

## ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ ТА РОЗРОБЛЕННЯ СИСТЕМИ КЕРУВАННЯ ПРИБРОЄМ НА БАЗІ ШТУЧНОГО ІНТЕЛЕКТУ

*Анотація.* Стаття присвячена розробці та реалізації інтелектуальної системи керування FPV-пристроєм, яка здатна виконувати автономне донаведення на ціль при втраті зв'язку з оператором. Актуальність дослідження обумовлена широким застосуванням FPV-пристроїв та необхідністю подолання їх ключового недоліку - залежності від стабільного радіозв'язку.

Система реалізована на платформі Orange Pi 5 Max та використовує модель YOLOv11 для детекції об'єктів, навчену на наборі даних VisDrone з подальшою оптимізацією для NPU. Для трекінгу множинних об'єктів застосовано алгоритм SORT з фільтром Калмана. Розроблено модульну багатопотокову архітектуру з інтеграцією протоколу MAVLink для комунікації з польотним контролером ArduPilot та PID-регуляторів для режиму автономного переслідування.

Експериментальні випробування на статичному стенді підтвердили працездатність системи, проте виявили певні обмеження продуктивності: модель YOLOv11n забезпечує латентність 70-80 мс (камера 50 мс + інференс 25-30 мс), що відповідає вимогам реального часу, тоді як YOLOv11s демонструє кращу точність детекції, але сумарну латентність понад 120 мс. INT8 квантизація забезпечила прискорення у 1.5 рази з мінімальним зниженням точності. Для практичного застосування рекомендовано використання швидкодіючої камери з меншою затримкою, портування критичних компонентів на C/C++ та оптимізацію системного середовища.

*Ключові слова:* FPV-пристрій, комп'ютерний зір, трекінг, детекція, YOLO, SORT, ORANGEPI, RK3588

**Постановка проблеми.** Стрімкий розвиток технологій безпілотних літальних апаратів відкриває нові можливості для автоматизації широкого спектру завдань у цивільній та спеціальній сферах. Масова поява FPV-пристроїв (англ. First Person View) здійснила прорив у галузі інформаційних технологій, що дозволило керувати пристроєм з ефектом присутності при забезпеченні високої маневреності, точності і можливості польотів у складному середовищі та забезпечити миттєву трансляцію з мінімальною затримкою. Серед недоліків FPV-пристроїв виділяють відсутність автоматичних режимів стабілізації (GPS) у більшості спортивних моделей та необхідність підтримувати стійкий зв'язок з оператором по відео каналу та каналу керування. При використанні FPV-пристроїв існує ризик втрати зв'язку, коли висота польоту знижується і зникає умова прямої видимості станції керування. Проблему втрати зв'язку можна

вирішити за допомогою системи донаведення, яка побудована на базі комп'ютерного зору і штучного інтелекту.

**Аналіз останніх досліджень і публікацій.** Серед існуючих рішень можна виділити комерційні, які більше акцентовані на задачах спортивних перегонів, екстремальної аерозйомки і кінематографії та в більшості випадків використовують супутникову навігацію. В реаліях сьогодення при активному розвитку цього напрямку існує декілька спеціальних платформ з запропонованими рішеннями та програмним забезпеченням від виробника (наприклад, платформи VGI-9 [1] та Zir System [2]). Окрім того існує багато різних аматорських та напівпрофесійних команд, які займаються вивченням алгоритмів керування пристроями в умовах протидії ризикам втрат зв'язку. Отже, надалі не пропонується абсолютно новий підхід або нове рішення, а в більшій мірі досліджується варіант створення подібної системи на базі недорогого одноплатного комп'ютера.

Для забезпечення стабілізації та виконання базових команд керування сучасні FPV-пристрої використовують спеціалізовані контролери. Найпоширенішими платформами є Betaflight, ArduPilot та PX4, з яких Ardupilot – платформа з відкритим вихідним кодом, з підтримкою зручного протоколу комунікації з MAVLink, який може бути використаний для зв'язку з компаньйон-комп'ютером (англ. companion computer) [3].

