «Б»). DOI: <https://doi.org/10.31891/2307-5732-2023-319-1-213-214>

*Праці, які засвідчують апробацію матеріалів дисертації*

7. Melnychenko O., Savenko O. A self-organized automated system to control unmanned aerial vehicles for object detection. *The 4th International Workshop on Intelligent Information Technologies & Systems of Information Security (IntelITSIS-2023)* : CEUR-Workshop Proceedings. Vol. 3373. (Khmelnyskyi, Ukraine, 22–24 March 2023). Aachen : CEUR-WS.org, 2023. P. 589–600 (*Scopus*). URL: <https://ceur-ws.org/Vol-3373/paper40.pdf>

8. Мельниченко О. В. Метод та підсистема самовідновлення після критичних збоїв. *Актуальні проблеми комп'ютерних наук АПКН-2022* : Матеріали XIV всеукр. наук.-практ. конф., м. Хмельницький, 18–19 лист. 2022 р. Хмельницький : ХНУ, 2022. С. 202–204. URL: [https://kn.khmnpu.edu.ua/wp-content/uploads/sites/18/apkn2022\\_corpuspaper.pdf](https://kn.khmnpu.edu.ua/wp-content/uploads/sites/18/apkn2022_corpuspaper.pdf)

9. Мельниченко О. В. Самоорганізована система управління декількома БПЛА для динамічного отримання зображень в тривимірному просторі. *Математичне та програмне забезпечення інтелектуальних систем (МПЗІС-2022)* : Тези доповідей XX міжнар. наук.-практ. конф., м. Дніпро, 23–25 лист. 2022 р. / під заг. ред. О.М. Кісельової. Дніпро : ДНУ, 2022. С. 139–140. URL: <http://mpzis.dnu.dp.ua/wp-content/uploads/2022/12/MPZIS-2022-1.pdf>

10. Мельниченко О. В. Метод керування та самоорганізована система планування маршрутів БПЛА. *2022 International Conference on Innovative Solutions in Software Engineering (ICISSE-2022)* : Conference Proceedings, Ivano-Frankivsk, Ukraine, 29–30 November 2022 / Kuz M., Kozenko M. eds. Ivano-Frankivsk : VSPNU, 2022. P. 26–30. URL: <https://shorturl.at/nyIMO>

11. Мельниченко О. В. Метод обчислення кількості розпізнаних структурних об'єктів певного класу. *Комп'ютерні ігри та мультимедіа як інноваційний підхід до комунікації* : Матеріали II Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів, м. Одеса, 29–30

верес. 2022 р. Одеса : Видавництво ОНТУ, 2022 р. С. 124–126. <https://card-file.ontu.edu.ua/handle/123456789/23485>

*Публікації, які додатково відображають наукові результати дисертації*

12. Мельниченко О. В. А. С. 116949, Україна. Комп'ютерна програма «Автоматизована система розпізнавання та обчислення кількості структурних об'єктів однієї природи». Дата реєстрації 09.03.2023.

## ДОДАТОК Б.

## ВІДОМОСТІ ПРО АПРОБАЦІЮ РЕЗУЛЬТАТІВ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Мельниченка Олександра Вікторовича

«Методи збору, розпізнавання та обробки зображень, отриманих із використанням БПЛА, для виявлення заданих об'єктів»  
за спеціальністю 122 Комп'ютерні науки

*Масові наукові, науково-теоретичні та науково-практичні заходи міжнародного рівня:*

1. 4th International Workshop on Intelligent Information Technologies & Systems of Information Security (Ukraine, Khmelnytskyi, Khmelnytskyi National University, 22–24 March 2023). Форма участі – виступ на секційному засіданні на тему: «*A self-organized automated system to control unmanned aerial vehicles for object detection*».

2. XX міжнародна науково-практична конференція «Математичне та програмне забезпечення інтелектуальних систем» (Україна, м. Дніпро, Дніпровський національний університет імені Олеся Гончара, 23–25 листопада 2022 р.). Форма участі – публікація тез на тему: «*Самоорганізована система управління декількома БПЛА для динамічного отримання зображень в тривимірному просторі*».

3. 2022 International Conference on Innovative Solutions in Software Engineering (Ukraine, Ivano-Frankivsk, Vasyl Stefanyk Precarpathian National University, 29–30 November 2022). Форма участі – виступ на секційному засіданні на тему: «*Метод керування та самоорганізована система планування маршрутів БПЛА*».

*Масові наукові, науково-теоретичні та науково-практичні заходи всеукраїнського рівня:*

1. XIV Всеукраїнська науково-практична конференція «Актуальні проблеми комп'ютерних наук АПКН-2022» (Україна, м. Хмельницький, Хмельницький національний університет, 18–19 листопада 2022 р.). Форма участі – публікація тез на тему: «*Метод та підсистема самовідновлення після критичних збоїв*».

2. II Всеукраїнська науково-технічна конференція молодих вчених, аспірантів та студентів «Комп'ютерні ігри та мультимедіа як інноваційний підхід до комунікації» (Україна, м. Одеса, Одеський національний технологічний університет, 29–30 вересня 2022 р.). Форма участі – публікація тез на тему: «*Метод обчислення кількості розпізнаних структурних об'єктів певного класу*».

ДОДАТОК В.  
АКТИ ВПРОВАДЖЕННЯ



«Затверджую»  
Директор ТОВ «ЮКС++»  
Кльоц Ю. П.  
травня 2023 р.

АКТ

про впровадження результатів дисертаційної роботи  
Мельниченка Олександра Вікторовича  
«Методи збору, розпізнавання та обробки зображень, отриманих із  
використанням БПЛА, для виявлення заданих об'єктів»

Результати дисертаційної роботи аспіранта кафедри комп'ютерної інженерії та інформаційних систем Хмельницького національного університету Мельниченка О. В. впроваджені на ТОВ «ЮКС++».

При впровадженні і розробці системи збору зображень та подальше розпізнавання заданих об'єктів на цих зображеннях, включаючи обробку результатів розпізнавання, були використані на ТОВ «ЮКС++» такі результати, які одержані Мельниченком О. В. особисто:

- 1) метод динамічного отримання зображень заданих структурних об'єктів в тривимірному просторі за допомогою декількох БПЛА, який на відміну від відомих відрізняється тим, що забезпечує в процесі активації кожного БПЛА генерацію відеоряду в центральну систему, прийняття рішень про подальшу роботу групи БПЛА, підтвердження цілісності програмної місії та визначення рівня критичності для продовження виконання роботи групи БПЛА за рахунок визначення станів модулів та комплексному врахуванні вимог розподілення, багаторівневості та автоматизованості, що покращує узгодження між різними БПЛА та досягнення переміщення всієї групи БПЛА від заданих початкових до кінцевих точок програмної місії автономно;
- 2) метод синхронізації відеопотоків в режимі реального часу, який відрізняється від відомих тим, що забезпечує накопичення даних про попередні програмні місії БПЛА, у випадку критичних збоїв, які спотворюють цілісність структур даних із джерел отримання відеопотоків, зберігає такі структури в спеціальному журналі помилок та не відправляє їх як вхідні параметри в наступну обробку, що дає змогу виконувати порівняння отриманих поточних результатів із минулими в режимі реального часу і це забезпечує оперативне отримання результатів та здійснення виявлення структурних об'єктів, які були пропущені в процесі минулих програмних місій;

3) метод виявлення заданих структурних об'єктів на зображеннях, який на відміну від оригінальної архітектури YOLOv5, полягає в тому, що модифіковано модуль фокусування нейронної мережі, видалено згортковий шар поєднуючий вхідну карту ознак з операцією конкатенції, актуалізовано механізм візуальної уваги для вилучення ознак, об'єднання шарів 4 і 15, 6 і 11, 10 і 21 оригінальної архітектури замінено на об'єднання шарів 5 і 18, 8 і 14, 13 і 24 вдосконаленої архітектури, вихідні карти ознак 14-го та 21-го шарів вдосконаленої архітектури об'єднано між собою, що дало змогу покращити точність виявлення і зменшити час навчання нейронної мережі.