Серед популярних компаньйон-комп'ютерів можна відокремити Nvidia Jetson Nano [4], який має потужний GPU для задач комп'ютерного зору, але і має високу ціну. Новітній, надпотужний одноплатний мінікомп'ютер Raspberry Pi 5 з використанням додаткового розширення NPU [5] має значно оновлену архітектуру, що дозволяє використовувати його для розробки, штучного інтелекту та серверів. Для проведення експериментів авторами було обрано Orange Pi 5 на базі чіпсету Rockchip RK3588, який вже має вбудований прискорювач NPU продуктивністю 6 TOPS [6], за його привабливі характеристики та низьку ціну.

Детекція об'єктів є фундаментальною задачею комп'ютерного зору, що полягає у визначенні положення та класифікації об'єктів на зображенні. Для систем керування FPV-пристроями критичними є швидкість обробки та точність детекції. Традиційні методи детекції мають низьку точність на складних сценах та повільну роботу. Двоступеневі детектори мають високу точність, але низьку швидкість. Для підтримки балансу точності та швидкості науковцями запропоновано одноступеневі детектори, SSD та модель YOLO [7]. У запропонованому авторами рішенні в якості детектора використана модель YOLOv11 з покращеною архітектурою для ефективної обробки малих об'єктів. В розробленій системі підтримується експорт у відкриту бібліотеку програмного забезпечення Open Neural Network Exchange (ONNX) для побудови нейронних мереж глибокого навчання та подальшої оптимізації і розгортання на цільових платформах.

Після детекції об'єктів на окремих кадрах постає задача їх трекінгу – встановлення відповідності між детекціями на послідовних кадрах. Традиційні алгоритми SORT та його розширення DeepSORT використовують переважно фільтр Калмана для передбачення і згладжування та Hungarian Algorithm для асоціації, мають високу продук-

тивність, але погано працюють з оклюзіями (ситуаціями з розташуванням об'єктів та досягненням до певної частини тривимірної сцени через існуючі перепони). Більш сучасні підходи задачі трекінгу реалізовані в алгоритмах ByteTrack, BotSort, StrongSORT, FairMOT, які дозволяють ефективніше працювати з сотнями швидко рухомих об'єктів зі стійкістю до оклюзій. Для реалізації трекінгу в системі авторами обрано швидкий алгоритм SORT, який забезпечує утримання одиночної цілі FPV-пристроєм у випадку, коли детекція може зникати та/або змінюватися. Використаний алгоритм має простоту імплементації та налагодження, високу швидкість роботи та достатню точність для більшості сценаріїв [8].

**Мета дослідження.** Метою даного дослідження є аналіз методів машинного зору та детекції об'єктів, розроблення та реалізація інтелектуальної системи керування FPV-пристроєм на базі штучного інтелекту, яка здатна виконувати автоматичну детекцію, трекінг та відстеження обраних об'єктів у режимі реального часу. Заплановано розгортання системи на граничному рівні (англ. Edge) пристрою з подальшим безпосереднім встановленням відповідного програмного забезпечення.

**Викладення основного матеріалу дослідження.** Основними вимогами до системи керування є продуктивність, точність та ресурсоефективність. Система має забезпечувати обробку відеопотоку з частотою не менше 30 відображених кадрів за одну секунду (англ. Frames Per Second, FPS) на цільовій апаратній платформі Orange Pi 5. Загальна латентність від отримання кадру до генерації керуючої команди не повинна перевищувати 100 мс для забезпечення своєчасної реакції на зміни у сцені. Вимогами для системи є висока гнучкість та сумісність для запуску в середовищі OS Ubuntu Linux 22.04. Для реалізації прототипу використано мову Python через швидкість прототипування, підтримку всіх необхідних компонентів та кросплатформеність.

**Архітектура системи.** Розроблена система керування пристроєм базується на принципах модульної архітектури з чітким розділенням відповідальності між компонентами (рис. 1).

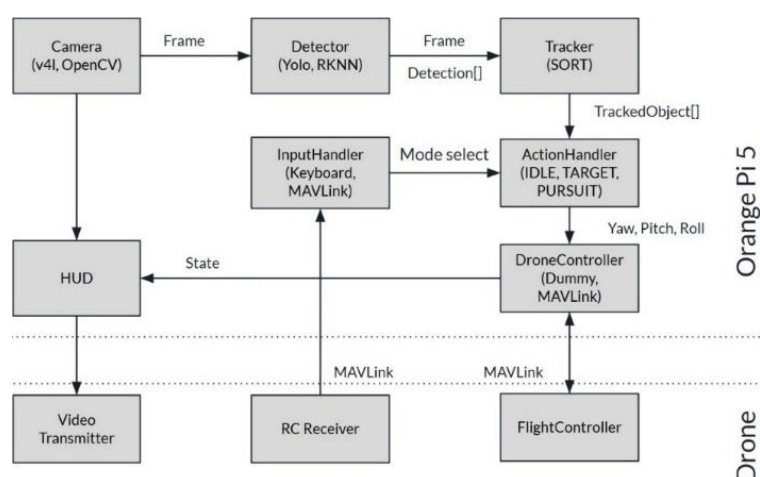


Рисунок 1 – Схема архітектури

Підсистема сприйняття складається з захоплення відеопотоку з камери, детекції об'єктів в кадрі та трекеру виявлених об'єктів.

Підсистема вводу відповідає за комунікацію з дроном, визначення команди на захоплення об'єкту чи на зміну режиму роботи.

Підсистема керування займається обробкою керуючої команди, установкою відповідного алгоритму роботи з виділенням об'єктом, та обчислення впливу на пристрій.

Підсистема виводу Heads Up Display (HUD) виводить інформаційну панель, поточний режим та підсвічує об'єкти, що відстежуються, формує зображення для передачі на відео передавач пристрою.

Система працює за принципом конвеєрної обробки з використанням багатопотоковості (рис. 2) для мінімізації латентності.

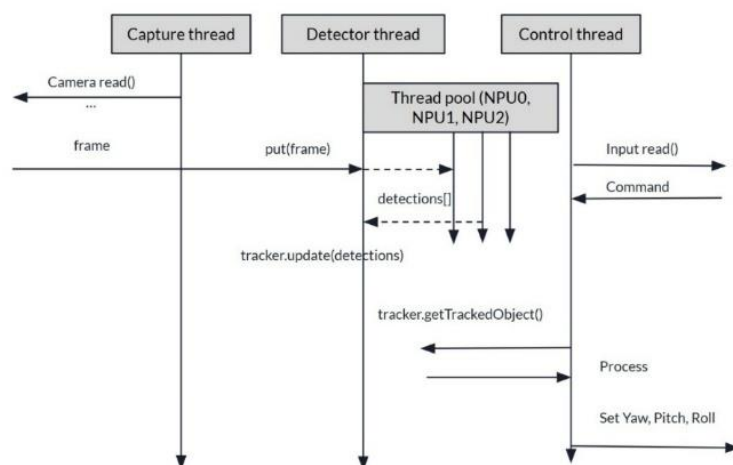


Рисунок 2 – Модель багатопотоковості

Перший потік відповідає за захват зображень з камери, другий – за детекції і відстежування. Чіпсет RK3588 має 3 ядра NPU, які можуть бути використані одночасно, тому детектор RKNN додатково містить пул потоків з завданнями, які назначаються на окреме ядро. Окремий потік виділено для обробки вводу та обчислення стану системи в залежності від поточного режиму та об'єкту, що відстежується.

**Практична апробація системи.** Сімейство моделей YOLOv11 представлено п'ятьма варіантами різного розміру від nano до extralarge.