Результати виконаних досліджень здобувачем включають методи, спосіб, алгоритми та засоби покращення ефективності збору, точності розпізнавання та обробки зображень, які отримані з використанням БПЛА. Це дало змогу створювати автоматизовані системи з використанням групи БПЛА для збору, отримання та розпізнавання заданих об'єктів на отриманих зображеннях.

Отриманні результати покращують ефективність збору, точність розпізнавання та обробки зображень, отриманих з використанням БПЛА, для виявлення заданих об'єктів порівняно з відомими аналогічними системами приблизно на 10-12%.

Цей акт не є підставою для фінансових розрахунків.

Директор



Юрій КЛЬОЦ



ЗАТВЕРДЖУЮ

Головний інженер, к.т.н., доцент

ДП «Іноватор»

Олександр ЯНОВИЦЬКИЙ

«10» травня 2023 р.

АКТ №041/1370

від 10.05.2023

м. Хмельницький

про впровадження результатів дисертаційної роботи  
Мельниченка Олександра Вікторовича

«Методи збору, розпізнавання та обробки зображень, отриманих із  
використанням БПЛА, для виявлення заданих об'єктів»

Результати дисертаційної роботи аспіранта кафедри комп'ютерної інженерії та інформаційних систем Хмельницького національного університету Мельниченка О. В. впроваджені на ДП «Іноватор» в КВ№4КЦ.

В процесі впровадження і розробки системи збору зображень групами БПЛА та подальше розпізнавання об'єктів на цих зображеннях і обробку результатів розпізнавання були використані на ДП «Іноватор» такі результати, які одержані Мельниченком О. В. особисто:

1) метод побудови маршрутів БПЛА згідно технологій самонавчання, який на відміну від відомих, полягає в позиціонуванні апаратних пристроїв у тривимірному координатному просторі в реальному часі за рахунок самонавчання при побудові маршрутів їх руху, що дає змогу покращити переміщення та синхронізацію між групою БПЛА або одного БПЛА в межах робочого сегменту і за рахунок цього збільшення кількості опрацьованих даних;

2) метод динамічного отримання зображень заданих структурних об'єктів в тривимірному просторі за допомогою декількох БПЛА, який на відміну від відомих відрізняється тим, що забезпечує в процесі активації кожного БПЛА генерацію відеоряду в центральну систему, прийняття рішень про подальшу роботу групи БПЛА; підтвердження цілісності програмної місії та визначення рівня критичності для продовження виконання роботи групи БПЛА за рахунок визначення станів модулів та комплексному врахуванні вимог розподілення, багаторівневості та автоматизованості.



За результатами виконаних досліджень здобувачем розроблено методи, спосіб, алгоритми та засоби покращення ефективності збору, точності розпізнавання та обробки зображень, отриманих з використанням БПЛА, для виявлення заданих об'єктів. Це дало змогу створювати автоматизовані системи з використанням групи БПЛА для збору, отримання та розпізнавання заданих об'єктів на отриманих зображеннях.

Отриманні результати дозволили покращити ефективність збору, точність розпізнавання та обробки зображень, отриманих з використанням БПЛА, для виявлення заданих об'єктів. Результати роботи використано КВ№4КЦ для розробки систем керування групою БПЛА та розпізнавання заданих об'єктів.

Цей акт не є підставою для фінансових розрахунків.

Начальник відділу КВ№4КЦ



Віктор АВДЕЄВ

Провідний інженер-конструктор



Андрій МАРТИНОВ

Технічний



«Затверджую»

«НОЛТ ТЕХНОЛОДЖИС»

Свистун С.О.

«25» травня 2023 р.

## АКТ

про впровадження результатів дисертаційної роботи  
Мельниченка Олександра Вікторовича  
«Методи збору, розпізнавання та обробки зображень отриманих з використанням  
БПЛА для виявлення заданих об'єктів»

Результати дисертаційної роботи аспіранта кафедри комп'ютерної інженерії та інформаційних систем Хмельницького національного університету Мельниченка О. В. впроваджені на ПП «НОЛТ ТЕХНОЛОДЖИС».

При розробці системи збору зображень та подальше розпізнавання заданих об'єктів на цих зображеннях, включаючи обробку результатів розпізнавання, були використані на ПП «НОЛТ ТЕХНОЛОДЖИС» такі результати, які одержані Мельниченком О. В. особисто:

- 1) метод динамічного отримання зображень заданих структурних об'єктів в тривимірному просторі за допомогою декількох БПЛА, який на відміну від відомих відрізняється тим, що забезпечує в процесі активації кожного БПЛА генерацію відеоряду в центральну систему, прийняття рішень про подальшу роботу групи БПЛА, підтвердження цілісності програмної місії та визначення рівня критичності для продовження виконання роботи групи БПЛА за рахунок визначення станів модулів та комплексному врахуванні вимог розподілення, багаторівневості та автоматизованості, що покращує узгодження між різними БПЛА та досягнення переміщення всієї групи БПЛА від заданих початкових до кінцевих точок програмної місії автономно;
- 2) метод виявлення заданих структурних об'єктів на зображеннях, який на відміну від оригінальної архітектури YOLOv5, полягає в тому, що модифіковано модуль фокусування нейронної мережі, видалено згортковий шар поєднуючий вхідну карту ознак з операцією конкатенції, актуалізовано механізм візуальної уваги для вилучення ознак, об'єднання шарів 4 і 15, 6 і 11, 10 і 21 оригінальної архітектури замінено на об'єднання шарів 5 і 18, 8 і 14, 13 і 24 вдосконаленої архітектури, вихідні карти ознак 14-го та 21-го шарів вдосконаленої архітектури об'єднано між собою, що дало змогу покращити точність виявлення і зменшити час навчання нейронної мережі.

Виконані дослідження включають методи, спосіб, алгоритми та засоби покращення ефективності збору, точності розпізнавання та обробки зображень, які отримані з використанням БПЛА, що дало змогу створювати автоматизовані системи з використанням групи БПЛА для збору, отримання та розпізнавання заданих об'єктів на отриманих зображеннях.

Отриманні результати покращують ефективність збору, точність розпізнавання та обробки зображень, отриманих з використанням БПЛА, для виявлення заданих об'єктів порівняно з відомими аналогічними системами приблизно на 8-12%.

Цей акт не є підставою для фінансових розрахунків.



Свистун С.О.

A handwritten signature in blue ink, appearing to be 'Свистун С.О.', written over the printed name.



«Затверджую»  
Директор ТОВ «Агротех сервіс»

«22» травня 2023 р.

*Каспер В. І.*

### АКТ

про впровадження результатів дисертаційної роботи  
Мельниченка Олександра Вікторовича  
«Методи збору, розпізнавання та обробки зображень отриманих з  
використанням БПЛА для виявлення заданих об'єктів»

Результати дисертаційної роботи аспіранта кафедри комп'ютерної інженерії та інформаційних систем Хмельницького національного університету Мельниченка О. В. впроваджені на ТОВ «Агротех сервіс».

В процесі впровадження і розробки системи збору зображень групами БПЛА та подальше розпізнавання об'єктів на цих зображеннях і обробку результатів розпізнавання були використані на ТОВ «Агротех сервіс» такі результати, які одержані Мельниченком О. В. особисто:

- 1) метод побудови маршрутів БПЛА згідно технологій самонавчання, який на відміну від відомих, полягає в позиціюванні апаратних пристроїв у тривимірному координатному просторі в реальному часі за рахунок самонавчання при побудові маршрутів їх руху, що дає змогу покращити переміщення та синхронізацію між групою БПЛА або одного БПЛА в межах робочого сегменту;
- 2) метод динамічного отримання зображень заданих структурних об'єктів в тривимірному просторі за допомогою декількох БПЛА, який на відміну від відомих відрізняється тим, що забезпечує в процесі активації кожного БПЛА генерацію відеоряду в центральну систему та досягнення переміщення всієї групи БПЛА від заданих початкових до кінцевих точок програмної місії автономно;
- 3) метод синхронізації відеопотоків в режимі реального часу, який відрізняється від відомих тим, що забезпечує накопичення даних про попередні програмні місії БПЛА, у випадку критичних збоїв, які спотворюють цілісність структур даних із джерел отримання відеопотоків, зберігає такі структури в спеціальному журналі помилок та не відправляє їх як вхідні параметри в наступну обробку, що дає змогу виконувати порівняння отриманих поточних результатів із минулими в режимі реального часу і це забезпечує оперативне отримання результатів та здійснення виявлення структурних об'єктів, які були пропущені в процесі минулих програмних місій;

4) метод виявлення заданих структурних об'єктів на зображеннях, який на відміну від оригінальної архітектури YOLOv5, полягає в тому, що модифіковано модуль фокусування нейронної мережі, видалено згортковий шар поєднуючий вхідну карту ознак з операцією конкатенції, актуалізовано механізм візуальної уваги для вилучення ознак, об'єднання шарів вдосконаленої архітектури, що дало змогу покращити точність виявлення і зменшити час навчання нейронної мережі.

За результатами виконаних досліджень здобувачем розроблено методи, спосіб, алгоритми та засоби покращення ефективності збору, точності розпізнавання та обробки зображень, отриманих з використанням БПЛА, для виявлення заданих об'єктів. Це дало змогу створювати автоматизовані системи з використанням групи БПЛА для збору, отримання та розпізнавання заданих об'єктів на отриманих зображеннях.

Отриманні результати дозволили покращити ефективність збору, точність розпізнавання та обробки зображень, отриманих з використанням БПЛА, для виявлення заданих об'єктів. Результати роботи використано безпосередньо при здійсненні обліку врожаю яблук. Використання такої системи надало можливість провести оцінювання отриманого врожаю, провести порівняння врожаїв за декілька років та спрогнозувати подальшу роботу.

Цей акт не є підставою для фінансових розрахунків.



Гасенко В. І.



«Затверджую»

Проректор з науково-педагогічної роботи

Віктор ЛОПАТОВСЬКИЙ

«02» травня 2023 р.

## АКТ

про впровадження в навчальний процес Хмельницького національного університету результатів дисертаційної роботи аспіранта кафедри комп'ютерної інженерії та інформаційних систем Мельниченка Олександра Вікторовича «Методи збору, розпізнавання та обробки зображень, отриманих із використанням БПЛА, для виявлення заданих об'єктів»

Ми, комісія в складі: декана факультету інформаційних технологій, д.т.н., професора Савенка О. С., завідувача кафедри комп'ютерної інженерії та інформаційних систем, д.т.н., професора Говорущенко Т. О., завідувача кафедри комп'ютерних наук, д.т.н., професора Бармака О. В. склала акт про те, що результати дисертаційної роботи Мельниченка О. В. впроваджені та використовуються в освітньому процесі на кафедрі комп'ютерної інженерії та інформаційних систем для спеціальності 126 Інформаційні системи та технології, 123 Комп'ютерна інженерія та кафедрі комп'ютерних наук для спеціальності 122 Комп'ютерні науки, зокрема в курсах «Методи та системи штучного інтелекту», «Комп'ютерні та кіберфізичні системи», «Методи, засоби та алгоритми в задачах обчислювального інтелекту та комп'ютерного зору», «Теорія, проектування та моделювання спеціалізованих комп'ютерних систем» та «Технології проектування інформаційних систем».

При викладанні цих дисциплін викладачами кафедр використовувалися наступні матеріали досліджень, отримані Мельниченком О. В. особисто:

1) метод побудови маршрутів БПЛА згідно технологій самонавчання, який на відміну від відомих, полягає в позиціюванні апаратних пристроїв у тривимірному координатному просторі в реальному часі за рахунок самонавчання при побудові маршрутів їх руху;

2) метод динамічного отримання зображень заданих структурних об'єктів в тривимірному просторі за допомогою декількох БПЛА, який на відміну від відомих відрізняється тим, що забезпечує в процесі активації кожного БПЛА генерацію відеоряду в центральну систему, прийняття рішень про подальшу роботу групи БПЛА, підтвердження цілісності програмної місії та визначення рівня критичності для продовження виконання роботи групи БПЛА;

3) метод синхронізації відеопотоків в режимі реального часу, що забезпечує накопичення даних про попередні програмні місії БПЛА, у випадку критичних збоїв, які спотворюють цілісність структур даних із джерел отримання відеопотоків, зберігає такі структури в спеціальному журналі помилок та не відправляє їх як вхідні параметри в наступну обробку;

4) метод виявлення заданих структурних об'єктів на зображеннях, який на відміну від оригінальної архітектури YOLOv5, полягає в тому, що модифіковано модуль фокусування нейронної мережі, видалено згортковий шар поєднуючий вхідну карту ознак з операцією конкатенції, актуалізовано механізм візуальної уваги для вилучення ознак;

5) розроблена автоматизована система з використанням групи БПЛА для збору, отримання та розпізнавання заданих об'єктів на отриманих зображеннях.

Отримані матеріали досліджень дозволили розробити лабораторні практикуми з використанням методів для розпізнавання образів заданих об'єктів, метод вибору маршруту та організація безпроводного зв'язку між обчислювальною системою та БПЛА.



*[Handwritten signature]*  
 \_\_\_\_\_  
*[Handwritten signature]*  
 \_\_\_\_\_  
*[Handwritten signature]*  
 \_\_\_\_\_

Савенко О. С.

Говорущенко Т. О.

Бармак О. В..

ДОДАТОК Г.  
КОМЕРЦІЙНІ ПЛАТФОРМИ ЦИФРОВОГО САДІВНИЦТВА

Таблиця Г.1

Комерційно доступні платформи на основі систем штучного інтелекту для  
проведення цифрового садівництва

Назва	Переваги	Недоліки
AgroScout [32]	<ol style="list-style-type: none"> <li>1. Висока точність виявлення та обчислення кількості плодів.</li> <li>2. Проста у використанні платформа та зручний інтерфейс.</li> <li>3. Гнучкі налаштування для різних типів фруктових садів і погодних умов.</li> </ol>	<ol style="list-style-type: none"> <li>1. Обмежена інтеграція з іншими технологіями цифрового садівництва.</li> <li>2. Обмежені типи культур, що можуть розпізнаватися.</li> <li>3. Дорога абонентська плата.</li> </ol>
Aerobotics [33]	<ol style="list-style-type: none"> <li>1. Інтеграція з іншими технологіями цифрового садівництва (наприклад, картографування врожайності).</li> <li>2. Сповіщення в реальному часі про спалахи шкідників і хвороб.</li> <li>3. Зручний користувацький інтерфейс.</li> </ol>	<ol style="list-style-type: none"> <li>1. Використання обмежено певними географічними регіонами (окремі регіони США та Південної Африки).</li> <li>2. Доступні лише кілька культур.</li> <li>3. Дорога абонентська плата.</li> </ol>
VineView [34]	<ol style="list-style-type: none"> <li>1. Висока точність виявлення плодів.</li> <li>2. Зручний інтерфейс та гнучкі налаштування.</li> <li>3. Сповіщення в реальному часі про спалахи шкідників і хвороб.</li> </ol>	<ol style="list-style-type: none"> <li>1. Використання обмежено виноградниками та виноградними культурами.</li> <li>2. Дорога абонентська плата.</li> <li>3. Обмежено окремими регіонами в Північній Америці та Європі.</li> </ol>