Найлегша модель YOLOv11n містить лише 2.6 мільйона параметрів та забезпечує швидкість понад 100 FPS на сучасних GPU, що робить її оптимальною для мобільних пристроїв та real-time застосувань, проте за рахунок зниженої точності модель YOLOv11s (9.4М параметрів) представляє оптимальний баланс між продуктивністю та точністю для Edge-пристроїв середнього класу. Саме ці дві моделі було обрано для експериментів в рамках даного дослідження.

Для навчання моделі детекції об'єктів використано комбінований підхід: тренування моделі проходить на великому відкритому датасеті VisDrone [9], пізніше модель може бути донавчена на користувацьких даних, які є специфічними для конкретної задачі. На рисунку 3 наведено графіки процесу навчання моделі на датасеті VisDrone на 300 епохах.

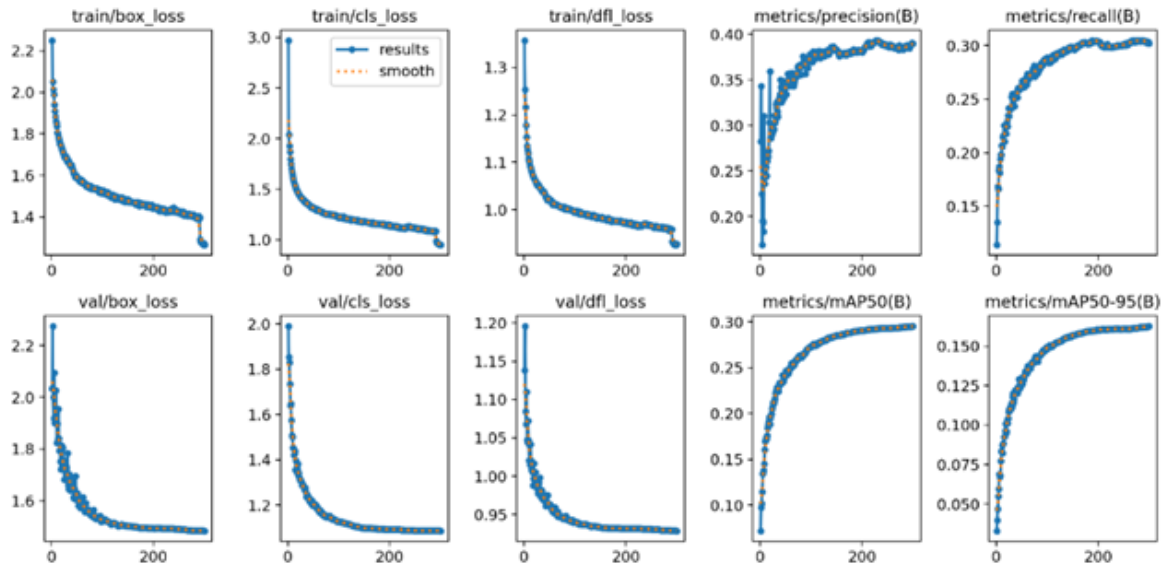


Рисунок 3 – Графік навчання моделі yolo11n на 300 епохах

Для оцінки точності моделей класифікації виконано порівняння графіків коефіцієнта довіри F1 від критичної метрики продуктивності F1-Confidence до/після навчання (рис. 4) показано. Значення точності mAP виявлення об’єктів для всіх класів лишається достатньо низьким, значення впевненості для окремого класу car є суттєвим для використання його в експериментах.