## Продовження таблиці Г.1

RootAI [35]	<ol style="list-style-type: none"> <li>1. Добра точність виявлення та обчислення кількості плодів.</li> <li>2. Інтеграція з іншими технологіями цифрового землеробства, як от, моніторинг погоди, аналіз ґрунту.</li> <li>3. Сповіщення в реальному часі про спалахи шкідників і хвороб.</li> </ol>	<ol style="list-style-type: none"> <li>1. Використання обмеженої кількості типів культур.</li> <li>2. Обмежено окремими регіонами в Індії та Африки.</li> <li>3. Дорога абонентська плата.</li> </ol>
Pix4Dmapper [36]	<ol style="list-style-type: none"> <li>1. Висока точність виявлення та обчислення кількості плодів.</li> <li>2. Інтеграція з іншими технологіями цифрового землеробства (аналіз NDVI, 3D-картування).</li> <li>3. Гнучкі налаштування для різних типів садів і погодних умов.</li> </ol>	<ol style="list-style-type: none"> <li>1. Обмежено окремими регіонами США та Європи.</li> <li>2. Дорога абонентська плата.</li> <li>3. Доступна обмежена кількість типів культур.</li> </ol>

Таблиця Г.2

## Опис технічних характеристик моделей БПЛА

Назва моделі БПЛА	Вага, кг	Місткість батареї, mAh	Тривалість польоту, хв	Технічні характеристики камери	Заводське призначення
senseFly eBee X [84]	1,1	4900	90	<ul style="list-style-type: none"> <li>– Resolution: 24 MP.</li> <li>– Video Resolution: 4K at 60 fps.</li> </ul>	Аерофотокартування та геодезія

## Продовження таблиці Г.2

Parrot Bluegrass Fields [85]	0,5	6700	25	– Resolution: 14 MP. – Video Resolution: 1080p HD at 30 fps.	Аерофото- картування та моніторинг агрокультур
DJI Agras T20 [86]	21,1	18000	23	– FOV: Horizontal: 98°, Vertical: 78°. – Video Resolution: 1080p HD at 30 fps.	Обприску- вання посівів
DJI Phantom 4 RTK [87]	1,388	5870	28	– Resolution: 20 MP. – Video Resolution: 4K at 60 fps.	Моніторинг агрокультур

## ДОДАТОК Д. ЛІСТИНГ ПРОГРАМНОГО КОДУ

Лістинг програмного коду, що реалізує запропонований у роботі метод побудови маршрутів БПЛА згідно технологій самонавчання.

```

struct Position{
public int X { get; set; }
public int Y { get; set; }
public int Z { get; set; }
}

public sealed class TreePart{
public int Width { get; set; }
public int Height { get; set; }
public Position StartPosition { get; set; }
public Position EndPosition { get; set; }
public int ActualFruitCount { get; set; }
public TreePart(int width, int height, Position startPosition, Position endPosition) {
Width = width;
Height = height;
StartPosition = startPosition;
EndPosition = endPosition;
ActualFruitCount = 0;
}
}

public sealed class Tree{
public TreePart Top { get; set; }
public TreePart Middle { get; set; }
public TreePart Bottom { get; set; }
public Position Position { get; set; }
public Tree(TreePart top, TreePart middle, TreePart bottom, Position position){
Top = top;
Middle = middle;
Bottom = bottom;
Position = position;}
}

using DJI.Sdk.Camera;
using DJI.Sdk.Managers;
using DJI.Sdk.Streaming;
using System;

public sealed class Mission{
private Drone[] drones;
private TreeWrapper[] treeWrappers;
private Guid missionId;
private DateTime creationDate;
private DateTime completionDate;
public Mission(){
drones = new Drone[0];
}
}

```

```

treeWrappers = new TreeWrapper[0];
missionId = Guid.NewGuid();
creationDate = DateTime.Now;
completionDate = DateTime.MinValue;}
public Guid MissionId{
get { return missionId; }}
public DateTime CreationDate{
get { return creationDate; }}
public DateTime CompletionDate{
get { return completionDate; }
set { completionDate = value; }
}
public void AddDrone(Drone drone){
Array.Resize(ref drones, drones.Length + 1);
drones[drones.Length - 1] = drone;}
public void AddTreeWrapper(TreeWrapper treeWrapper){
Array.Resize(ref treeWrappers, treeWrappers.Length + 1);
treeWrappers[treeWrappers.Length - 1] = treeWrapper;
}
public void Start(){
// Connecting to the drones and setting up DJI SDK
foreach (Drone drone in drones){
// Getting the drone manager
DroneManager droneManager = DjiSdk.GetDroneManager();
// Selecting a drone
Drone drone = droneManager.GetConnectedDrones()[0];
Camera camera = drone.GetCamera();
VideoStreamingManager videoStreamingManager = camera.GetVideoStreamingManager();
videoStreamingManager.EnableVideoData();
videoStreamingManager.StartStream();
videoStreamingManager.OnVideoDataReceived += VideoDataReceivedHandler;
videoStreamingManager.StopStream();
VideoStreamingManager videoStreamingManager = camera.GetVideoStreamingManager();
videoStreamingManager.EnableVideoData();
videoStreamingManager.StartStream();
videoStreamingManager.OnVideoDataReceived += VideoDataReceivedHandler;
}
foreach (Drone drone in drones){
Camera camera = drone.GetCamera();
VideoStreamingManager videoStreamingManager = camera.GetVideoStreamingManager();
videoStreamingManager.StopStream();}
completionDate = DateTime.Now;}
}
public class DroneCommandActor {
private readonly IHubContext<DroneCommandsHub> _commandsHubContext;
private readonly ILoggingService _loggingService;
public DroneCommandActor(
IHubContext<DroneCommandsHub> commandsHubContext,
ILoggingService loggingService) {
_commandsHubContext = commandsHubContext;
_loggingService = loggingService;
}
public async Task<bool> TryStartTakeOffAsync(TakeOffCommandMessage commandMessage) {
try {await _commandsHubContext.Clients.All
.SendAsync(HubEvents.DroneTakeOffCommandEvent, commandMessage);

```

```

return true;
} catch (System.Exception exc) {
return false;}
}
public async Task<bool> TryStartLandingAsync(LandingCommandMessage commandMessage) {
try {await _commandsHubContext
.Clients.All
.SendAsync(HubEvents.DroneLandingCommandEvent, commandMessage);
return true;
} catch (System.Exception exc) {
return false;}}}}
namespace Harvested.AI.Actors {
public sealed class DroneConnectionServerActor : UntypedActor {
private readonly ILoggingService _loggingService;
private readonly IFlyingSessionService _flyingSessionService;
private readonly IDroneService _droneService;
public DroneConnectionServerActor(ILoggingService loggingService, IDroneService droneService,
IFlyingSessionService flyingSessionService) {
Context.System.Tcp().Tell(new Tcp.Bind(Self, new IPEndPoint(IPAddress.Any,
ConfigurationManager.DroneListeningPort)));

_loggingService = loggingService;
_flyingSessionService = flyingSessionService;
_droneService = droneService;}
protected override void OnReceive(object message) {
if (message is Tcp.Bound bound) {
Debug.WriteLine($"Listening on {bound.LocalAddress}");
} else if (message is Tcp.Connected connected) {
IPEndPoint ip = (IPEndPoint)connected.RemoteAddress;
var tcpReceiveProps = Props.Create(() =>
new TcpConnectionEncryptionActor(Self, ip, _droneService, _flyingSessionService, _loggingService));
Sender.Tell(new Tcp.Register(Context.ActorOf(tcpReceiveProps)));
} else Unhandled(message);}
}
namespace Harvested.AI.Actors {
public class DroneInfoDeliveringActor : ReceiveActor {
private readonly IHubContext<CoordinatesHub> _coordinatesHubContext;
private readonly ILoggingService _loggingService;
public DroneInfoDeliveringActor(
IHubContext<CoordinatesHub> coordinatesHubContext,
ILoggingService loggingService) {
_coordinatesHubContext = coordinatesHubContext;
_loggingService = loggingService;
Receive<DroneIncomeCoordinateMessage>(message => SendCoordinates(message));
Receive<DroneStateChangedIncomeMessage>(message => SendDroneState(message));
Receive<SessionStateChangedMessage>(message => SendFlyingSessionNotification(message));
Receive<VideoFeedUpdated>(message => SendVideoPart(message));}
private void SendFlyingSessionNotification(SessionStateChangedMessage message) {
try {_coordinatesHubContext
.Clients
.Group(message.Message.OwnerId.ToString())
.SendAsync(
message.HubMethod,
message.Message)
.PipeTo(Self);

```

```

} catch (Exception exc) {
_loggingService.LogError(exc.Message, exc);}
}
private void SendDroneState(DroneStateChangedIncomeMessage message) {
try {
_coordinatesHubContext
.Clients
.Group(message.Message.UserIdentity.Id.ToString())
.SendAsync(
message.HubMethod,
message.Message)
.PipeTo(Self);
} catch (Exception exc) {
_loggingService.LogError(exc.Message, exc);}}
private void SendCoordinates(DroneIncomeCoordinateMessage coordinates) {
try {
_coordinatesHubContext
.Clients
.Group(coordinates.OwnerId.ToString())
.SendAsync(
coordinates.HubMethod,
new[] { coordinates.Message })
.PipeTo(Self);
} catch (System.Exception exc) {
_loggingService.LogError(exc.Message, exc);}}
private void SendVideoPart(VideoFeedUpdated part) {
try {_coordinatesHubContext.Clients
.Group(part.OwnerId.ToString())
.SendAsync(HubEvents.DroneVideoFeedUpdatedHubEvent, new[] { part.Message })
.PipeTo(Self);
} catch (System.Exception exc) {
_loggingService.LogError(exc.Message, exc);}}
}

```

```

CREATE TABLE Missions (
  MissionId INT PRIMARY KEY,
  Name NVARCHAR(255) NOT NULL,
  StartDate DATETIME NOT NULL,
  EndDate DATETIME NOT NULL
);

```

```

CREATE TABLE Drones (
  DroneId INT PRIMARY KEY,
  Name NVARCHAR(255) NOT NULL,
  MissionId INT NOT NULL,
  FOREIGN KEY (MissionId) REFERENCES Missions(MissionId)
);

```

```

CREATE TABLE VideoFrames (
  FrameId INT PRIMARY KEY,
  DroneId INT NOT NULL,
  Rotation BIGINT NOT NULL,

```

```

    CreatedAt DATETIME NOT NULL,
    ResolutionWidth INT NOT NULL,
    ResolutionHeight INT NOT NULL,
    FrameRate INT NOT NULL,
    CodecFormat NVARCHAR(255) NOT NULL,
    FOREIGN KEY (DroneId) REFERENCES Drones(DroneId)
);

CREATE TABLE Trees (
    TreeId INT PRIMARY KEY,
    MissionId INT NOT NULL,
    StartCoordinateX FLOAT NOT NULL,
    StartCoordinateY FLOAT NOT NULL,
    EndCoordinateX FLOAT NOT NULL,
    EndCoordinateY FLOAT NOT NULL,
    FOREIGN KEY (MissionId) REFERENCES Missions(MissionId)
);

CREATE PROCEDURE dbo.Mission_CRUD
    @MissionId INT,
    @Name NVARCHAR(255),
    @StartDate DATETIME,
    @EndDate DATETIME,
    @Operation NVARCHAR(10)
AS
BEGIN
    SET NOCOUNT ON;

    IF @Operation = 'CREATE'
    BEGIN
        -- Insert a new mission
        INSERT INTO Missions (MissionId, Name, StartDate, EndDate)
        VALUES (@MissionId, @Name, @StartDate, @EndDate);
    END

    IF @Operation = 'READ'
    BEGIN
        -- Select a mission by MissionId
        SELECT * FROM Missions WHERE MissionId = @MissionId;
    END

    IF @Operation = 'UPDATE'
    BEGIN
        -- Update a mission by MissionId
        UPDATE Missions
        SET Name = @Name,
            StartDate = @StartDate,
            EndDate = @EndDate
        WHERE MissionId = @MissionId;
    END

    IF @Operation = 'DELETE'
    BEGIN
        -- Delete a mission by MissionId
        DELETE FROM Missions WHERE MissionId = @MissionId;
    END

```

```

END
END

```

```

CREATE PROCEDURE dbo.Drone_CRUD
    @DroneId INT,
    @Name NVARCHAR(255),
    @MissionId INT,
    @Operation NVARCHAR(10)
AS
BEGIN
    SET NOCOUNT ON;

    IF @Operation = 'CREATE'
    BEGIN
        -- Insert a new drone
        INSERT INTO Drones (DroneId, Name, MissionId)
        VALUES (@DroneId, @Name, @MissionId);
    END

    IF @Operation = 'READ'
    BEGIN
        -- Select a drone by DroneId
        SELECT * FROM Drones WHERE DroneId = @DroneId;
    END

    IF @Operation = 'UPDATE'
    BEGIN
        -- Update a drone by DroneId
        UPDATE Drones
        SET Name = @Name,
            MissionId = @MissionId
        WHERE DroneId = @DroneId;
    END

    IF @Operation = 'DELETE'
    BEGIN
        DELETE FROM Drones WHERE DroneId = @DroneId;
    END
END

```

```

CREATE PROCEDURE dbo.VideoFrame_CRUD
    @FrameId INT,
    @DroneId INT,
    @Rotation BIGINT,
    @CreatedAt DATETIME,
    @ResolutionWidth INT,
    @ResolutionHeight INT,
    @FrameRate INT,
    @CodecFormat NVARCHAR(255),
    @Operation NVARCHAR(10)
AS
BEGIN
    SET NOCOUNT ON;

    IF @Operation = 'CREATE'

```



```

BEGIN
    INSERT INTO VideoFrames (FrameId, DroneId, Rotation, CreatedAt, ResolutionWidth, ResolutionHeight,
FrameRate, CodecFormat)
        VALUES (@FrameId, @DroneId, @Rotation, @CreatedAt, @ResolutionWidth, @ResolutionHeight,
@FrameRate, @CodecFormat);
    END

    IF @Operation = 'READ'
    BEGIN
        SELECT * FROM VideoFrames WHERE

FrameId = @FrameId;
    END

    IF @Operation = 'UPDATE'
    BEGIN
        UPDATE VideoFrames
        SET DroneId = @DroneId,
            Rotation = @Rotation,
            CreatedAt = @CreatedAt,
            ResolutionWidth = @ResolutionWidth,
            ResolutionHeight = @ResolutionHeight,
            FrameRate = @FrameRate,
            CodecFormat = @CodecFormat
        WHERE FrameId = @FrameId;
    END

    IF @Operation = 'DELETE'
    BEGIN
        DELETE FROM VideoFrames WHERE FrameId = @FrameId;
    END

    END

CREATE PROCEDURE dbo.Tree_CRUD
    @TreeId INT,
    @MissionId INT,
    @StartCoordinateX FLOAT,
    @StartCoordinateY FLOAT,
    @EndCoordinateX FLOAT,
    @EndCoordinateY FLOAT,
    @Operation NVARCHAR(10)
AS
BEGIN
    SET NOCOUNT ON;

    IF @Operation = 'CREATE'
    BEGIN
        INSERT INTO Trees (TreeId, MissionId, StartCoordinateX, StartCoordinateY, EndCoordinateX,
EndCoordinateY)
            VALUES (@TreeId, @MissionId, @StartCoordinateX, @StartCoordinateY, @EndCoordinateX,
@EndCoordinateY);
    END

    IF @Operation = 'READ'
    BEGIN