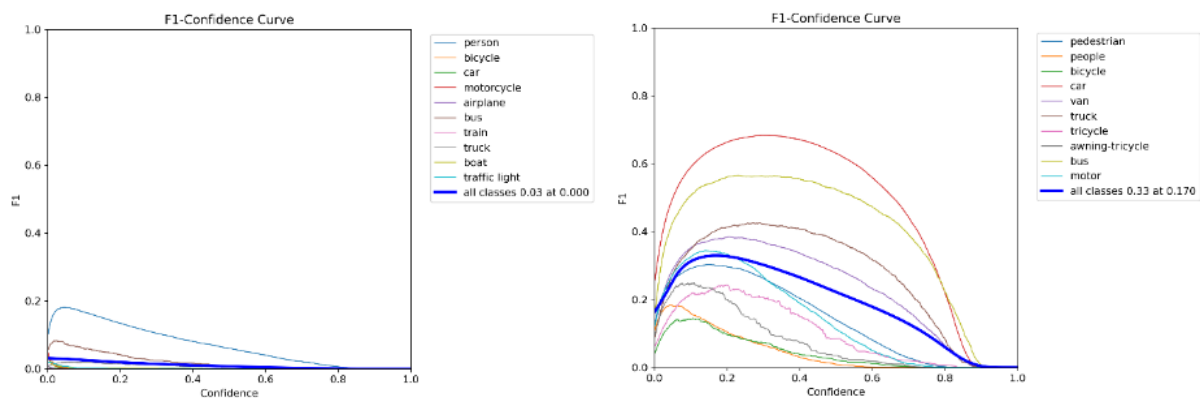


Рисунок 4 – Графіки до та після навчання моделі YOLO11n

Для забезпечення процесу (англ. Fine-tuning) додаткового тренування (донавчання) вже попередньо навченої моделі (англ. pre-trained model) на меншому, вузькоспеціалізованому наборі даних з відео перехоплення, - створений датасет interceptor. Сформований датасет використано для тестів навчання та процедури створення та анотації власних наборів даних. Донавчання на датасеті interceptor проведено за технологією transfer learning, яка полягає в заморозці перших 10 шарів моделі YOLO для збереження низькорівневих ознак та зменшенні learning rate для запобігання катастрофічного забування.

Після навчання модель оптимізовано для роботи на NPU. Процес включає конвертацію у формат ONNX з визначеними перетвореннями: видаленню дублікатів і перек-

риваючих рамок під час детекції об'єктів (англ. Non-Max Suppression) та вирішення проблеми дисбалансу (англ. Distribution focal loss) шарів, через некоректну роботу з квантизацією та уповільненням виконання на Rockchip NPU, з подальшим переносом цих операцій на CPU. Наступним кроком перетворення моделі є конвертація у формат RKNN з INT8 квантизацією, що дає незначне зниження точності, але значно прискорює інференс та зменшує споживання пам'яті.

Тестовий стенд складається з пристрою Orange Pi 5 Max та 13МП камерою. Пристрій підтримує підключення монітору через інтерфейс HDMI для контролю виконання програмного додатку у десктопному середовищі. В реальних умовах система вимагає налаштувань в Headless режимі з виводом відео-напрямку у фреймбуфер (ділянка пам'яті для тимчасового зберігання зображення перед його виведенням на екран). Далі цифровий відеосигнал потрапляє на цифро-аналоговий конвертор звідки у форматі аналогового кольорового телебачення (англ. PAL) подається на відеопередавач пристрою.

Проведено експериментальні випробування розробленої системи керування пристроєм з метою верифікації працездатності ключових компонентів та оцінки їх продуктивності. Випробування виконувалися на статичному стенді без фізичної інтеграції з пристроєм.

Під час роботи зібрані метрики (табл. 1).

Таблиця 1

Основні метрики роботи системи

Модель	Затримка детектора, мс
yolo11n квантизована	25-30
yolo11n не квантизована	48-50
yolo11s квантизована	65-70

Отримане значення затримки трекеру менше 2 мс на кількості об'єктів до 10. На додачу до затримки камери в 45-50 мс отримано ~70 мс повної затримки на моделі YOLO11n, що вкладається в задані 100 мс. Модель YOLO11s продемонструвала кращу роботу детекції, виявивши об'єкти, які знаходились за перепонами, але затримка роботи детектора склала до 70 мс, що сумарно виходить за рамки заданого значення.

Використання квантизованої моделі yolo11n дало приріст продуктивності в 1.5 рази, з некритичним зниженням точності (об'єкти мали рамки різних розмірів). Під час роботи системи завантаження NPU ядер чіпсету RK3588 не перевищувало 18-25%, що пов'язано з фактом використання повільної камери. Температурний режим пристрою під час роботи знаходився в адекватних границях через використання радіатора та кулера активного охолодження, температура процесорного ядра не піднімалась вище 30 градусів. Камера нагрівалась до температури 40 градусів, тому рекомендовано використання окремого додаткового охолодження камери.

**Висновки.** У ході виконання дослідження розроблено та реалізовано інтелектуальну систему керування FPV-пристроєм на базі штучного інтелекту, здатну виконувати автоматичну детекцію та трекінг об'єктів у реальному часі на граничному пристрої

Orange Pi 5 з NPU RK3588. Система базується на архітектурі YOLOv11n та YOLOv11s, навчених на наборі даних VisDrone з подальшою INT8 квантизацією для NPU, та алгоритмі SORT з фільтром Калмана для трекінгу об'єктів. Спроектовано модульну багато-потоківу архітектуру з інтеграцією протоколу для комунікації MAVLink з польотним контролером ArduPilot та реалізовано PID-регулятори для режиму автономного переслідування цілі. Експериментальне тестування підтвердило працездатність системи на статичному стенді, проте виявило обмеження продуктивності: затримка захоплення кадру з камери становила 50 мс, інференс моделі YOLOv11s – 70 мс, що перевищує вимоги систем керування реального часу (<100 мс сумарно). Для досягнення прийнятної латентності рекомендовано використання камери з нижчою роздільністю та зменшеною затримкою, портування критичних компонентів на C/C++ та оптимізацію середовища виконання через вимкнення непотрібних системних сервісів.

Подальший розвиток передбачає тестування в симуляційному середовищі SITL, інтеграцію з реальним пристроєм та випробування для валідації повної функціональності системи автономного донаведення.

#### ЛІТЕРАТУРА

1. VGI-9 team. Autonomous Guidance for Combat Drones. URL: <https://vgi.com.ua/en/> (дата звернення 07.12.2025)
2. ZIP система розпізнавання та ураження цілей. URL: <https://zir-system.com/> (дата звернення 07.12.2025)
3. ArduPilot Dev Team. (2024). ArduPilot Documentation. URL: <https://ardupilot.org/> (дата звернення 07.12.2025)
4. NVIDIA. (2024). Jetson Platform. URL: <https://www.nvidia.com/en-us/autonomous-machines/jetson-store/> (дата звернення 07.12.2025)
5. Raspberry Pi Foundation. (2024). Raspberry Pi 4 Model B Specifications. URL: <https://www.raspberrypi.org/> (дата звернення 07.12.2025)
6. Orange Pi. (2024). Orange Pi 5 Specifications. URL: <http://www.orangepi.org/> (дата звернення 07.12.2025)
7. Ultralytics. (2024). YOLOv11: Next Generation Object Detection. URL: <https://docs.ultralytics.com/models/yolo11/> (дата звернення 07.12.2025)
8. Bewley, A., et al. (2016). Simple online and realtime tracking. IEEE International Conference on Image Processing (ICIP), 3464-3468.
9. Du, D., et al. (2019). VisDrone-DET2019: The vision meets drone object detection in image challenge results. Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops.