```

```
SELECT * FROM Trees WHERE TreeId = @TreeId;
END
```

```
IF @Operation = 'UPDATE'
BEGIN
    UPDATE Trees
    SET MissionId = @MissionId,
        StartCoordinateX = @StartCoordinateX,
        StartCoordinateY = @StartCoordinateY,
        EndCoordinateX = @EndCoordinateX,
        EndCoordinateY = @EndCoordinateY
    WHERE TreeId = @TreeId;
END
```

```
IF @Operation = 'DELETE'
BEGIN
    DELETE FROM Trees WHERE TreeId = @TreeId;
END
END
```

Лістинг програмного коду, що реалізує запропонований у роботі метод динамічного отримання зображень заданих структурних об'єктів в тривимірному просторі за допомогою декількох БПЛА.

```
public class TcpConnectionEncryptionActor : UntypedActor {
    private readonly IActorRef _connection;
    private readonly IActorRef _droneInfoDeliveringActor;
    private readonly ILoggingService _loggingService;
    private byte[] _buffer;
    private readonly DroneConnectionManager _manager;
    public TcpConnectionEncryptionActor(IActorRef connection, IPEndPoint ip, IDroneService droneService,
    IFlyingSessionService flyingSessionService, ILoggingService loggingService) {
        _droneInfoDeliveringActor = Context.ActorOf(Context.DI().Props<DroneInfoDeliveringActor>(),
    ActorNames.DRONE_INFO_DELIVERING_ACTOR);
        _manager = new DroneConnectionManager(ip.Address, droneService, flyingSessionService, loggingService);
        _manager.OnDroneOnlineStatusChanged += Manager_OnDroneStatusChanged;
        _manager.OnDroneFlyingStatusChanged += Manager_OnDroneStatusChanged;
        _manager.OnDroneFlyingSessionChanged += Manager_OnDroneFlyingSessionChanged;
        _manager.OnDroneLocationChanged += Manager_OnDroneLocationChanged;
        _manager.OnDroneVideoPartReceive += Manager_OnDroneVideoPartReceive;
        _loggingService = loggingService;
        _connection = connection;}
    protected override void OnReceive(object message) {
        if (message is Tcp.Received received) {
            if (received.Data[0] == 'x')
                Context.Stop(Self);
            else {
                try {
                    if (_buffer == null) {
                        _buffer = received.Data.ToArray();
                    } else {
```

```

    _buffer = _buffer
    .Concat(received.Data.ToArray())
    .ToArray();
  } } catch (Exception exc) {
    _loggingService.LogError(exc.Message, exc);}
  } else if (message is Tcp.PeerClosed closed) {
    // Received drone connection port
    string port = Encoding.UTF8.GetString(_buffer);
    _ = _manager.ConnectToServerAsync(port);
    //_ = ConnectToDroneAsync(port);
    _buffer = null;
  } else Unhandled(message);}
private Task ConnectToDroneAsync(string port) =>
Task.Factory.StartNew(async () => {
do {
if (_manager != null && !_manager.IsClosed) {
await _manager.ConnectToServerAsync(port);
//await Task.Delay(200);
} } while (_manager != null);
}, TaskCreationOptions.LongRunning);
private void Manager_OnDroneVideoPartReceive(object sender, byte[] videoFeed) {
if (sender is DroneConnectionManager manager) {
//_droneInfoDeliveringActor.Tell(new VideoFeedUpdated(videoFeed, _manager.Drone.UserIdentity.Id,
HubEvents.DroneVideoFeedUpdatedHubEvent), _connection);}
private void Manager_OnDroneLocationChanged(object sender, DroneLocationChangedEventArgs e) {
_droneInfoDeliveringActor.Tell(new DroneIncomeCoordinateMessage(
new DroneCoordinatesMessage(_manager.SerialNumber, e.Latitude, e.Longitude),
HubEvents.CoordinatesChangedHubEvent,
e.OwnerId
), _connection);}
private void Manager_OnDroneFlyingSessionChanged(object sender, DroneFlyingSessionChangedEventArgs e) {
_droneInfoDeliveringActor.Tell(
new SessionStateChangedMessage(
new DroneFlyingSessionStateChanged(
e.FlyingSessionId,
e.DroneId,
e.IsFlying
? DroneFlyingSessionStateChanged.SessionState.Started
: DroneFlyingSessionStateChanged.SessionState.Stopped,
e.UserId),
HubEvents.DroneSessionChangedHubEvent
),_connection);
}
private void Manager_OnDroneStatusChanged(object sender, DroneStatusChangedEventArgs e) {
_droneInfoDeliveringActor.Tell(new DroneStateChangedIncomeMessage(e.Drone,
HubEvents.DroneStatusChangedHubEvent), _connection);}
namespace Harvester.Drone.DTO{
public interface IDroneRpcController{
Task<DroneSDKError> FlightControllerHandler_StartTakeoffAsync();
Task<DroneSDKError> FlightControllerHandler_StartAutoLandingAsync();
Task<string> GetDroneSerialNumberAsync();
Task<DroneSDKError> GimbalHandler_RotateByAngleAsync(GimbalAngleRotationDto
gimbalAngleRotationDto);
//void UpdateJoystickValue(float throttle, float yaw, float pitch, float roll);
event EventHandler<IntMsgEventArgs> BatteryHandler_ChargeRemainingInPercentChanged;

```



```

options.SkipNegotiation = true;})
.Build();
hubConnection.Closed += OnHubConnectionClosed;
return hubConnection;}
private Task OnHubConnectionClosed(Exception arg) =>
Task.Run(() => {
IsConnected = false;
TryConnectToHub();});
private Task DisposeCurrentHubConnectionAsync() =>
Task.Run(async () => {try {
if (_hubConnection != null) {
_hubConnection.Closed -= OnHubConnectionClosed;
switch (_hubConnection.State) {
case HubConnectionState.Disconnected:
// Not sure is it necessary to do smth in that case.
break;
case HubConnectionState.Connected:
await _hubConnection.StopAsync();
await _hubConnection.DisposeAsync();
break;
default:
// TODO: unsupported case. Feel fre to handle it.
break;}
_hubConnection = null;}
IsConnected = false;
} catch (Exception ex) {
Console.WriteLine($"ERROR: --{ex.Message}");
Debugger.Break();
_hubConnection = null;
IsConnected = false;}});
private async void TryConnectToHub() {
int connectionShotsCounter = 0;
while (!IsConnected
&& connectionShotsCounter < ReconnectTriesLimit) {
try {await _hubConnection.StartAsync();
IsConnected = true;} catch (Exception exc) {
Console.WriteLine("LENS signal service [{0}]. Hub connection shot #{1}. Details: {2}", SocketHubGateway,
connectionShotsCounter, exc.Message);
IsConnected = false;
connectionShotsCounter++;}
await Task.Delay(DelayForConnectionShot);}}}
[Authorize]
[AssignControllerLocalizedRoute(WebApiEnvironmnet.Current, WebApiVersion.ApiVersion1,
ApplicationSegments.DroneActivities)]
[AssignControllerRoute(WebApiEnvironmnet.Current, WebApiVersion.ApiVersion1,
ApplicationSegments.DroneActivities)]
public class DroneActivitiesController : WebApiControllerBase {
private readonly IDroneService _droneService;
private readonly IFlyingSessionService _flyingSessionService;
public DroneActivitiesController(
IDroneService droneService,
IFlyingSessionService flyingSessionService,
IResponseFactory responseFactory,
IStringLocalizer<DroneActivitiesController> localizer)
: base(responseFactory, localizer) {

```

```

    _droneService = droneService;
    _flyingSessionService = flyingSessionService;}
    [HttpGet]
    [AssignActionRoute(DroneActivitiesSegment.FlyingSession)]
    public async Task<IActionResult> GetActiveFlyingSessionAsync([FromQuery]string droneSerialNumber) {
        try
            {return
                Ok(SuccessResponseBody(await
_flyingSessionService.GetDroneActiveFlyingSessionAsync(droneSerialNumber), Localizer["Successful"]));
        } catch (System.Exception exc) {
            Log.Error(exc.Message);
            return BadRequest(ErrorResponseBody(exc.Message, System.Net.HttpStatusCode.BadRequest));} }
    [HttpGet]
    [AssignActionRoute(DroneActivitiesSegment.FlyingActiveSessionByDroneId)]
    public async Task<IActionResult> GetActiveFlyingSessionByDroneIdAsync([FromQuery]Guid droneId) {
        try {return Ok(SuccessResponseBody(await _flyingSessionService.GetDroneActiveFlyingSessionAsync(droneId,
ClaimHelper.GetUserId(User)), Localizer["Successful"]));} catch (System.Exception exc) {
            Log.Error(exc.Message);
            return
                BadRequest(ErrorResponseBody(exc.Message,
System.Net.HttpStatusCode.BadRequest));} }
    [HttpGet]
    [AssignActionRoute(DroneActivitiesSegment.FlyingSessionAll)]
    public async Task<IActionResult> GetFlyingSessionsAsync([FromQuery]string droneSerialNumber) {
        try
            {
                return
                    Ok(SuccessResponseBody(await
_flyingSessionService.GetDronesFlyingHistoryAsync(droneSerialNumber), Localizer["Successful"]));
            } catch (System.Exception exc) {
                Log.Error(exc.Message);
                return
                    BadRequest(ErrorResponseBody(exc.Message, System.Net.HttpStatusCode.BadRequest));} }
    [HttpGet]
    [AllowAnonymous]
    [AssignActionRoute(DroneActivitiesSegment.DroneVideoFeed)]
    public FileStreamResult GetVideoStream([FromQuery]Guid droneId) {
        HttpContext.Response.Headers.Add("Content-Disposition", "attachment; filename=foo_vid.mp4");
        FileInfo file = new FileInfo(Path.Combine(ConfigurationManager.ExportDirectoryPath, "foo_vid.mp4"));
        if (file.Exists) {
            FileStream stream = file.OpenRead();
            return new FileStreamResult(stream, "video/mp4");}
        return null;} }
    }
    [Authorize]
    [AssignControllerLocalizedRoute(WebApiEnvironmnet.Current,
                                    WebApiVersion.ApiVersion1,
ApplicationSegments.Drones)]
    [AssignControllerRoute(WebApiEnvironmnet.Current,
                                    WebApiVersion.ApiVersion1,
ApplicationSegments.Drones)]
    public class DroneController : WebApiControllerBase {
        private readonly IDroneService _droneService;
        public DroneController(IDroneService droneService,
                                IResponseFactory responseFactory,
                                IStringLocalizer<DroneController> localizer)
            : base(responseFactory, localizer) {
            _droneService = droneService;}
    [HttpGet]
    [AssignActionRoute(DroneSegment.GetDrones)]
    public async Task<IActionResult> GetDronesAsync() {
        try {
            return
                Ok(SuccessResponseBody(await
_droneService.GetRegisteredDronesAsync(ClaimHelper.GetUserId(User))));
        } catch (System.Exception exc) {
            Log.Error(exc.Message);

```

```

return BadRequest(ErrorResponseBody(exc.Message, System.Net.HttpStatusCode.BadRequest));}}
[HttpGet]
[AssignActionRoute(DroneSegment.GetDrone)]
public async Task<IActionResult> GetDroneByIdAsync([FromQuery]Guid id) {
try {
return Ok(SuccessResponseBody(await _droneService.GetRegisteredDroneByIdAsync(id,
ClaimHelper.GetUserId(User))));
} catch (System.Exception exc) {
Log.Error(exc.Message);
return BadRequest(ErrorResponseBody(exc.Message, System.Net.HttpStatusCode.BadRequest));}}
[HttpPost]
[AssignActionRoute(DroneSegment.AddDrone)]
public async Task<IActionResult> AddDronesAsync([FromBody]DroneModel drone) {
try {
return Ok(SuccessResponseBody(await _droneService.SaveDroneAsync(ClaimHelper.GetUserId(User), drone)));
} catch (System.Exception exc) {
Log.Error(exc.Message);
return BadRequest(ErrorResponseBody(exc.Message, System.Net.HttpStatusCode.BadRequest));}}
[HttpPut]
[AssignActionRoute(DroneSegment.UpdateDrone)]
public async Task<IActionResult> UpdateDronesAsync([FromBody]DroneModel drone) {
try {
return Ok(SuccessResponseBody(await _droneService.SaveDroneAsync(ClaimHelper.GetUserId(User), drone)));
} catch (CosmosException cExc) {
switch (cExc.StatusCode) {
case System.Net.HttpStatusCode.NotFound:
return BadRequest(ErrorResponseBody(Localizer["Drone not found"], System.Net.HttpStatusCode.BadRequest));
default:
return BadRequest(ErrorResponseBody(cExc.Message, System.Net.HttpStatusCode.BadRequest));
}} catch (System.Exception exc) {
Log.Error(exc.Message);
return BadRequest(ErrorResponseBody(exc.Message, System.Net.HttpStatusCode.BadRequest));
}}
[HttpDelete]
[AssignActionRoute(DroneSegment.DeleteDrone)]
public async Task<IActionResult> DeleteDronesAsync([FromQuery]Guid droneId) {
try {
return Ok(SuccessResponseBody(await _droneService.DeleteDroneAsync(droneId,
ClaimHelper.GetUserId(User))));
} catch (System.Exception exc) {
Log.Error(exc.Message);
return BadRequest(ErrorResponseBody(exc.Message, System.Net.HttpStatusCode.BadRequest));}} [Authorize]
[AssignControllerLocalizedRoute(WebApiEnvironmnet.Current, WebApiVersion.ApiVersion1,
ApplicationSegments.Apples)]
[AssignControllerRoute(WebApiEnvironmnet.Current, WebApiVersion.ApiVersion1,
ApplicationSegments.Apples)]
return BadRequest(ErrorResponseBody(exc.Message, System.Net.HttpStatusCode.BadRequest));}}
[HttpDelete]
[AssignActionRoute(ApplesSegment.DeleteApple)]
public async Task<IActionResult> DeleteApples([FromQuery]Guid id) {
try {
AppleVarieties result = await _appleService.DeleteUserAppleVariety(ClaimHelper.GetUserId(User), id);

```

Лістинг програмного коду, що реалізує запропонований у роботі метод синхронізації відеопотоків в режимі реального часу.

```

class VideoFrame{
public:
long Rotation;
std::chrono::system_clock::time_point CreatedAt;
int ResolutionWidth;
int ResolutionHeight;
std::string Id;
int FrameRate;
std::string CodecFormat;
VideoFrame()
: Id(generateId()){ }
private:
std::string generateId(){
std::random_device rd;
std::mt19937 gen(rd());
std::uniform_int_distribution<> dis(0, 15);
constexpr char hexChars[] = "0123456789ABCDEF";
std::string id(32, '0');
for (auto& ch : id){
ch = hexChars[dis(gen)];}
return id; } };
class EnhancedVideoFrame{
public:
std::unordered_map<std::string, std::shared_ptr<VideoFrame>> VideoFrames;};
class Resolution{
public:
int Width;
int Height; };
class VideoProcessing{
private:
std::mutex lockObject;
public:
std::shared_ptr<EnhancedVideoFrame> ProcessVideoFrames(){
std::vector<std::shared_ptr<VideoFrame>> videoFrames = GenerateVideoFrames();
std::unordered_map<std::chrono::system_clock::time_point, std::shared_ptr<VideoFrame>> frameDictionary =
RotateAndStoreFrames(videoFrames);
std::shared_ptr<EnhancedVideoFrame> enhancedVideoFrame = MergeFrames(frameDictionary);
return enhancedVideoFrame;}
private:
std::vector<std::shared_ptr<VideoFrame>> GenerateVideoFrames(){
std::vector<std::shared_ptr<VideoFrame>> videoFrames;
for (int i = 1; i <= 3; ++i){
auto videoFrame = std::make_shared<VideoFrame>();
videoFrame->Rotation = i * 10;
videoFrame->CreatedAt = std::chrono::system_clock::now() + std::chrono::milliseconds(i * 50);
videoFrame->ResolutionWidth = 1920;
videoFrame->ResolutionHeight = 1080;
videoFrame->FrameRate = 30;
videoFrame->CodecFormat = "H.264";
}
}
}