## REFERENCES

1. VGI-9 team. Autonomous Guidance for Combat Drones. URL: <https://vgi.com.ua/en/> (дата звернення 07.12.2025)
2. ZIR target recognition and engagement system. URL: <https://zir-system.com/> (дата звернення 07.12.2025)
3. ArduPilot Dev Team. (2024). ArduPilot Documentation. URL: <https://ardupilot.org/> (дата звернення 07.12.2025)
4. NVIDIA. (2024). Jetson Platform. URL: <https://www.nvidia.com/en-us/autonomous-machines/jetson-store/> (дата звернення 07.12.2025)
5. Raspberry Pi Foundation. (2024). Raspberry Pi 4 Model B Specifications. URL: <https://www.raspberrypi.org/> (дата звернення 07.12.2025)
6. Orange Pi. (2024). Orange Pi 5 Specifications. URL: <http://www.orangepi.org/> (дата звернення 07.12.2025)
7. Ultralytics. (2024). YOLOv11: Next Generation Object Detection. URL: <https://docs.ultralytics.com/models/yolo11/> (дата звернення 07.12.2025)
8. Bewley, A., et al. (2016). Simple online and realtime tracking. IEEE International Conference on Image Processing (ICIP), 3464-3468.
9. Du, D., et al. (2019). VisDrone-DET2019: The vision meets drone object detection in image challenge results. Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops.

Received 21.04.2026.  
Accepted 24.04.2026.  
Published 30.04.2026

### ***Research into technologies and development of artificial intelligence-based device control system***

*Analysis of recent studies and publications. Among the existing solutions, commercial ones can be distinguished, which are more focused on the tasks of sports racing, extreme aerial photography and cinematography and in most cases use satellite navigation. In today's realities, with the active development of this direction, there are several special platforms with proposed solutions and software from the manufacturer (for example, the VGI-9 [1] and Zir System [2] platforms).*

*Purpose of research. The purpose of this research is to analyze machine vision and object detection methods, develop and implement an intelligent FPV device control system based on artificial intelligence, which is capable of performing automatic detection, tracking and tracing of selected objects in real time. It is planned to deploy the system at the edge level of the device with subsequent direct installation of the appropriate software.*

*Presentation of the main research material. The main requirements for the control system are performance, accuracy and resource efficiency. The system must provide video stream processing with a frequency of at least 30 displayed frames per second (Frames Per Second, FPS) on the target hardware platform Orange Pi 5.*

*System architecture. The developed device control system is based on the principles of modular architecture with a clear separation of responsibilities between components. The*

*system operates on the principle of pipeline processing using multi-threading to minimize latency.*

*Practical testing of the system. The YOLOv11 model family is represented by five variants of different sizes from nano to extralarge.*

*A combined approach was used to train the object detection model: the model is trained on the large open VisDrone dataset [9], later the model can be further trained on user data that is specific to a particular task.*

*The test bench consists of an Orange Pi 5 Max device and a 13MP camera. In real conditions, the system requires settings in Headless mode with video output to the framebuffer (a memory area for temporary storage of the image before displaying it on the screen).*

*Experimental tests of the developed device control system were conducted to verify the operability of key components and assess their performance. The tests were performed on a static stand without physical integration with the device.*

*Conclusions. In the course of the research, an intelligent FPV device control system based on artificial intelligence was developed and implemented, capable of performing automatic detection and tracking of objects in real time on the Orange Pi 5 edge device with NPU RK3588. A modular multi-threaded architecture was designed with the integration of the MAVLink communication protocol with the ArduPilot flight controller and PID controllers for the autonomous target pursuit mode were implemented. Experimental testing confirmed the system's operability on a static stand, but revealed performance limitations.*

*Keywords: FPV, computer vision, object tracking, object detection, YOLO, SORT, ORANGEPI, RK3588.*

**Сподинець Олег Павлович** - магістр, 121 Інженерія програмного забезпечення Дніпровський національний університет імені Олеся Гончара.

ORCID: <https://orcid.org/0009-0004-9259-8560>

**Божуха Лілія Миколаївна** - кандидат фізико-математичних наук, доцент, Дніпровський національний університет імені Олеся Гончара

ORCID: <https://orcid.org/0000-0002-1420-1220>

**Spodynets Oleh** - master, 121 software engineering Oles Honchar Dnipro national university.

ORCID: <https://orcid.org/0000-0002-1520-1211>

**Bozhukha Liliya Nikolaevna** - Candidate of Physical and Mathematical Sciences, Associate Professor, Oles Honchar Dnipro National University

ORCID: <https://orcid.org/0009-0004-9259-8560>