```



```

videoFrames.push_back(videoFrame);}
return videoFrames;}
std::unordered_map<std::chrono::system_clock::time_point, std::shared_ptr<VideoFrame>>
RotateAndStoreFrames(const std::vector<std::shared_ptr<VideoFrame>>& videoFrames){
    std::unordered_map<std::chrono::system_clock::time_point, std::shared_ptr<VideoFrame>> frameDictionary;
    long rotation = videoFrames[0]->Rotation;
    for (const auto& videoFrame : videoFrames){
        std::lock_guard<std::mutex> lock(lockObject);
        videoFrame->Rotation = rotation;
        if (frameDictionary.count(videoFrame->CreatedAt) > 0){
            continue;}
        frameDictionary[videoFrame->CreatedAt] = videoFrame;}
    return frameDictionary;
}
std::shared_ptr<EnhancedVideoFrame> MergeFrames(const
std::unordered_map<std::chrono::system_clock::time_point, std::shared_ptr<VideoFrame>>& frameDictionary){
    auto enhancedVideoFrame = std::make_shared<EnhancedVideoFrame>();
    for (const auto& kvp : frameDictionary){
        return enhancedVideoFrame;
    }
}
cv::Mat compressVideoFrame(cv::Mat frame, float scale, float angle){
    int width = frame.cols;
    int height = frame.rows;
    cv::Point2f center(width / 2.0f, height / 2.0f);
    cv::Mat rotationMatrix = cv::getRotationMatrix2D(center, angle, scale);
    cv::Mat compressedFrame;
    cv::warpAffine(frame, compressedFrame, rotationMatrix, cv::Size(width, height));
    return compressedFrame;}

```

Лістинг програмного коду, що реалізує вдосконалений метод виявлення заданих структурних об'єктів на зображеннях.

```

import torch
import torch.nn as nn
import torch.nn.functional as F
import cv2
import numpy as np
from filterpy.kalman import KalmanFilter
from scipy.spatial import distance

class ModifiedBackbone(nn.Module):
    def __init__(self):
        super(ModifiedBackbone, self).__init__()
        self.split1 = nn.Conv2d(3, 32, kernel_size=3, stride=2, padding=1)
        self.split2 = nn.Conv2d(3, 32, kernel_size=3, stride=2, padding=1)
        self.split3 = nn.Conv2d(3, 32, kernel_size=3, stride=2, padding=1)
        self.split4 = nn.Conv2d(3, 32, kernel_size=3, stride=2, padding=1)
        self.combine = nn.Conv2d(128, 32, kernel_size=1, stride=1)
        self.conv1 = nn.Conv2d(32, 32, kernel_size=3, stride=1, padding=1)
        self.conv2 = nn.Conv2d(32, 32, kernel_size=3, stride=1, padding=1)
        self.bn = nn.BatchNorm2d(32)
        self.activation = nn.Hardswish(inplace=True)

```

```

def forward(self, x):
    x1 = self.split1(x)
    x2 = self.split2(x)
    x3 = self.split3(x)
    x4 = self.split4(x)
    x_combined = torch.cat((x1, x2, x3, x4), dim=1)
    x_combined = self.combine(x_combined)
    x = self.conv1(x_combined)
    x = self.conv2(x)
    x = self.bn(x)
    x = self.activation(x)
    return x

class ModifiedYOLOv5(nn.Module):
    def __init__(self):
        super(ModifiedYOLOv5, self).__init__()
        self.backbone = ModifiedBackbone()

    def forward(self, x):
        x = self.backbone(x)
        return x

def track_objects(frame, detected_objects):
    kalman_filter = KalmanFilter(dim_x=4, dim_z=2)
    kalman_filter.F = np.array([[1, 0, 1, 0],
                                [0, 1, 0, 1],
                                [0, 0, 1, 0],
                                [0, 0, 0, 1]])
    kalman_filter.H = np.array([[1, 0, 0, 0],
                                [0, 1, 0, 0]])
    kalman_filter.P *= 10
    kalman_filter.R = np.array([[0.1, 0],
                                [0, 0.1]])
    kalman_filter.Q = np.eye(4) * 0.01

    tracked_objects = []
    for obj in detected_objects:
        x, y, w, h = obj
        z = np.array([[x + w / 2], [y + h / 2]])
        kalman_filter.predict()
        kalman_filter.update(z)
        state = kalman_filter.x
        x, y, w, h = obj
        centroid = np.array([x + w / 2, y + h / 2])
        mahalanobis_distance = distance.mahalanobis(centroid, state, inv_covariance)
        tracked_objects_with_distance.append((obj, mahalanobis_distance))

    return tracked_objects
using Python.Runtime;
using System.Collections.Generic;

public class ObjectDetected
{
    public string ColorFrame { get; set; }
}

```

```

public string EnhancedFrameId { get; set; }
public List<Coordinate> FrameCoordinates { get; set; }
}

public class Coordinate
{
    public int X { get; set; }
    public int Y { get; set; }
}

public class PythonScriptService
{
    private dynamic _pythonEngine;

    public PythonScriptService()
    {
        PythonEngine.Initialize();
        _pythonEngine = Py.Import("__main__");
    }

    public List<ObjectDetected> DetectObjects(EnhancedVideoFrame frame)
    {
        using (Py.GIL())
        {
            dynamic pythonModule = Py.Import("your_python_module"); // Replace with your actual Python module name
            dynamic pythonFrame = ConvertToPythonFrame(frame);
            dynamic detectedObjects = pythonModule.detect_objects(pythonFrame);

            List<ObjectDetected> objects = ConvertToCSharpObjects(detectedObjects);

            return objects;
        }
    }

    private List<ObjectDetected> ConvertToCSharpObjects(dynamic pythonObjects)
    {
        List<ObjectDetected> objects = new List<ObjectDetected>();

        foreach (dynamic pythonObject in pythonObjects)
        {
            ObjectDetected obj = new ObjectDetected();

            obj.ColorFrame = pythonObject.color_frame;
            obj.EnhancedFrameId = pythonObject.enhanced_frame_id;

            obj.FrameCoordinates = new List<Coordinate>();
            foreach (dynamic pythonCoordinate in pythonObject.frame_coordinates)
            {
                Coordinate coordinate = new Coordinate();
                coordinate.X = pythonCoordinate.x;
                coordinate.Y = pythonCoordinate.y;

                obj.FrameCoordinates.Add(coordinate);
            }
        }
    }
}

```

```
    objects.Add(obj);  
  }  
  return objects;  
}  
}
```

```
[ApiController]  
[Route("api/[controller]")]  
public class ObjectDetectionController : ControllerBase  
{  
  private readonly PythonScriptService _pythonScriptService;  
  public ObjectDetectionController(PythonScriptService pythonScriptService)  
  {  
    _pythonScriptService = pythonScriptService;  
  }  
  [HttpPost]  
  public IActionResult DetectObjectCounts([FromBody] EnhancedVideoFrame frame)  
  {  
    var objectCounts = _pythonScriptService.DetectObjectCounts(frame);  
    return Ok(objectCounts);  
  }  
}
```